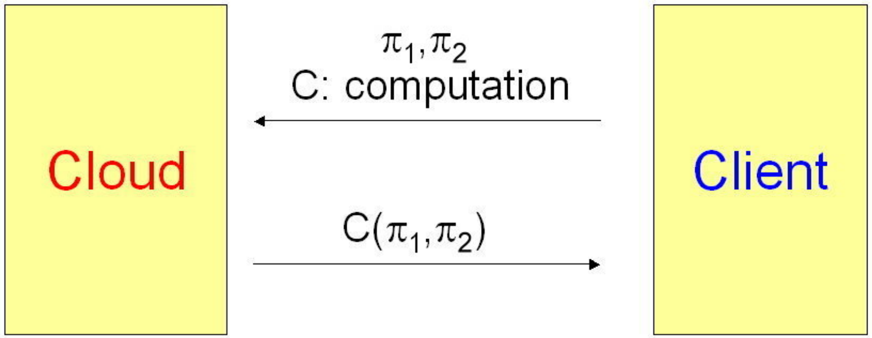
Fully Homomorphic Encryption Using Ideal Lattices

Presenter: Alison Tsai-Yin Lin

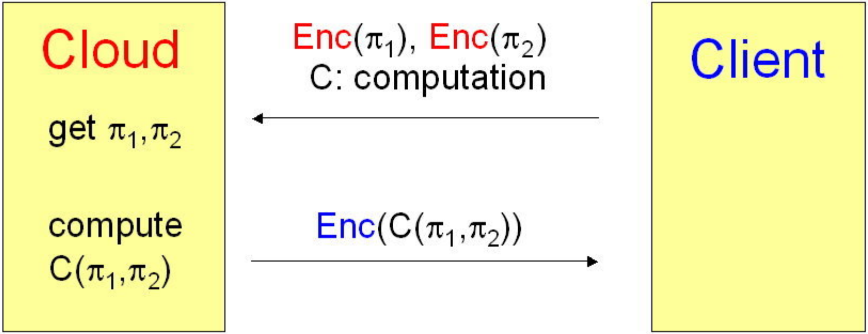
November 16 2010

Cloud computing problem



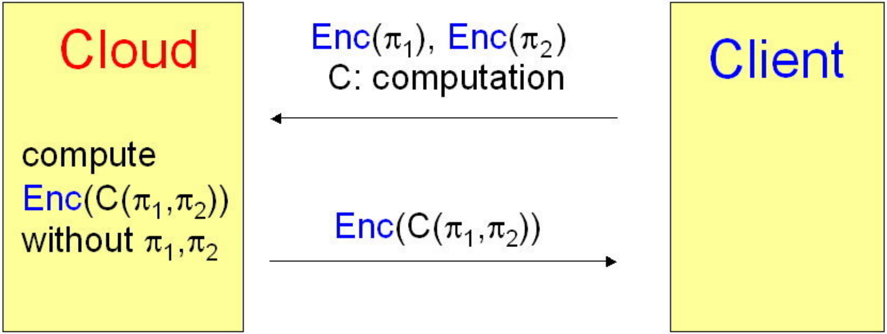
- How to keep $\pi_1, \pi_2, C(\pi_1, \pi_2)$ private from others?

Cloud computing problem



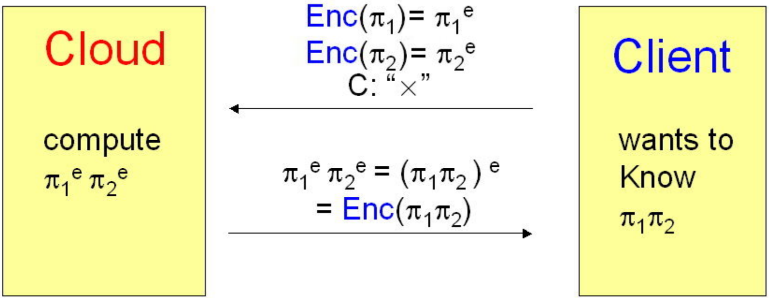
- How to keep $\pi_1, \pi_2, C(\pi_1, \pi_2)$ private from both others and Cloud?

Cloud computing problem



- Use **homomorphic encryption scheme** to do this.

