# Packing Messages and Optimizing Bootstrapping in GSW-FHE


$\boxed{\text{Ryo Hiromasa}^{\dagger}}$ Masayuki Abe[‡] Tatsuaki Okamoto[‡]

[†]Kyoto University [‡]NTT



PKC '15

April 1, 2015

# Fully Homomorphic Encryption (FHE)

$c \leftarrow \mathsf{Enc}(m)$

$f, c$

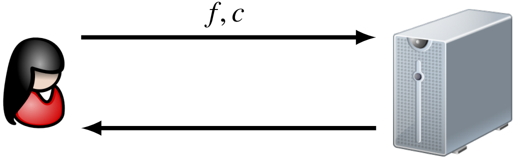
$\hat{c} \leftarrow \mathsf{Eval}(f, c)$

$(f(m) \leftarrow \mathsf{Dec}(\hat{c}))$

$\hat{c}$

# Fully Homomorphic Encryption (FHE)

$c \leftarrow \mathsf{Enc}(m)$ $\quad$ $f, c$ $\quad$ $\hat{c} \leftarrow \mathsf{Eval}(f, c)$



$(f(m) \leftarrow \mathsf{Dec}(\hat{c}))$ $\quad$ $\hat{c}$

- FHE from ...
  - ✓ Ideal lattices: [Gen09]
  - ✓ Integers: [DGHV10]
  - ✓ RLWE: [BV11b]
  - ✓ LWE: [BV11a]
  - ✓ Approx. eigenvector: [GSW13]

# Fully Homomorphic Encryption (FHE)

$c_1 \leftarrow \mathsf{Enc}(m_1)$

$\vdots$

$c_r \leftarrow \mathsf{Enc}(m_r)$

$(f(m_i) \leftarrow \mathsf{Dec}(\hat{c}_i))$

$f, c_1, \ldots, c_r$

$\hat{c}_1, \ldots, \hat{c}_r$

$\hat{c}_1 \leftarrow \mathsf{Eval}(f, c_1)$

$\vdots$

$\hat{c}_r \leftarrow \mathsf{Eval}(f, c_r)$

- FHE from . . .
  - ✓ Ideal lattices: [Gen09]
  - ✓ Integers: [DGHV10]
  - ✓ RLWE: [BV11b]
  - ✓ LWE: [BV11a]
  - ✓ Approx. eigenvector: [GSW13]

# Fully Homomorphic Encryption (FHE)

$c_1 \leftarrow \mathsf{Enc}(m_1)$

$\vdots$

$c_r \leftarrow \mathsf{Enc}(m_r)$

$(f(m_i) \leftarrow \mathsf{Dec}(\hat{c}_i))$

$\xrightarrow{f, c_1, \ldots, c_r}$

$\xleftarrow{\hat{c}_1, \ldots, \hat{c}_r}$

$\hat{c}_1 \leftarrow \mathsf{Eval}(f, c_1)$

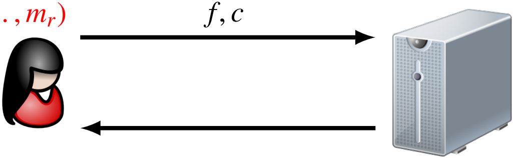
$\vdots$

$\hat{c}_r \leftarrow \mathsf{Eval}(f, c_r)$

- FHE from …
  - ✓ Ideal lattices: [Gen09]
  - ✓ Integers: [DGHV10]
  - ✓ RLWE: [BV11b]
  - ✓ LWE: [BV11a]
  - ✓ Approx. eigenvector: [GSW13]

# Single-Instruction-Multiple-Data (SIMD) FHE

$c \leftarrow \text{Enc}(m_1, \dots, m_r)$ $f, c$ $\hat{c} \leftarrow \text{Eval}(f, c)$



$(f(m_1), \dots, f(m_r) \leftarrow \text{Dec}(\hat{c}))$ $\hat{c}$

- FHE from ...
  - ✓ Ideal lattices: [Gen09]
  - ✓ Integers: [DGHV10]
  - ✓ RLWE: [BV11b]
  - ✓ LWE: [BV11a]
  - ✓ Approx. eigenvector: [GSW13]

