Efficiently processing complex-valued data in homomorphic encryption

C. Bootland, W. Castryck, I. Iliashenko and F. Vercauteren

Mathcrypt 2018, Aug 19, 2018

Homomorphic encryption

$$\mathbf{ct}(\mathsf{msg}_1) \star \mathbf{ct}(\mathsf{msg}_2) = \mathbf{ct}(\mathsf{msg}_1 * \mathsf{msg}_2)$$

Homomorphic encryption

$$\mathbf{ct}(\mathsf{msg}_1) \star \mathbf{ct}(\mathsf{msg}_2) = \mathbf{ct}(\mathsf{msg}_1 * \mathsf{msg}_2)$$

Most schemes ($\mathtt{BGV}, \mathtt{Bra} - \mathtt{FV}, \mathtt{HEAAN}$) are defined over

$$R_q = \mathbb{Z}[X]/\langle q, X^n + 1\rangle.$$

and based on

Decision Ring-LWE

Sample $a \xleftarrow{\$} R_q$, secret $s \leftarrow \chi_{\mathsf{k}}$ and noise $e \leftarrow \chi_{\mathsf{e}}$. Compute

$$b = a \cdot s + e.$$

Distinguish $(b, a) \in R_q^2$ from a uniformly random pair.

HOMOMORPHIC ENCRYPTION

General approach:

- $\texttt{Encrypt}(\mathsf{msg} \in \mathcal{P} \subseteq R_q) : \mathbf{ct} = (\mathsf{msg}, 0) + (b, a)$
- $\texttt{Evaluate}(\mathbf{ct}, \dots) = \mathbf{ct}'$
- $\texttt{Decrypt}(\mathbf{ct}' \in R_q^2) : \mathbf{ct}'[0] - \mathbf{ct}'[1] \cdot s = \mathsf{msg}' + e' \to \mathsf{msg}'$

$\|e'\| < B$, where B depends on $\mathcal{P}$.

General approach:

- $\texttt{Encrypt}(\mathsf{msg} \in \mathcal{P} \subseteq R_q) : \mathbf{ct} = (\mathsf{msg}, 0) + (b, a)$
- $\texttt{Evaluate}(\mathbf{ct}, \dots) = \mathbf{ct}'$
- $\texttt{Decrypt}(\mathbf{ct}' \in R_q^2) : \mathbf{ct}'[0] - \mathbf{ct}'[1] \cdot s = \mathsf{msg}' + e' \to \mathsf{msg}'$

$\|e'\| < B$, where B depends on $\mathcal{P}$.

Typical choice:

Ciphertext: $R_q = \mathbb{Z}[X]/\left\langle q, X^n + 1\right\rangle$ with $q \simeq poly(n)$

Plaintext: $R_t = \mathbb{Z}[X]/\left\langle t, X^n + 1\right\rangle$ for some $t \geq 2$ and $t \ll q$

Coefficient representatives are taken in $[q/2, q/2)$ and $[t/2, t/2)$, respectively.

DATA ENCODING

$\mathbb{Z} \to R_t$ ($\texttt{Bra} - \texttt{FV}, \texttt{BGV}$):

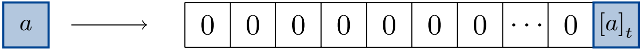
$a \longrightarrow$

0	0	0	0	0	0	0	$\cdots$	0	$[a]_t$

– Bijective as long as $|a| < t/2$.

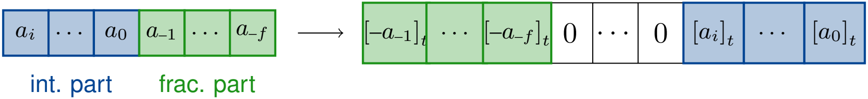
Data encoding

$\mathbb{Z} \to R_t$ ($\mathtt{Bra-FV, BGV}$):



– Bijective as long as $|a| < t/2$.