- Example: RSA is a multiplicatively homomorphic encryption scheme, i.e.,
  $Enc(\pi_1\pi_2) = (\pi_1\pi_2)^e = {\pi_1}^e{\pi_2}^e = Enc(\pi_1)Enc(\pi_2)$



Homomorphic Encryption

- Example: RSA is a multiplicatively homomorphic scheme, i.e.,
  $Enc(\pi_1\pi_2) = (\pi_1\pi_2)^e = {\pi_1}^e{\pi_2}^e = Enc(\pi_1)Enc(\pi_2)$

- Example: RSA is a multiplicatively homomorphic scheme, i.e.,
  $Enc(\pi_1\pi_2) = (\pi_1\pi_2)^e = {\pi_1}^e{\pi_2}^e = Enc(\pi_1)Enc(\pi_2)$
- RSA is not additively homomorphic because
  $(\pi_1 + \pi_2)^e \neq {\pi_1}^e + {\pi_2}^e$

Homomorphic Encryption

- Example: RSA is a multiplicatively homomorphic scheme, i.e., $Enc(\pi_1\pi_2) = (\pi_1\pi_2)^e = \pi_1{}^e\pi_2{}^e = Enc(\pi_1)Enc(\pi_2)$
- RSA is not additively homomorphic because $(\pi_1 + \pi_2)^e \neq \pi_1{}^e + \pi_2{}^e$
- A cryptosystem which supports both " $+$ " and " $\times$ " is called a fully homomorphic encryption scheme

- Example: RSA is a multiplicatively homomorphic scheme, i.e.,
  $Enc(\pi_1\pi_2) = (\pi_1\pi_2)^e = {\pi_1}^e{\pi_2}^e = Enc(\pi_1)Enc(\pi_2)$
- RSA is not additively homomorphic because $(\pi_1 + \pi_2)^e \neq {\pi_1}^e + {\pi_2}^e$
- A cryptosystem which supports both " $+$ " and " $\times$ " is called a fully homomorphic encryption scheme
- Idea: Use mod. mod is homomorphic under " $+$ " and " $\times$ ".

- Definition A homomorphic public key encryption scheme has 4 algorithms: KeyGen, Enc, Dec, and Evaluate s.t.

- Definition A homomorphic public key encryption scheme has 4 algorithms: KeyGen, Enc, Dec, and Evaluate s.t.

  Evaluate[pk, C, Enc(pk,π_1),... , Enc(pk,π_t)]
  = Enc[pk, C(π_1,..., π_t)], for all circuit C.

Factor Ring and Lattice

- Let $f(x) \in \mathbb{Z}[x]$ with degree n.
- Let $R := \mathbb{Z}[x]/f(x)$.
- Identify R with n-dim integer lattice $\mathbb{Z}^n$ by

$$a_0 + \cdots + a_{n-1}x^{n-1} \in \mathbb{Z}[x]/f(x) \longleftrightarrow (a_0, \ldots, a_{n-1}) \in \mathbb{Z}^n$$

Factor Ring and Lattice

- Let $f(x) \in \mathbb{Z}[x]$ with degree n.
- Let $R := \mathbb{Z}[x]/f(x)$.
- Identify R with n-dim integer lattice $\mathbb{Z}^n$ by

  $a_0 + \cdots + a_{n-1}x^{n-1} \in \mathbb{Z}[x]/f(x) \longleftrightarrow (a_0, \ldots, a_{n-1}) \in \mathbb{Z}^n$
- additive subgroup of R $\longleftrightarrow$ additive subgroup of $\mathbb{Z}^n$, i.e., sublattice of $\mathbb{Z}^n$
- Let $I \subseteq R$ be an ideal. $I \longleftrightarrow$ sublattice of $\mathbb{Z}^n$.

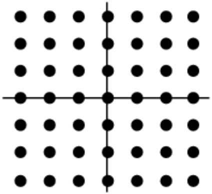
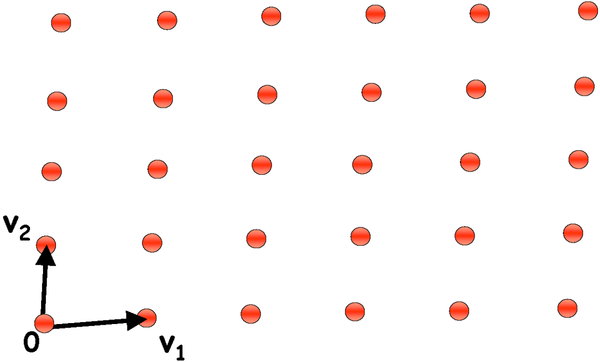


Figure: $\mathbb{Z}^2$ lattice

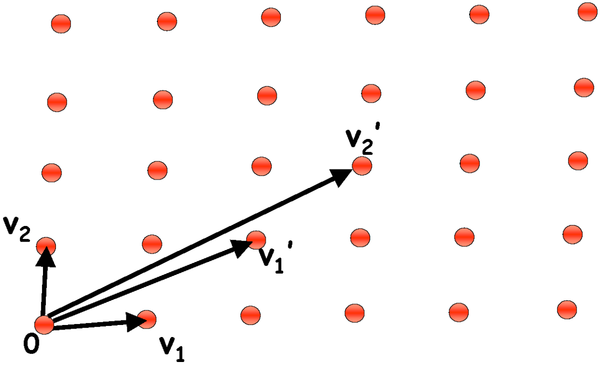
Sublattice of $\mathbb{Z}^n$ and basis

- $B = \{v_1, v_2\}$ is a basis of the lattice $L \subseteq \mathbb{Z}^n$.