# Single-Instruction-Multiple-Data (SIMD) FHE

$c \leftarrow \mathsf{Enc}(m_1, \ldots, m_r)$ $\quad f, c \quad$ $\hat{c} \leftarrow \mathsf{Eval}(f, c)$

$(f(m_1), \ldots, f(m_r) \leftarrow \mathsf{Dec}(\hat{c}))$ $\quad \hat{c}$

- FHE from ...
  - ✓ Ideal lattices: [Gen09]
  - ✓ Integers: [DGHV10]
  - ✓ RLWE: [BV11b]
  - ✓ LWE: [BV11a]
  - ✓ Approx. eigenvector: [GSW13]
- SIMD FHE from ...
  - ✓ Ideal lattices: [SV10]
  - ✓ Integers: [CCKLLTY13]
  - ✓ RLWE: [SV10]
  - ✓ LWE: [BGH13]
  - ✗ Approx. eigenvector: ?

# This Talk

1. SIMD FHE from approximate eigenvector method.

# This Talk

1. SIMD FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.
   - ✓ A natural extension of SIMD FHE.
     Can compute more complicated homomorphic operations.

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.
   - ✓ A natural extension of SIMD FHE.
     Can compute more complicated homomorphic operations.
   - ✗ Requires an additional (but reasonable in FHE literature) assumption for security.

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.
   - ✓ A natural extension of SIMD FHE.
     Can compute more complicated homomorphic operations.
   - ✗ Requires an additional (but reasonable in FHE literature) assumption for security.
2. Application: optimizing bootstrapping of [AP14].

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.
   - ✓ A natural extension of SIMD FHE.
     Can compute more complicated homomorphic operations.
   - ✗ Requires an additional (but reasonable in FHE literature) assumption for security.
2. Application: optimizing bootstrapping of [AP14].
   - ✓ Lattice approximation factor: $O(n^3) \to O(n^{2.5})$.

# This Talk

1. SIMD (Matrix) FHE from approximate eigenvector method.
   - ✓ Simple homomorphic SIMD operations
     (just matrix addition/multiplication).
   - ✓ Supports homomorphic matrix addition and multiplication.
   - ✓ A natural extension of SIMD FHE.
     Can compute more complicated homomorphic operations.
   - ✗ Requires an additional (but reasonable in FHE literature) assumption for security.
2. Application: optimizing bootstrapping of [AP14].
   - ✓ Lattice approximation factor: $O(n^3) \to O(n^{2.5})$.
   - ✓ Allows us to get the best factor $O(n^{1.5+\epsilon})$ without successive dimension-modulus reduction.

# Starting point: Gentry-Sahai-Waters FHE (GSW-FHE)

- Learning with Errors (LWE): $\boldsymbol{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}, \boldsymbol{t} \xleftarrow{U} \mathbb{Z}_q^{n}, \boldsymbol{e} \xleftarrow{R} \chi^m,$

$$\begin{aligned} \boldsymbol{b}^T &= \boldsymbol{t}^T\boldsymbol{A} + \boldsymbol{e}^T \\ \left(\begin{array}{c} \boldsymbol{b}^T \\ \hline \boldsymbol{A} \end{array}\right) &\approx \text{Uniform}(\mathbb{Z}_q^{(n+1)\times m}) \end{aligned}$$

# Starting point: Gentry-Sahai-Waters FHE (GSW-FHE)

- Learning with Errors (LWE): $\boldsymbol{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}, \boldsymbol{t} \xleftarrow{U} \mathbb{Z}_q^n, \boldsymbol{e} \xleftarrow{R} \chi^m,$

$$\begin{aligned} \boldsymbol{b}^T &= \boldsymbol{t}^T\boldsymbol{A} + \boldsymbol{e}^T \\ \left(\frac{\boldsymbol{b}^T}{\boldsymbol{A}}\right) &\approx \text{Uniform}(\mathbb{Z}_q^{(n+1)\times m}) \end{aligned}$$