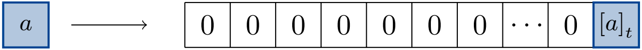
$\mathbb{Q} \to R_t$ ($\mathtt{Bra-FV, BGV}$):



– Bijective as long as plaintext coefficients $< t/2$ and $i + f < n$.

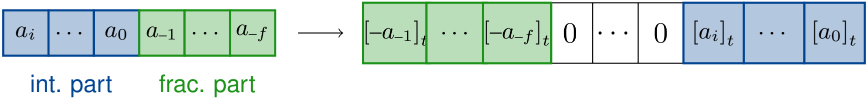
Data encoding

$\mathbb{Z} \to R_t$ ($\texttt{Bra-FV, BGV}$):



– Bijective as long as $|a| < t/2$.

$\mathbb{Q} \to R_t$ ($\texttt{Bra-FV, BGV}$):



– Bijective as long as plaintext coefficients $< t/2$ and $i + f < n$.

$\mathbb{C}^{n/2} \to R$ (HEAAN):

$$(a_1, \ldots, a_{n/2}) \mapsto \left\lfloor FFT^{-1}(a_1, \ldots, a_{n/2}, \overline{a_{n/2}}, \ldots, \overline{a_1})^* \right\rceil$$

* with primitive roots of unity and scaling

– Introduces approximation error.

Polynomial plaintext modulus [CLPX18]

Replace t by $X - b$:

$$R_{X-b} = R/\langle X - b \rangle \cong \mathbb{Z}/\langle b^n + 1 \rangle .$$

POLYNOMIAL PLAINTEXT MODULUS [CLPX18]

Replace t by $X - b$:

$$R_{X-b} = R/\langle X - b\rangle \cong \mathbb{Z}/\langle b^n + 1\rangle .$$

Encoding:

$$\mathbb{Z} \to R_{X-b} : \qquad a \mapsto \text{ small } a(x) \equiv a \mod (X - b)$$

Polynomial plaintext modulus [CLPX18]

Replace t by $X - b$:

$$R_{X-b} = R/\langle X - b\rangle \cong \mathbb{Z}/\langle b^n + 1\rangle .$$

Encoding:

$$\mathbb{Z} \to R_{X-b} : \qquad a \mapsto \text{ small } a(x) \equiv a \mod (X - b)$$

- + Bijective as long as $|a| \leq (b^n + 1)/2$ (often exponential!).
- + Noise depends on b (can be just 2!).
- – Not applicable to BGV: q_i's must be in $\Theta(b^n + 1)$.

Going further: arbitrary plaintext modulus?

$$R_{g(X)} = R/\langle g(X) \rangle \cong ???$$

Going further: arbitrary plaintext modulus?

$$R_{g(X)} = R/\langle g(X) \rangle \cong ???$$

If $g(X) = X^2 + b$,

$$R_{g(X)} \cong \mathbb{Z}[X]/\left\langle b^{n/2} + 1, X^2 + b \right\rangle.$$

$$R_{g(X)} = R/\langle g(X)\rangle \cong ???$$

If $g(X) = X^2 + b$,

$$R_{g(X)} \cong \mathbb{Z}[X]/\left\langle b^{n/2} + 1, X^2 + b\right\rangle .$$

Moreover, if $b \equiv \alpha^2 \mod (b^{n/2} + 1)$, the map

$$i \mapsto \alpha^{-1} \cdot X$$

defines an isomorphism

$$R_{g(X)} \cong \mathbb{Z}[i]/\left\langle b^{n/2} + 1\right\rangle .$$

$$R_{g(X)} = R/\langle g(X)\rangle \cong ???$$

If $g(X) = X^2 + b$,

$$R_{g(X)} \cong \mathbb{Z}[X]/\left\langle b^{n/2} + 1, X^2 + b\right\rangle .$$

Moreover, if $b \equiv \alpha^2 \mod (b^{n/2} + 1)$, the map

$$i \mapsto \alpha^{-1} \cdot X$$

defines an isomorphism

$$R_{g(X)} \cong \mathbb{Z}[i]/\left\langle b^{n/2} + 1\right\rangle .$$

We can encode big Gaussian integers!