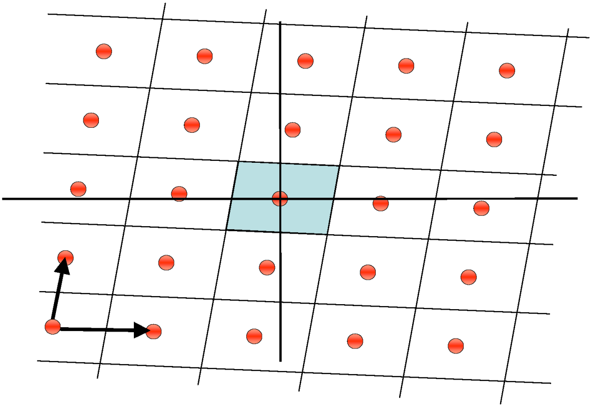
Sublattice of $\mathbb{Z}^n$ and basis

- $B_{sk} = \{v_1, v_2\}$ and $B_{pk} = \{v_1', v_2'\}$ are both bases of the lattice L.

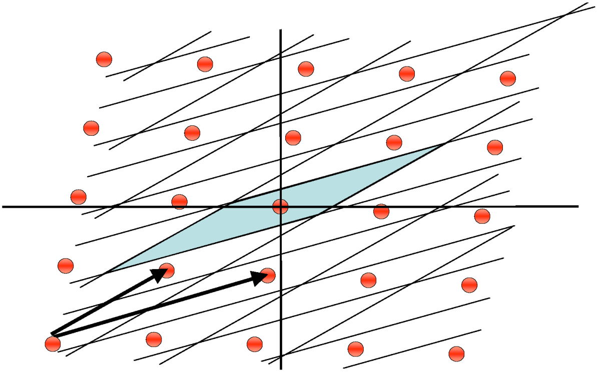


Sublattice of $\mathbb{Z}^n$ and basis

- Let $\mathcal{P}(B_{sk})$ be the parallelepiped of $B_{sk} = \{v_1, v_2\}$, i.e., $\mathcal{P}(B_{sk}) := \{c_1 v_1 + c_2 v_2 \mid c_1, c_2 \in [-\frac{1}{2}, \frac{1}{2})\}$

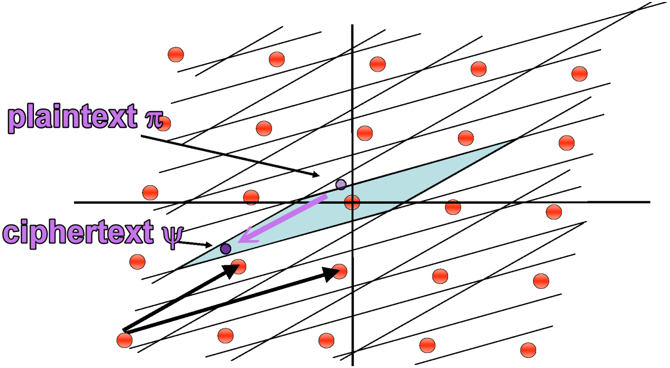


Sublattice of $\mathbb{Z}^n$ and basis

- $\mathcal{P}(B_{pk}) := \{c_1 v'_1 + c_2 v'_2 \mid c_1, c_2 \in [-\frac{1}{2}, \frac{1}{2})\}$
- 

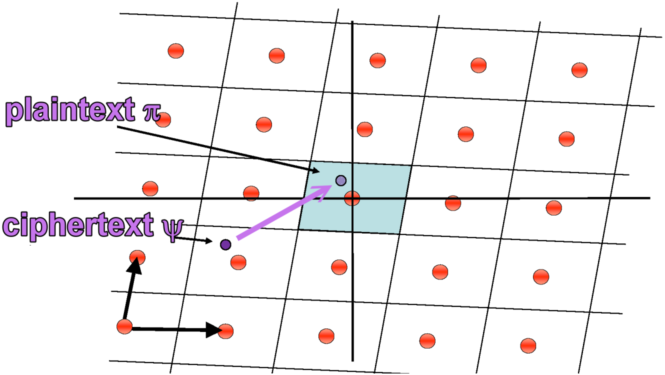
Sketch of the Encryption Scheme

- Enc: $\pi \mapsto \psi = (\pi \mod B_{pk})$
- Enc is homomorphic under " $+$ "," $\times$ "