**GSW-FHE**

- Secret key is $\boldsymbol{s} \in \mathbb{Z}_q^{(n+1)}$.
- $\boldsymbol{G} = (1, 2, \ldots, 2^{\lceil \log q \rceil - 1}) \otimes \boldsymbol{I}$.
- Public key is a LWE matrix $\boldsymbol{B} \in \mathbb{Z}_q^{(n+1)\times m}$ s.t. $\boldsymbol{sB} \approx \boldsymbol{0}$.
- A ciphertext of $m \in \{0, 1\}$ is a matrix $\boldsymbol{C} = \boldsymbol{BR} + m \cdot \boldsymbol{G} \bmod q$ s.t.

$$\boldsymbol{sC} = \text{noise} + m \cdot \boldsymbol{sG}.$$

# Extension to Matrix GSW-FHE: A Natural Extension

## Condition to be Sufficed

For a secret key $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$, a ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t.

$$\boldsymbol{S}\boldsymbol{C} = \text{noise} + \boldsymbol{M}\boldsymbol{S}\boldsymbol{G}.$$

# Extension to Matrix GSW-FHE: A Natural Extension

**Condition to be Sufficed**

For a secret key $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$, a ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t.

$$\boldsymbol{S}\boldsymbol{C} = \text{noise} + \boldsymbol{M}\boldsymbol{S}\boldsymbol{G}.$$

✓ Homomorphic matrix addition: just matrix addition.

$$\boldsymbol{S}(\boldsymbol{C}_1 + \boldsymbol{C}_2) = \text{noise} + (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{S}\boldsymbol{G}.$$

# Extension to Matrix GSW-FHE: A Natural Extension

**Condition to be Sufficed**

For a secret key $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$, a ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t.

$$\boldsymbol{SC} = \text{noise} + \boldsymbol{MSG}.$$

✓ Homomorphic matrix addition: just matrix addition.

$$\boldsymbol{S}(\boldsymbol{C}_1 + \boldsymbol{C}_2) = \text{noise} + (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{SG}.$$

- $\boldsymbol{G}^{-1}(\boldsymbol{X})$: outputs a small matrix $\boldsymbol{X}'$ s.t. $\boldsymbol{GX}' = \boldsymbol{X}$.

# Extension to Matrix GSW-FHE: A Natural Extension

**Condition to be Sufficed**

For a secret key $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$, a ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t.

$$\boldsymbol{SC} = \text{noise} + \boldsymbol{MSG}.$$

✓ Homomorphic matrix addition: just matrix addition.

$$\boldsymbol{S}(\boldsymbol{C}_1 + \boldsymbol{C}_2) = \text{noise} + (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{SG}.$$

- $\boldsymbol{G}^{-1}(\boldsymbol{X})$: outputs a small matrix $\boldsymbol{X}'$ s.t. $\boldsymbol{GX}' = \boldsymbol{X}$.

✓ Homomorphic marix multiplication: computes $\boldsymbol{G}^{-1}(\cdot)$ and matrix multiplication. If we let $\boldsymbol{C}_2' \xleftarrow{R} \boldsymbol{G}^{-1}(\boldsymbol{C}_2)$, then

$$\begin{aligned}\boldsymbol{SC}_1\boldsymbol{C}_2' &= (\text{noise} + \boldsymbol{M}_1\boldsymbol{SG})\boldsymbol{C}_2' \\ &= \text{noise} + \boldsymbol{M}_1\boldsymbol{SC}_2 \\ &= \text{noise} + \boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{SG}.\end{aligned}$$

# Extension to Matrix GSW-FHE: Construction

## Computing Ciphertexts

For a matrix $\boldsymbol{X}$ s.t. $\boldsymbol{S}\boldsymbol{X} = \boldsymbol{M}\boldsymbol{S}$, ciphertexts are required to be of the form:

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{R} + \boldsymbol{X}\boldsymbol{G}.$$

# Extension to Matrix GSW-FHE: Construction

**Computing Ciphertexts**

For a matrix $\boldsymbol{X}$ s.t. $\boldsymbol{S}\boldsymbol{X} = \boldsymbol{M}\boldsymbol{S}$, ciphertexts are required to be of the form:

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{R} + \boldsymbol{X}\boldsymbol{G}.$$

- By construction, $\boldsymbol{S}$ includes an identity matrix: $\boldsymbol{S} = [\boldsymbol{I} \parallel \boldsymbol{S}']$.