Generalization to cyclotomic integers

Use $g(X) = X^m + b$ with $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$\mathbb{Z}[\zeta_{2m}] / \left\langle b^{n/m} + 1 \right\rangle \cong R_{X^m+b}.$$

Use $g(X) = X^m + b$ with $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$\mathbb{Z}[\zeta_{2m}] / \left\langle b^{n/m} + 1 \right\rangle \cong R_{X^m+b}.$$

Encoding:

1. Encode $2m$-th roots of unity:
   $\sum_{i<m} a_i \cdot \zeta_{2m}^i \mapsto \sum_{i<m} a_i \cdot \alpha^{-i} \cdot X^i$

Use $g(X) = X^m + b$ with $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$\mathbb{Z}[\zeta_{2m}] / \left\langle b^{n/m} + 1 \right\rangle \cong R_{X^m+b}.$$

Encoding:

1. Encode $2m$-th roots of unity:
$\sum_{i<m} a_i \cdot \zeta_{2m}^i \mapsto \sum_{i<m} a_i \cdot \alpha^{-i} \cdot X^i$
2. Expand coefficients in base b:
$\sum_{i<m} a_i \cdot \alpha^{-i} X^i \mapsto \sum_{i<m} \sum_{j<n/m} c_{ij} b^j X^i$

Generalization to cyclotomic integers

Use $g(X) = X^m + b$ with $b \equiv \alpha^m \mod (b^{n/m}+1)$, then

$$\mathbb{Z}[\zeta_{2m}] / \left\langle b^{n/m} + 1 \right\rangle \cong R_{X^m+b}.$$

Encoding:

1. Encode $2m$-th roots of unity:
   $\sum_{i<m} a_i \cdot \zeta_{2m}^i \mapsto \sum_{i<m} a_i \cdot \alpha^{-i} \cdot X^i$
2. Expand coefficients in base b:
   $\sum_{i<m} a_i \cdot \alpha^{-i} X^i \mapsto \sum_{i<m} \sum_{j<n/m} c_{ij} b^j X^i$
3. Use $b \equiv -X^m \mod (X^m + b)$
   $\sum_i \sum_j c_{ij} b^j X^i \mapsto \sum_i \sum_j c_{ij} (-X)^{mj} X^i$

Use $g(X) = X^m + b$ with $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$\mathbb{Z}[\zeta_{2m}] / \left\langle b^{n/m} + 1 \right\rangle \cong R_{X^m+b}.$$

Encoding:

1. Encode $2m$-th roots of unity:
   $\sum_{i<m} a_i \cdot \zeta_{2m}^i \mapsto \sum_{i<m} a_i \cdot \alpha^{-i} \cdot X^i$
2. Expand coefficients in base b:
   $\sum_{i<m} a_i \cdot \alpha^{-i} X^i \mapsto \sum_{i<m} \sum_{j<n/m} c_{ij} b^j X^i$
3. Use $b \equiv -X^m \mod (X^m + b)$
   $\sum_i \sum_j c_{ij} b^j X^i \mapsto \sum_i \sum_j c_{ij} (-X)^{mj} X^i$

As a result, $|c_{ij}| \leq \lfloor (b+1)/2 \rfloor$.