Sketch of the Encryption Scheme

- Enc: $\pi \mapsto \psi = (\pi \mod B_{pk})$
- Dec: $\psi \mapsto \pi = (\psi \mod B_{sk})$



Obstacle of the Scheme

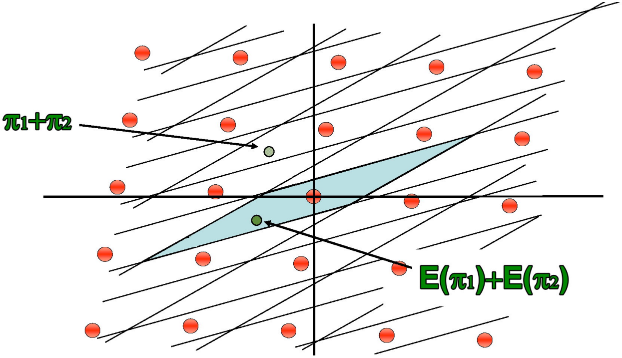
- Dec : everything $\longrightarrow \mathcal{P}(B_{sk})$

Obstacle of the Scheme

- Dec : everything $\longrightarrow \mathcal{P}(B_{sk})$
- If $\pi \notin \mathcal{P}(B_{sk}) \Rightarrow$Dec(Enc($\pi$))$\neq \pi$, i.e., indecypherable.
- We say such plaintext/ciphertext is noisy.
- However, " $+$ " and " $\times$ " make noise.

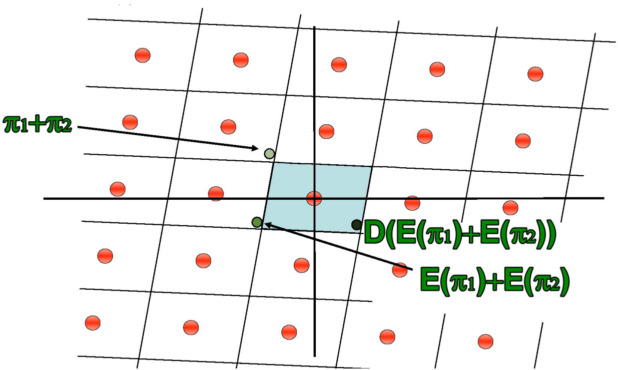
Obstacle of the Scheme

- Dec : everything $\longrightarrow \mathcal{P}(B_{sk})$
- If $\pi \notin \mathcal{P}(B_{sk}) \Rightarrow$Dec(Enc($\pi$))$\neq \pi$, i.e., indecypherable.
- We say such plaintext/ciphertext is noisy.
- However, " $+$ " and " $\times$ " make noise.



Obstacle of the Scheme

- Dec : everything $\longrightarrow \mathcal{P}(B_{sk})$
- If $\pi \notin \mathcal{P}(B_{sk}) \Rightarrow$Dec(Enc($\pi$))$\neq \pi$, i.e., indecypherable.
- We say such plaintext/ciphertext is noisy.
- However, " $+$ " and " $\times$ " make noise.



Idea of Bootstrapping

- Refresh the ciphertext vector when its norm is too long
- i.e., decrypt Enc(π) under **another key**

Idea of Bootstrapping

- Refresh the ciphertext vector when its norm is too long
- i.e., decrypt Enc(π) under **another key**

- Example Given ψ=Enc(pk1,π), Enc(pk2,sk1)

  Recall Evaluate[pk, C, Enc(pk,π_1), Enc(pk,π_2)]
  = Enc[pk, C(π_1,π_2)], for circuit C$\in \mathcal{C}$.

Idea of Bootstrapping

- Refresh the ciphertext vector when its norm is too long
- i.e., decrypt Enc(π) under **another key**

- Example Given ψ=Enc(pk1,π), Enc(pk2,sk1)

  Recall Evaluate[pk, C, Enc(pk,π_1), Enc(pk,π_2)]
  $=$ Enc[pk, C(π_1,π_2)], for circuit C$\in \mathcal{C}$.

  Now if Dec$\in \mathcal{C}$, then compute Enc[pk2,Enc(pk1,π))]

Idea of Bootstrapping

- Refresh the ciphertext vector when its norm is too long
- i.e., decrypt Enc(π) under **another key**
- Example Given ψ=Enc(pk1,π), Enc(pk2,sk1)

  Recall Evaluate[pk, C, Enc(pk,π_1), Enc(pk,π_2)]
  = Enc[pk, C(π_1,π_2)], for circuit C$\in \mathcal{C}$.