# Extension to Matrix GSW-FHE: Construction

**Computing Ciphertexts**

For a matrix $\boldsymbol{X}$ s.t. $\boldsymbol{SX} = \boldsymbol{MS}$, ciphertexts are required to be of the form:

$$\boldsymbol{C} = \boldsymbol{BR} + \boldsymbol{XG}.$$

- By construction, $\boldsymbol{S}$ includes an identity matrix: $\boldsymbol{S} = [\boldsymbol{I} \,\|\, \boldsymbol{S}']$.
- Set $\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = \left( \frac{\boldsymbol{MS}}{\boldsymbol{0}} \right)$$

  - ✗ $\boldsymbol{X}$ can not be computed publicly.
    (In [GSW13], $\boldsymbol{X} = m \cdot \boldsymbol{I}$, so we can publicly compute it.)
  - ✓ In FHE, symmetric→asymmetric is easy [Bar10, Rot11].
  - ✗ It requires to introduce a circular security assumption.

# Extension to Matrix GSW-FHE: Symmetric→Asymmetric

- Let $\boldsymbol{M}_{(i,j)}$ be the matrix with $1$ in $(i, j)$-th position and $0$ in the others.

$$\begin{array}{cc} & \quad\ \ j \\ i & \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

# Extension to Matrix GSW-FHE: Symmetric→Asymmetric

- Let $\boldsymbol{M}_{(i,j)}$ be the matrix with $1$ in $(i, j)$-th position and $0$ in the others.

$$\begin{array}{cc} & \begin{array}{ccc} \phantom{\mathbf{0}} & j & \phantom{\mathbf{0}} \end{array} \\ i & \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

- Publish encryptions of $\boldsymbol{M}_{(i,j)}$ $(= \boldsymbol{P}_{(i,j)})$ as a part of the public key.
  (✗ this enlarges the key size by a #(encrypted bits) factor.)

# Extension to Matrix GSW-FHE: Symmetric→Asymmetric

- Let $M_{(i,j)}$ be the matrix with $1$ in $(i, j)$-th position and $0$ in the others.

$$\begin{array}{cc} & \quad\; j \\ i & \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

- Publish encryptions of $M_{(i,j)}$ ($= P_{(i,j)}$) as a part of the public key.
  (✗ this enlarges the key size by a #(encrypted bits) factor.)

$$P_{(i,j)} = BR_{(i,j)} + \begin{bmatrix} \mathbf{0} \\ s_j^T \\ \mathbf{0} \end{bmatrix} G$$

# Extension to Matrix GSW-FHE: Symmetric→Asymmetric

- Let $\boldsymbol{M}_{(i,j)}$ be the matrix with $1$ in $(i,j)$-th position and $0$ in the others.

$$\begin{array}{cc} & j \\ i & \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

- Publish encryptions of $\boldsymbol{M}_{(i,j)}$ $(= \boldsymbol{P}_{(i,j)})$ as a part of the public key. (✗ this enlarges the key size by a #(encrypted bits) factor.)

$$\boldsymbol{P}_{(i,j)} = \boldsymbol{B}\boldsymbol{R}_{(i,j)} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{s}_j^T \\ \mathbf{0} \end{bmatrix} \boldsymbol{G} \approx \text{Uniform}\ (\Leftarrow \text{✗ circular security})$$

# Extension to Matrix GSW-FHE: Symmetric→Asymmetric

- Let $\boldsymbol{M}_{(i,j)}$ be the matrix with $1$ in $(i, j)$-th position and $0$ in the others.

$$\begin{array}{cc} & j \\ i & \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