Decoding:

1. Reduction modulo $X^m + b$
   $\sum_{i<n} c_i X^i \mapsto \sum_{i<n} c_i X^i \mod (X^m + b)$

Generalization to Cyclotomic Integers

Decoding:

1. Reduction modulo $X^m + b$
   $\sum_{i<n} c_i X^i \mapsto \sum_{i<n} c_i X^i \mod (X^m + b)$

2. Decode $2m$-th roots of unity:
   $\sum_{i<m} c'_i X^i \mapsto \sum_{i<m} c'_i \alpha^i \zeta_{2m}^i$

Generalization to Cyclotomic Integers

Decoding:

1. Reduction modulo $X^m + b$
   $$\sum_{i<n} c_i X^i \mapsto \sum_{i<n} c_i X^i \mod (X^m + b)$$
2. Decode $2m$-th roots of unity:
   $$\sum_{i<m} c'_i X^i \mapsto \sum_{i<m} c'_i \alpha^i \zeta_{2m}^i$$
3. Take a representative of $c'_i \alpha^i$ in $\left[- \left\lfloor b^{n/m}/2 \right\rfloor, \left\lceil b^{n/m}/2 \right\rceil\right]$

How to choose b?

- If $b = 2^{m/2}$, then $\alpha \equiv b^{n/4m}(b^{n/2m} - 1) \mod (b^{n/m} + 1)$.

How to choose b?

- If $b = 2^{m/2}$, then $\alpha \equiv b^{n/4m}(b^{n/2m} - 1) \mod (b^{n/m} + 1)$.
- If an odd b satisfies $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$b \equiv \pm 1 \mod 4m.$$

How to choose b?

- If $b = 2^{m/2}$, then $\alpha \equiv b^{n/4m}(b^{n/2m} - 1) \mod (b^{n/m} + 1)$.
- If an odd b satisfies $b \equiv \alpha^m \mod (b^{n/m} + 1)$, then

$$b \equiv \pm 1 \mod 4m.$$

Finding b requires factorization of generalized Fermat numbers.

How to encode arbitrary complex numbers?

$$\mathbb{Z}[\zeta_{2m}] \to R_{X^m+b}$$

How to encode arbitrary complex numbers?

$$\mathbb{C} \xrightarrow{?} \mathbb{Z}[\zeta_{2m}] \to R_{X^m+b}$$

How to encode arbitrary complex numbers?

$$\mathbb{C} \xrightarrow{?} \mathbb{Z}[\zeta_{2m}] \to R_{X^m+b}$$

- Fractional encoding [CLPX18]
  - approximates $\mathbb{C} \to \mathcal{P} + i \cdot \mathcal{P}$, where $\mathcal{P} \subset \mathbb{Q}$
  - encodes elements of $\mathcal{P}$ to $\mathbb{Z}_{b^{n/2}+1}$ (i.e. $m = 2$)
- Integer coefficient approximation [CSV17]
  - solves a CVP instance in the lattice $\mathbb{Z}[\zeta_{2m}]$

FRACTIONAL ENCODING

- Encoding
  1. Choose $\mathcal{P} = \left\{c + \frac{d}{b^{n/4}}\right\} \subset \mathbb{Q}$ with $c, d \in \mathbb{Z}$
     - $|c|, |d| \leq \frac{b^{n/4}-1}{2}$, for even b
     - $|c| \leq \frac{(b^{n/4-1}-1)b}{2(b-1)}; |d| \leq \frac{(b^{n/4}-1)b}{2(b-1)}$, for odd b

- Encoding
  1. Choose $\mathcal{P} = \left\{c + \frac{d}{b^{n/4}}\right\} \subset \mathbb{Q}$ with $c, d \in \mathbb{Z}$
     - $|c|, |d| \leq \frac{b^{n/4}-1}{2}$, for even b
     - $|c| \leq \frac{(b^{n/4-1}-1)b}{2(b-1)}$; $|d| \leq \frac{(b^{n/4}-1)b}{2(b-1)}$, for odd b
  2. Approximate $z \in \mathbb{C}$ to some $\frac{x_0}{y_0} + i \cdot \frac{x_1}{y_1}$ with $\frac{x_0}{y_0}, \frac{x_1}{y_1} \in \mathcal{P}$.