  Now if Dec$\in \mathcal{C}$, then compute Enc[pk2,Enc(pk1,π))]
  Evaluate(pk2, Dec, [Enc(pk2,sk1),(Enc(pk2,Enc(pk1,π))])
  =Enc(pk2,Dec(sk1,Enc(pk1,π)))
  =Enc(pk2,π)

Idea of Bootstrapping

- Refresh the ciphertext vector when its norm is too long
- i.e., decrypt Enc(π) under **another key**

- Example Given ψ=Enc(pk1,π), Enc(pk2,sk1)

  Recall Evaluate[pk, C, Enc(pk,π_1), Enc(pk,π_2)]
  $=$ Enc[pk, C(π_1,π_2)], for circuit C$\in \mathcal{C}$.

  Now if Dec$\in \mathcal{C}$, then compute Enc[pk2,Enc(pk1,π))]
  Evaluate(pk2, Dec, [Enc(pk2,sk1),(Enc(pk2,Enc(pk1,π))])
  =Enc(pk2,Dec(sk1,Enc(pk1,π)))
  =Enc(pk2,π)

- We call the encryption scheme is **bootstrappable** if Dec$\in \mathcal{C}$

- We want to lower the complexity of decryption circuit s.t. Dec $\in \mathcal{C}$(the set of permitted circuits).

Bootstrappable encryption scheme

- We want to lower the complexity of decryption circuit s.t. Dec $\in \mathcal{C}$(the set of permitted circuits).
- Q: How much depth is enough? (d)

Bootstrappable encryption scheme

- We want to lower the complexity of decryption circuit s.t. Dec $\in \mathcal{C}$(the set of permitted circuits).
- Q: How much depth is enough? (d)
- Q: How long can a plaintext be before being inputted to a circuit C? (r_{ENC})

- We want to lower the complexity of decryption circuit s.t. Dec $\in \mathcal{C}$(the set of permitted circuits).
- Q: How much depth is enough? (d)
- Q: How long can a plaintext be before being inputted to a circuit C? (r_{ENC})
- Q: How long can a decypherable plaintext be? (r_{DEC})

Bootstrappable encryption scheme

- We want to lower the complexity of decryption circuit s.t. Dec $\in \mathcal{C}$(the set of permitted circuits).
- Q: How much depth is enough? (d)
- Q: How long can a plaintext be before being inputted to a circuit C? (r_{ENC})
- Q: How long can a decypherable plaintext be? (r_{DEC})
- Let r_{ENC}, r_{DEC} be radii s.t.

$$\mathcal{C} = \{C : R \rightarrow R \mid \forall \pi_i \in B(r_{ENC}), C(\pi_1, ..., \pi_t) \in B(r_{DEC})\}$$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;

  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \le |u| + |v| \quad ; \quad |u \times v| \le \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- r_i := max radius at level i of C;
  so $r_d = r_{ENC}$ and $r_0 \le r_{DEC}$
- $r_i \le \gamma_R \cdot r_{i+1}^2$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "+" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;

  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$
- $r_i \leq \gamma_R \cdot r_{i+1}^2$

  $\Rightarrow r_0 \leq \gamma_R^{1+2+\cdots+2^{d-1}} \cdot r_d^{2^d}$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;

  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$
- $r_i \leq \gamma_R \cdot r_{i+1}^2$

  $\Rightarrow r_0 \leq \gamma_R^{1+2+\cdots+2^{d-1}} \cdot r_d^{2^d} \leq \gamma_R^{2^d} r_d^{2^d} = (\gamma_R r_d)^{2^d}$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;

  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$
- $r_i \leq \gamma_R \cdot r_{i+1}^2$

  $\Rightarrow r_0 \leq \gamma_R^{1+2+\cdots+2^{d-1}} \cdot r_d^{2^d} \leq \gamma_R^{2^d} r_d^{2^d} = (\gamma_R r_d)^{2^d} \leq r_{DEC}$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;
  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$
- $r_i \leq \gamma_R \cdot r_{i+1}^2$
  $\Rightarrow r_0 \leq \gamma_R^{1+2+\cdots+2^{d-1}} \cdot r_d^{2^d} \leq \gamma_R^{2^d} r_d^{2^d} = (\gamma_R r_d)^{2^d} \leq r_{DEC}$
  $\Rightarrow d \leq \log\log r_{DEC} - \log\log(\gamma_R \cdot r_{ENC})$

Analyze the depth of permitted circuit

- To analyze r_{ENC}, r_{DEC}, d, let's see how "$+$" and "$\times$" increse the length of vectors

$$|u+v| \leq |u| + |v| \quad ; \quad |u \times v| \leq \gamma_R \cdot |u| \cdot |v|$$