- Publish encryptions of $\boldsymbol{M}_{(i,j)}$ $(= \boldsymbol{P}_{(i,j)})$ as a part of the public key. (✗ this enlarges the key size by a #(encrypted bits) factor.)

$$\boldsymbol{P}_{(i,j)} = \boldsymbol{B}\boldsymbol{R}_{(i,j)} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{s}_j^T \\ \mathbf{0} \end{bmatrix} \boldsymbol{G} \approx \text{Uniform}$$ (⇐ ✗ circular security)

- PubEnc$_{\mathsf{pk}}(\boldsymbol{M} \in \{0,1\}^{r \times r})$: Let $\boldsymbol{M}[i, j]$ be the $(i, j)$-th element of $\boldsymbol{M}$.

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{R} + \sum_{(i,j)\in[r]\times[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{P}_{(i,j)}.$$

# Example: Implementing SIMD FHE

## SIMD Ciphertexts

- A ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t. $\boldsymbol{SC} = \text{noise} + \boldsymbol{MSG}$.
- Store $(m_1, \ldots, m_r) \in \{0,1\}^r$ in the diagonal entries of $\boldsymbol{M}$ :

$$\boldsymbol{SC} = \text{noise} + \begin{pmatrix} m_1 & & \\ & \ddots & \\ & & m_r \end{pmatrix} \boldsymbol{SG}.$$

# Example: Implementing SIMD FHE

## SIMD Ciphertexts

- A ciphertext of $\boldsymbol{M} \in \{0,1\}^{r \times r}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t. $\boldsymbol{S}\boldsymbol{C} = \text{noise} + \boldsymbol{M}\boldsymbol{S}\boldsymbol{G}$.
- Store $(m_1, \ldots, m_r) \in \{0,1\}^r$ in the diagonal entries of $\boldsymbol{M}$ :

$$\boldsymbol{S}\boldsymbol{C} = \text{noise} + \begin{pmatrix} m_1 & & \\ & \ddots & \\ & & m_r \end{pmatrix} \boldsymbol{S}\boldsymbol{G}.$$

✓ Homomorphic SIMD addition:

$$\boldsymbol{S}(\boldsymbol{C}_1 + \boldsymbol{C}_2) = \text{noise} + \left( \begin{pmatrix} m_{1,1} & & \\ & \ddots & \\ & & m_{1,r} \end{pmatrix} + \begin{pmatrix} m_{2,1} & & \\ & \ddots & \\ & & m_{2,r} \end{pmatrix} \right) \boldsymbol{S}\boldsymbol{G}.$$

# Example: Implementing SIMD FHE

## SIMD Ciphertexts

- A ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t. $\boldsymbol{SC}$ = noise + $\boldsymbol{MSG}$.
- Store $(m_1, \ldots, m_r) \in \{0,1\}^r$ in the diagonal entries of $\boldsymbol{M}$ :

$$\boldsymbol{SC} = \text{noise} + \begin{pmatrix} m_1 & & \\ & \ddots & \\ & & m_r \end{pmatrix} \boldsymbol{SG}.$$

✓ <u>Homomorphic SIMD multiplication</u>:

$$\boldsymbol{SC}_1\boldsymbol{C}_2' = \text{noise} + \begin{pmatrix} m_{1,1} & & \\ & \ddots & \\ & & m_{1,r} \end{pmatrix} \begin{pmatrix} m_{2,1} & & \\ & \ddots & \\ & & m_{2,r} \end{pmatrix} \boldsymbol{SG}.$$

# Example: Implementing SIMD FHE

## SIMD Ciphertexts

- A ciphertext of $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$ s.t. $\boldsymbol{S}\boldsymbol{C}$ = noise + $\boldsymbol{M}\boldsymbol{S}\boldsymbol{G}$.
- Store $(m_1, \ldots, m_r) \in \{0,1\}^r$ in the diagonal entries of $\boldsymbol{M}$ :

$$\boldsymbol{S}\boldsymbol{C} = \text{noise} + \begin{pmatrix} m_1 & & \\ & \ddots & \\ & & m_r \end{pmatrix} \boldsymbol{S}\boldsymbol{G}.$$