Fractional encoding

- Encoding
  1. Choose $\mathcal{P} = \left\{c + \frac{d}{b^{n/4}}\right\} \subset \mathbb{Q}$ with $c, d \in \mathbb{Z}$
     - $|c|, |d| \leq \frac{b^{n/4}-1}{2}$, for even b
     - $|c| \leq \frac{(b^{n/4-1}-1)b}{2(b-1)}; |d| \leq \frac{(b^{n/4}-1)b}{2(b-1)}$, for odd b
  2. Approximate $z \in \mathbb{C}$ to some $\frac{x_0}{y_0} + i \cdot \frac{x_1}{y_1}$ with $\frac{x_0}{y_0}, \frac{x_1}{y_1} \in \mathcal{P}$.
  3. Encode

$$\frac{x_0}{y_0} + i \cdot \frac{x_1}{y_1} \mapsto \left[\frac{x_0}{y_0}\right]_{b^{n/2}+1} + i \cdot \left[\frac{x_1}{y_1}\right]_{b^{n/2}+1} \in \mathbb{Z}[i] / \left\langle b^{n/2} + 1 \right\rangle .$$

FRACTIONAL ENCODING

- Encoding
  1. Choose $\mathcal{P} = \left\{c + \frac{d}{b^{n/4}}\right\} \subset \mathbb{Q}$ with $c, d \in \mathbb{Z}$
     - $|c|, |d| \leq \frac{b^{n/4}-1}{2}$, for even b
     - $|c| \leq \frac{(b^{n/4-1}-1)b}{2(b-1)}; |d| \leq \frac{(b^{n/4}-1)b}{2(b-1)}$, for odd b
  2. Approximate $z \in \mathbb{C}$ to some $\frac{x_0}{y_0} + i \cdot \frac{x_1}{y_1}$ with $\frac{x_0}{y_0}, \frac{x_1}{y_1} \in \mathcal{P}$.
  3. Encode

$$\frac{x_0}{y_0} + i \cdot \frac{x_1}{y_1} \mapsto \left[\frac{x_0}{y_0}\right]_{b^{n/2}+1} + i \cdot \left[\frac{x_1}{y_1}\right]_{b^{n/2}+1} \in \mathbb{Z}[i] / \left\langle b^{n/2} + 1 \right\rangle .$$

- Decoding

$$x + i \cdot y \mapsto \begin{cases} \frac{\left[x \cdot b^{n/4}\right]_{b^{n/2}+1} + i \cdot \left[y \cdot b^{n/4}\right]_{b^{n/2}+1}}{b^{n/4}}, & \text{for odd } b \\ \frac{\left[x \cdot b^{n/4-1}\right]_{b^{n/2}+1} + i \cdot \left[y \cdot b^{n/4-1}\right]_{b^{n/2}+1}}{b^{n/4-1}}, & \text{for even } b \end{cases}$$

Encoding

1. For a given $z \in \mathbb{C}$, choose constants $C, T > 0$ and compute $a_i = \lceil \Re(C\zeta_{2m}^i) \rfloor, b_i = \lceil \Im(C\zeta_{2m}^i) \rfloor$

Encoding

1. For a given $z \in \mathbb{C}$, choose constants $C, T > 0$ and compute $a_i = \lceil \Re(C\zeta_{2m}^i) \rfloor, b_i = \lceil \Im(C\zeta_{2m}^i) \rfloor$

2. Solve:

SVP in the lattice given by

$$\begin{pmatrix} & & a_0 & b_0 & 0 \\ & I_m & \vdots & \vdots & \vdots \\ & & a_{m-1} & b_{m-1} & 0 \\ 0 & \dots & \lceil \Re(Cz) \rfloor & \lceil \Im(Cz) \rfloor & T \end{pmatrix}$$

OR CVP in the lattice given by

$$\begin{pmatrix} & a_0 & b_0 \\ I_m & \vdots & \vdots \\ & a_{m-1} & b_{m-1} \end{pmatrix}$$

with a target vector:
$(0, \dots, 0, \lceil \Re(Cz) \rfloor, \lceil \Im(Cz) \rfloor)$