- $+$ vs $\times$: $+$ causes much less expansion than $\times$.
- C : fan-in-2, depth d circuit
- $r_i :=$ max radius at level i of C;

  so $r_d = r_{ENC}$ and $r_0 \leq r_{DEC}$
- $r_i \leq \gamma_R \cdot r_{i+1}^2$

  $\Rightarrow r_0 \leq \gamma_R^{1+2+\cdots+2^{d-1}} \cdot r_d^{2^d} \leq \gamma_R^{2^d} r_d^{2^d} = (\gamma_R r_d)^{2^d} \leq r_{DEC}$

  $\Rightarrow d \leq \log\log r_{DEC} - \log\log(\gamma_R \cdot r_{ENC})$

- Now we have ideas of circuit depth of the permitted circuits. We are going to reduce the decryption circuit. Let's see the encryption scheme more concretely.

- Parameters: Ring $R = \mathbb{Z}[x]/(f(x))$, basis B_I of ideal lattice I, radii r_{DEC} and r_{ENC}, "$+$" and "$\times$" in R.

Encryption scheme (more concrete)

- Parameters: Ring $R = \mathbb{Z}[x]/(f(x))$, basis B_I of ideal lattice I, radii r_{DEC} and r_{ENC}, "$+$" and "$\times$" in R.
- KeyGen: Output Bases (B_{sk}, B_{pk}) of a ideal lattice J, where $I + J = R$. Plaintext space: R/I.

Encryption scheme (more concrete)

- Parameters: Ring $R = \mathbb{Z}[x]/(f(x))$, basis B_I of ideal lattice I, radii r_{DEC} and r_{ENC}, "$+$" and "$\times$" in R.
- KeyGen: Output Bases (B_{sk}, B_{pk}) of a ideal lattice J, where $I + J = R$. Plaintext space: R/I.
- Enc(B_{pk}, π): Set $\pi + i \leftarrow (\pi + I)$.
  Set $\psi \leftarrow \pi + i \mod B_{pk}$

Encryption scheme (more concrete)

- Parameters: Ring $R = \mathbb{Z}[x]/(f(x))$, basis B_I of ideal lattice I, radii r_{DEC} and r_{ENC}, "$+$" and "$\times$" in R.
- KeyGen: Output Bases (B_{sk}, B_{pk}) of a ideal lattice J, where $I + J = R$. Plaintext space: R/I.
- Enc(B_{pk}, π): Set $\pi + i \leftarrow (\pi + I)$.
  Set $\psi \leftarrow \pi + i \mod B_{pk}$
- Dec(B_{sk}, ψ): Output $(\psi \mod B_{sk}) \mod B_I$
  $\rightarrow \pi \mod B_I$

Encryption scheme (more concrete)

- Parameters: Ring $R = \mathbb{Z}[x]/(f(x))$, basis B_I of ideal lattice I, radii r_{DEC} and r_{ENC}, "$+$" and "$\times$" in R.
- KeyGen: Output Bases (B_{sk}, B_{pk}) of a ideal lattice J, where $I + J = R$. Plaintext space: R/I.
- Enc(B_{pk}, π): Set $\pi + i \leftarrow (\pi + I)$.
  Set $\psi \leftarrow \pi + i \mod B_{pk}$
- Dec(B_{sk}, ψ): Output $(\psi \mod B_{sk}) \mod B_I$
  $\rightarrow \pi \mod B_I$
- Add(B_{pk}, ψ_1, ψ_2): Output $\psi \leftarrow \psi_1 + \psi_2 \mod B_{pk}$
- Mult(B_{pk}, ψ_1, ψ_2): Output $\psi \leftarrow \psi_1 \cdot \psi_2 \mod B_{pk}$

Reduce the decryption circuit - Effort 1

- Dec: $\psi \mapsto (\psi \mod B_{sk}) \mod B_I$

Reduce the decryption circuit - Effort 1

- Dec: $\psi \mapsto (\psi \mod B_{sk}) \mod B_I$
  $= \psi - B_{sk} \cdot \lfloor B_{sk}^{-1} \cdot \psi \rceil \mod B_I$

Reduce the decryption circuit - Effort 1

- Dec: $\psi \mapsto (\psi \mod B_{sk}) \mod B_I$
  $= \psi - B_{sk} \cdot \lfloor B_{sk}^{-1} \cdot \psi \rceil \mod B_I$
  $= \psi - \lfloor v_{sk} \times \psi \rceil \mod B_I$
  for some $v_{sk} \in \mathbb{Q}[x]/(f(x))$.