✓ Plaintext-slot permutation: Let $\boldsymbol{\Sigma}$ be a permutation matrix of $\sigma$, $\boldsymbol{W}_\Sigma, \boldsymbol{W}_{\Sigma^T}$ be ciphertexts of $\boldsymbol{\Sigma}, \boldsymbol{\Sigma}^T$.

$$\boldsymbol{S}(\boldsymbol{W}_\Sigma \boldsymbol{C}' \boldsymbol{W}'_{\Sigma^T}) = \text{noise} + \begin{pmatrix} m_{\sigma(1)} & & \\ & \ddots & \\ & & m_{\sigma(r)} \end{pmatrix} \boldsymbol{S}\boldsymbol{G}.$$

# Related Work: Graph-Induced Multilinear Maps [GGH15]

- Recall: a GSW-FHE ciphertext of $m \in \{0, 1\}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{N \times N}$ s.t.

$$(\boldsymbol{s}\boldsymbol{G})\boldsymbol{C} = m \cdot (\boldsymbol{s}\boldsymbol{G}) + \text{noise}.$$

  - ⋆ $\boldsymbol{s}\boldsymbol{G}$ is called <u>approximate eigenvector</u>.
  - ⋆ $m$ is the eigenvalue.

# Related Work: Graph-Induced Multilinear Maps [GGH15]

- Recall: a GSW-FHE ciphertext of $m \in \{0, 1\}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{N \times N}$ s.t.

$$(\boldsymbol{s}\boldsymbol{G})\boldsymbol{C} = m \cdot (\boldsymbol{s}\boldsymbol{G}) + \text{noise}.$$

  - ★ $\boldsymbol{s}\boldsymbol{G}$ is called approximate eigenvector.
  - ★ $m$ is the eigenvalue.

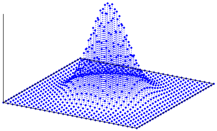
- [GGH15]: approximate eigenvector→approximate eigenspace.
  An encoding of $\boldsymbol{M} \in \mathbb{Z}^{r \times r}$ is a matrix $\boldsymbol{D} \in \mathbb{Z}_q^{N \times N}$ s.t. for
  $\boldsymbol{S} \xleftarrow{U} \mathbb{Z}_q^{r \times N}, \boldsymbol{E} \xleftarrow{R} \chi^{r \times N}$,

$$\boldsymbol{S}\boldsymbol{D} = \boldsymbol{M}\boldsymbol{S} + \boldsymbol{E}.$$

  - ★ $\boldsymbol{S}$ is an approximate eigenspace.

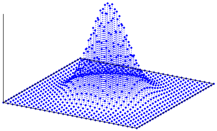
# Related Work: Graph-Induced Multilinear Maps [GGH15]

- PreSamp(trapdoor, $\boldsymbol{A}, z, \sigma$): outputs $\boldsymbol{x}$ s.t. $\boldsymbol{A}\boldsymbol{x} = z$ according to a discrete Gaussian dist. with parameter $\sigma$.



# Related Work: Graph-Induced Multilinear Maps [GGH15]

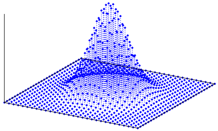
- PreSamp(trapdoor, $A, z, \sigma$): outputs $x$ s.t. $Ax = z$ according to a discrete Gaussian dist. with parameter $\sigma$.



- [GGH15]'s encoding: $D \xleftarrow{R}$ PreSamp(trapdoor, $S$, $MS + E, \sigma$).

# Related Work: Graph-Induced Multilinear Maps [GGH15]

- PreSamp(trapdoor, $\boldsymbol{A}, \boldsymbol{z}, \sigma$): outputs $\boldsymbol{x}$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$ according to a discrete Gaussian dist. with parameter $\sigma$.