Encoding

1. For a given $z \in \mathbb{C}$, choose constants $C, T > 0$ and compute $a_i = \lceil \Re(C\zeta_{2m}^i) \rfloor, b_i = \lceil \Im(C\zeta_{2m}^i) \rfloor$

2. Solve:

SVP in the lattice given by

$$\begin{pmatrix} & & a_0 & b_0 & 0 \\ & I_m & \vdots & \vdots & \vdots \\ & & a_{m-1} & b_{m-1} & 0 \\ 0 & \dots & \lceil \Re(Cz) \rfloor & \lceil \Im(Cz) \rfloor & T \end{pmatrix}$$

OR CVP in the lattice given by

$$\begin{pmatrix} & a_0 & b_0 \\ I_m & \vdots & \vdots \\ & a_{m-1} & b_{m-1} \end{pmatrix}$$

with a target vector:
$(0, \dots, 0, \lceil \Re(Cz) \rfloor, \lceil \Im(Cz) \rfloor)$

3. Use a SVP solution $\pm(z_0, \dots, z_{m-1}, \dots, -T)$ or a CVP solution $(z_0, \dots, z_{m-1}, \sum z_i a_i, \sum z_i b_i)$ and output

$$\sum_{i=1}^{m-1} z_i \zeta_{2m}^i \simeq z.$$

Back to HE

$$\mathbb{C} \to \mathbb{Z}[\zeta_{2m}] \to R_{X^m+b}$$

Back to HE

$$\mathbb{C} \to \mathbb{Z}[\zeta_{2m}] \to R_{X^m+b} \xrightarrow{?} R_q$$

Adapting the Bra-FV scheme

- Parameters
  - $\Delta = \left\lfloor \frac{q}{t} \right\rfloor$
  - the decomposition base w, the error distribution χ_{e} and the key distribution χ_{k}
- $\texttt{KeyGen}()$
  - $\texttt{sk} = (1, s)$ with $s \leftarrow \chi_{\mathsf{k}}$
  - $\texttt{pk} = \left([-(as + e)]_q, a \right)$ with $a \xleftarrow{\$} R_q, e \leftarrow \chi_{\mathsf{e}}$
  - $\texttt{evk} = \{ ([-(a_i s + e_i)]_q + w^i s^2, a_i) \}_i$ for $a_i \xleftarrow{\$} R_q, e_i \leftarrow \chi_{\mathsf{e}}$.
- $\texttt{Encrypt}\left(\mathsf{msg} \in R_t\right)$
  - $u \leftarrow \chi_{\mathsf{k}}, e_0, e_1 \leftarrow \chi_{\mathsf{e}}$
  - $\mathbf{ct} = \left([\Delta \cdot \mathsf{msg} + u \cdot \texttt{pk}[0] + e_0]_q, [u \cdot \texttt{pk}[1] + e_1]_q \right)$
- $\texttt{Decrypt}\left(\mathbf{ct} \in R_q^2\right)$

$$\left[\left\lfloor \frac{t}{q} \cdot [\mathbf{ct}[0] + \mathbf{ct}[1] \cdot s]_q \right\rceil \right]_t = \mathsf{msg}'$$