  Remark v_{sk} and $v_{sk} \times \psi \in \mathbb{Q}[x]/(f(x)) \approx \mathbb{Q}^n$

Reduce the decryption circuit - Effort 2

- Recall Dec: $\psi \mapsto \psi - \lfloor v_{sk} \times \psi \rceil \mod B_I$
- Hide v_{sk} in $\{t_1, ..., t_K\}$
- secret $S \subseteq \{1, ..., K\}$ s.t. $v_{sk} = \sum_{i \in S} t_i \mod B_I$. So $v_{sk} \times \psi = \sum_{i \in S}(t_i \times \psi)$

Reduce the decryption circuit - Effort 2

- Recall Dec: $\psi \mapsto \psi - \lfloor v_{sk} \times \psi \rceil \mod B_I$
- Hide v_{sk} in $\{t_1, ..., t_K\}$
- secret $S \subseteq \{1, ..., K\}$ s.t. $v_{sk} = \sum_{i \in S} t_i \mod B_I$. So $v_{sk} \times \psi = \sum_{i \in S}(t_i \times \psi)$
- Encryptor: Compute $t_i \times \psi$ for all $i = 1, ..., K$ and sends those to Decryptor

Reduce the decryption circuit - Effort 2

- Recall Dec: $\psi \mapsto \psi - \lfloor v_{sk} \times \psi \rceil \mod B_I$
- Hide v_{sk} in $\{t_1, ..., t_K\}$
- secret $S \subseteq \{1, ..., K\}$ s.t. $v_{sk} = \sum_{i \in S} t_i \mod B_I$. So $v_{sk} \times \psi = \sum_{i \in S}(t_i \times \psi)$
- Encryptor: Compute $t_i \times \psi$ for all $i = 1, ..., K$ and sends those to Decryptor
- Decryptor: Just **add up** $\sum_{i \in S}(t_i \times \psi) = v_{sk} \times \psi$ without doing multiplication
- Theorem bootstrappable when $|S| \leq \frac{\log(r_{DEC}/m)}{\alpha \cdot 2^c \cdot \log(\gamma_R \cdot r_{ENC})}$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S}(t_i \times \psi) \rceil \mod B_I$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S} (t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, \ldots, a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, \ldots, z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S}(t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, ..., a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, ..., z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

  Proof Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S}(t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, ..., a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, ..., z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

  Proof Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

$$a_1 = \cdots a_1^{(1)} a_1^{(0)} . a_1^{(-1)} a_1^{(-2)} \cdots\cdots$$

$$a_2 = \cdots a_2^{(1)} a_2^{(0)} . a_2^{(-1)} a_2^{(-2)} \cdots\cdots$$

$$\vdots$$

$$a_t = \cdots a_t^{(1)} a_t^{(0)} . a_t^{(-1)} a_t^{(-2)} \cdots\cdots$$

Reduce the decryption circuit - Effort 3

- Recall **Dec**: $\psi \mapsto \psi - \lfloor \sum_{i \in S} (t_i \times \psi) \rceil \mod B_I$
- <u>Lemma</u> For $a_1, \ldots, a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, \ldots, z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

<u>Proof</u> Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

$$
\begin{array}{ll}
a_1 = \cdots a_1^{(1)} a_1^{(0)} . a_1^{(-1)} a_1^{(-2)} \cdots\cdots & z_1 = \boxed{\cdots a_1^{(1)} a_1^{(0)}} \; a_1^{(-1)} \cdots a_1^{(-T)} \\
a_2 = \cdots a_2^{(1)} a_2^{(0)} . a_2^{(-1)} a_2^{(-2)} \cdots\cdots & z_2 = \boxed{\cdots a_2^{(1)} a_2^{(0)}} \; a_2^{(-1)} \cdots a_2^{(-T)} \quad = z_{t+1} \\
\vdots & \vdots \\
a_t = \cdots a_t^{(1)} a_t^{(0)} . a_t^{(-1)} a_t^{(-2)} \cdots\cdots & z_t = \boxed{\cdots a_t^{(1)} a_t^{(0)}} \; a_t^{(-1)} \cdots a_t^{(-T)}
\end{array}
$$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S} (t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, \ldots, a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$,
  $\exists\ z_1, \ldots, z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