- [GGH15]'s encoding: $\boldsymbol{D} \xleftarrow{R} \text{PreSamp}(\text{trapdoor}, \boldsymbol{S}, \boldsymbol{M}\boldsymbol{S} + \boldsymbol{E}, \sigma)$.
- Matrix GSW-FHE: a ciphertext of $\boldsymbol{M} \in \{0,1\}^{r \times r}$ is computed by

$$\boldsymbol{C} \xleftarrow{R} \text{PreSamp}\left(\text{trapdoor}, \boldsymbol{G}, \left(\frac{\boldsymbol{M}\boldsymbol{S}}{\boldsymbol{0}}\right)\boldsymbol{G} + \boldsymbol{B}\boldsymbol{R}, \sigma\right).$$

  - ⋆ We have $(\boldsymbol{S}\boldsymbol{G})\boldsymbol{C} = \boldsymbol{M}(\boldsymbol{S}\boldsymbol{G}) + \text{noise}$.
  - ⋆ $\boldsymbol{S}\boldsymbol{G}$ can be seen as an approximate eigenspace.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

### Recent Developments of Bootstrapping GSW-FHE

✓ In GSW-FHE, the noise grows asymmetrically:
Let $|\text{noise}(c_i)| < B_i$. $|\text{noise}(c_1 \cdot c_2)| < \text{poly}(n) \cdot B_1 + B_2$.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

## Recent Developments of Bootstrapping GSW-FHE

✓ In GSW-FHE, the noise grows asymmetrically:
Let $|\text{noise}(c_i)| < B_i$. $|\text{noise}(c_1 \cdot c_2)| < \text{poly}(n) \cdot B_1 + B_2$.

$\rightarrow$ $|\text{noise}(c_i)| < B \rightarrow |\text{noise}(\underbrace{c_1 \cdot (c_2 \cdot (\cdots (c_{\ell-1} \cdot c_\ell) \cdots))}_{\ell})| < \ell \cdot \text{poly}(n) \cdot B$.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

## Recent Developments of Bootstrapping GSW-FHE

✓ In GSW-FHE, the noise grows asymmetrically:
Let $|\text{noise}(c_i)| < B_i$. $|\text{noise}(c_1 \cdot c_2)| < \text{poly}(n) \cdot B_1 + B_2$.

→ $|\text{noise}(c_i)| < B \rightarrow |\text{noise}(\underbrace{c_1 \cdot (c_2 \cdot (\cdots (c_{\ell-1} \cdot c_\ell) \cdots))}_{\ell})| < \ell \cdot \text{poly}(n) \cdot B$.

✓ To bootstrap with smaller noise, we want to compute the decryption in sequence.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

## Recent Developments of Bootstrapping GSW-FHE

✓ In GSW-FHE, the noise grows asymmetrically:
Let $|\text{noise}(c_i)| < B_i$. $|\text{noise}(c_1 \cdot c_2)| < \text{poly}(n) \cdot B_1 + B_2$.

$\rightarrow$ $|\text{noise}(c_i)| < B \rightarrow |\text{noise}(\underbrace{c_1 \cdot (c_2 \cdot (\cdots (c_{\ell-1} \cdot c_\ell) \cdots)))}_{\ell}| < \ell \cdot \text{poly}(n) \cdot B$.

✓ To bootstrap with smaller noise, we want to compute the decryption in sequence.

- [BV14]: uses the Barrington's theorem.

# Application: Bootstrapping

- Bootstrapping transforms FHE schemes to have unlimited amount of homomorphism.
- This is done by homomorphically evaluating the decryption circuit.

**Recent Developments of Bootstrapping GSW-FHE**

✓ In GSW-FHE, the noise grows asymmetrically:
Let $|\text{noise}(c_i)| < B_i$. $|\text{noise}(c_1 \cdot c_2)| < \text{poly}(n) \cdot B_1 + B_2$.

$\rightarrow$ $|\text{noise}(c_i)| < B \rightarrow |\text{noise}(\underbrace{c_1 \cdot (c_2 \cdot (\cdots (c_{\ell-1} \cdot c_\ell) \cdots))}_{\ell})| < \ell \cdot \text{poly}(n) \cdot B$.

✓ To bootstrap with smaller noise, we want to compute the decryption in sequence.