  Proof Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

$$
\begin{array}{ll}
a_1 = \cdots a_1^{(1)} a_1^{(0)} . a_1^{(-1)} a_1^{(-2)} \cdots\cdots & z_1 = \boxed{\cdots a_1^{(1)} a_1^{(0)}} \\
a_2 = \cdots a_2^{(1)} a_2^{(0)} . a_2^{(-1)} a_2^{(-2)} \cdots\cdots & z_2 = \boxed{\cdots a_2^{(1)} a_2^{(0)}} \\
\vdots & \vdots \\
a_t = \cdots a_t^{(1)} a_t^{(0)} . a_t^{(-1)} a_t^{(-2)} \cdots\cdots & z_t = \boxed{\cdots a_t^{(1)} a_t^{(0)}}
\end{array}
\quad
\left.\begin{array}{|c|}
a_1^{(-1)} \cdots a_1^{(-T)} \\
a_2^{(-1)} \cdots a_2^{(-T)} \\
\\
a_t^{(-1)} \cdots a_t^{(-T)}
\end{array}\right. = z_{t+1}
$$

  Remark If $x \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$ then $\lfloor x \rceil = \lfloor x + \epsilon \rceil$ for any $|\epsilon| < \frac{1}{4}$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S} (t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, \dots, a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, \dots, z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

  Proof Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

$$
\begin{array}{ll}
a_1 = \cdots a_1^{(1)} a_1^{(0)} . a_1^{(-1)} a_1^{(-2)} \cdots\cdots & z_1 = \boxed{\cdots a_1^{(1)} a_1^{(0)}} \quad a_1^{(-1)} \cdots a_1^{(-T)} \\
a_2 = \cdots a_2^{(1)} a_2^{(0)} . a_2^{(-1)} a_2^{(-2)} \cdots\cdots & z_2 = \boxed{\cdots a_2^{(1)} a_2^{(0)}} \quad a_2^{(-1)} \cdots a_2^{(-T)} \quad = z_{t+1} \\
\vdots & \vdots \\
a_t = \cdots a_t^{(1)} a_t^{(0)} . a_t^{(-1)} a_t^{(-2)} \cdots\cdots & z_t = \boxed{\cdots a_t^{(1)} a_t^{(0)}} \quad a_t^{(-1)} \cdots a_t^{(-T)}
\end{array}
$$

  Remark If $x \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$ then $\lfloor x \rceil = \lfloor x + \epsilon \rceil$ for any $|\epsilon| < \frac{1}{4}$
  $|a_1^{(-T-1)} \cdot 2^{(-T-1)} + a_1^{(-T-2)} \cdot 2^{(-T-2)} + \cdots| < 2^{-T} \leq \frac{1}{4t}$

Reduce the decryption circuit - Effort 3

- Recall Dec: $\psi \mapsto \psi - \lfloor \sum_{i \in S}(t_i \times \psi) \rceil \mod B_I$
- Lemma For $a_1, \ldots, a_t \in \mathbb{Q}$ with $\sum_i a_i \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$, $\exists\ z_1, \ldots, z_{t+1} \in \mathbb{Z}$ s.t. $\lfloor \sum_1^t a_i \rceil = \sum_1^{t+1} z_i$

Proof Let $T := \lceil log_2 t \rceil + 2$. i.e., $2^{-T} \leq \frac{1}{4t}$.

$$a_1 = \cdots a_1^{(1)} a_1^{(0)} . a_1^{(-1)} a_1^{(-2)} \cdots\cdots \qquad z_1 = \boxed{\cdots a_1^{(1)} a_1^{(0)}} \quad a_1^{(-1)} \cdots a_1^{(-T)}$$

$$a_2 = \cdots a_2^{(1)} a_2^{(0)} . a_2^{(-1)} a_2^{(-2)} \cdots\cdots \qquad z_2 = \boxed{\cdots a_2^{(1)} a_2^{(0)}} \quad a_2^{(-1)} \cdots a_2^{(-T)} \quad = z_{t+1}$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$a_t = \cdots a_t^{(1)} a_t^{(0)} . a_t^{(-1)} a_t^{(-2)} \cdots\cdots \qquad z_t = \boxed{\cdots a_t^{(1)} a_t^{(0)}} \quad a_t^{(-1)} \cdots a_t^{(-T)}$$

Remark If $x \mod 1 \in [-\frac{1}{4}, \frac{1}{4}]$ then $\lfloor x \rceil = \lfloor x + \epsilon \rceil$ for any $|\epsilon| < \frac{1}{4}$

$|a_1^{(-T-1)} \cdot 2^{(-T-1)} + a_1^{(-T-2)} \cdot 2^{(-T-2)} + \cdots| < 2^{-T} \leq \frac{1}{4t}$

$$\Rightarrow \sum_{j=-T-1}^{\infty} a_i^{(j)} < t \cdot \frac{1}{4t} = \frac{1}{4}$$

Summary

- "mod an ideal" in a ring is well-defined but "mod a basis of a lattice" causes some problem for long vectors.
- So we need to pull those vector back by bootsrapping.
- In order to be bootstrappable, we analyze the restriction of DEC circuit. And then reduce it.