- [BV14]: uses the Barrington's theorem.
- [AP14]: encodes the decryption to a subset sum.

# Application: Optimizing the Bootstrapping of [AP14]

- The decryption of standard lattice-based FHE is

$$\mathsf{Dec}_{\boldsymbol{s}}(\boldsymbol{c}) = \lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2 = \lfloor \sum_i c_i s_i \rceil_2 = \lfloor \sum_{i : c_i = 1} s_i \rceil_2 .$$

# Application: Optimizing the Bootstrapping of [AP14]

- The decryption of standard lattice-based FHE is

$$\mathsf{Dec}_{\boldsymbol{s}}(\boldsymbol{c}) = \lfloor\langle \boldsymbol{c}, \boldsymbol{s}\rangle\rceil_2 = \lfloor \sum_i c_i s_i \rceil_2 = \lfloor \sum_{i : c_i = 1} s_i \rceil_2 .$$

- Additive groups are isomorphic to groups of cyclic permutations.

# Application: Optimizing the Bootstrapping of [AP14]

- The decryption of standard lattice-based FHE is

$$\mathsf{Dec}_{\boldsymbol{s}}(\boldsymbol{c}) = \lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2 = \lfloor \sum_i c_i s_i \rceil_2 = \lfloor \sum_{i : c_i = 1} s_i \rceil_2 .$$

- Additive groups are isomorphic to groups of cyclic permutations.
- [AP14] encodes $\langle \boldsymbol{c}, \boldsymbol{s} \rangle$ to compositions of cyclic permutations.

# Application: Optimizing the Bootstrapping of [AP14]

- The decryption of standard lattice-based FHE is

$$\mathrm{Dec}_{\boldsymbol{s}}(\boldsymbol{c}) = \lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2 = \lfloor \sum_i c_i s_i \rceil_2 = \lfloor \sum_{i:c_i=1} s_i \rceil_2 .$$

- Additive groups are isomorphic to groups of cyclic permutations.
- [AP14] encodes $\langle \boldsymbol{c}, \boldsymbol{s} \rangle$ to compositions of cyclic permutations.
- The rounding

$$\lfloor x \rceil_2 = \begin{cases} 1 & \text{if } x \approx q/4 \\ 0 & \text{otherwise} \end{cases}$$

consists of checking equality and summing their results.

# Application: Optimizing the Bootstrapping of [AP14]

- The decryption of standard lattice-based FHE is

$$\mathsf{Dec}_{\boldsymbol{s}}(\boldsymbol{c}) = \lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2 = \lfloor \sum_i c_i s_i \rceil_2 = \lfloor \sum_{i:c_i=1} s_i \rceil_2 .$$

- Additive groups are isomorphic to groups of cyclic permutations.
- [AP14] encodes $\langle \boldsymbol{c}, \boldsymbol{s} \rangle$ to compositions of cyclic permutations.
- The rounding

$$\lfloor x \rceil_2 = \begin{cases} 1 & \text{if } x \approx q/4 \\ 0 & \text{otherwise} \end{cases}$$

consists of checking equality and summing their results.

✓ Our optimization represents Dec except for the summation as a sequence of homomorphic multiplications.

- ✓ Lattice approximation factor: $O(n^3) \to O(n^{2.5})$.
- ✓ This allows us to get the best factor $O(n^{1.5+\epsilon})$ without successive dimension-modulus reduction.

# Conclusion

- Our result: SIMD (Matrix) GSW-FHE.
  - ✓ Simple homomorphic SIMD operations (just matrix addition/multiplication).
  - ✓ Supports homomorphic matrix addition and multiplication.
  - ✓ A natural extension of SIMD FHE. Can compute more complicated homomorphic operations.
  - ✗ Requires an additional assumption for security.
  - ⋆ A FHE variant of the recent MMPs [GGH15].
- Application: optimizing [AP14]'s bootstrapping.
  - ✓ Lattice approximation factor: $O(n^3) \to O(n^{2.5})$.
  - ✓ We can get the best factor $O(n^{1.5+\epsilon})$ without successive dimension-modulus reduction.