Adapting the Bra-FV scheme

- Parameters
  - $\Delta_b = \left\lfloor \frac{q}{X^m+b} \mod (X^n+1) \right\rceil = \left\lfloor -\frac{q}{b^{n/m}+1} \sum_{i=1}^{n/m} (-b)^{i-1} X^{n-im} \right\rceil$
  - the decomposition base w, the error distribution χ_{e} and the key distribution χ_{k}
- $\texttt{KeyGen}()$
  - $\mathsf{sk} = (1, s)$ with $s \leftarrow \chi_{\mathsf{k}}$
  - $\mathsf{pk} = ([-(as+e)]_q, a)$ with $a \xleftarrow{\$} R_q, e \leftarrow \chi_{\mathsf{e}}$
  - $\mathtt{evk} = \{([-(a_i s + e_i)]_q + w^i s^2, a_i)\}_i$ for $a_i \xleftarrow{\$} R_q, e_i \leftarrow \chi_{\mathsf{e}}$.
- $\texttt{Encrypt}\left(\mathsf{msg} \in R_{X^m+b}\right)$
  - $u \leftarrow \chi_{\mathsf{k}}, e_0, e_1 \leftarrow \chi_{\mathsf{e}}$
  - $\mathbf{ct} = \left([\Delta_b \cdot \mathsf{msg} + u \cdot \mathsf{pk}[0] + e_0]_q, [u \cdot \mathsf{pk}[1] + e_1]_q\right)$
- $\texttt{Decrypt}\left(\mathbf{ct} \in R_q^2\right)$

$$\left[\left\lfloor \frac{X^m+b}{q} \cdot [\mathbf{ct}[0] + \mathbf{ct}[1] \cdot s]_q \right\rceil\right]_{X^m+b} = \mathsf{msg}'$$

Noise growth

- Fresh encryption

$$\|v_{\texttt{Mul}}\|^{\text{can}} \leq \frac{t}{q}\left(\frac{\sqrt{3n}}{2}tn + \sigma\left(32\sqrt{2/3}n + 6\sqrt{n}\right)\right)$$

- After multiplication (of ciphertexts with noise v_1, v_2)

$$\begin{aligned}\|v_{\texttt{Mul}}\|^{\text{can}} \leq t\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n\right)\left(\|v_1\|^{\text{can}} + \|v_2\|^{\text{can}}\right) \\ + 3\|v_1\|^{\text{can}}\|v_2\|^{\text{can}} \\ + \frac{t}{q}\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n + \frac{8}{\sqrt{3}}(\ell+1)\sigma wn + \frac{40}{3\sqrt{3}}n\sqrt{n}\right).\end{aligned}$$

NOISE GROWTH

- Fresh encryption

$$\|v_{\mathtt{Mul}}\|^{\mathsf{can}} \leq \frac{b+1}{q}\left(\frac{\sqrt{3n}}{2}bn + \sigma\left(32\sqrt{2/3}n + 6\sqrt{n}\right)\right)$$

- After multiplication (of ciphertexts with noise v_1, v_2)

$$\begin{aligned}\|v_{\mathtt{Mul}}\|^{\mathsf{can}} \leq (b+1)\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n\right)\left(\|v_1\|^{\mathsf{can}} + \|v_2\|^{\mathsf{can}}\right) \\ + 3\|v_1\|^{\mathsf{can}}\|v_2\|^{\mathsf{can}} \\ + \frac{b+1}{q}\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n + \frac{8}{\sqrt{3}}(\ell+1)\sigma w n + \frac{40}{3\sqrt{3}}n\sqrt{n}\right).\end{aligned}$$

- Fresh encryption

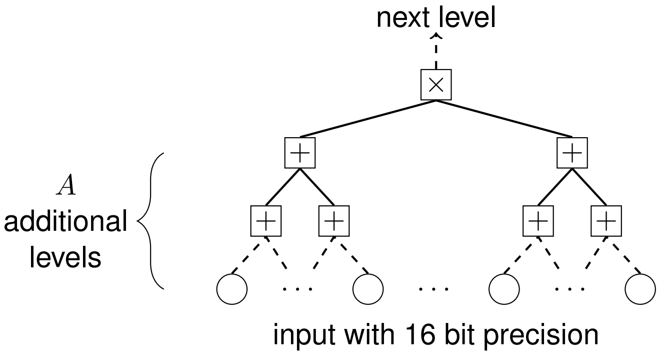
$$\|v_{\mathtt{Mul}}\|^{\mathsf{can}} \leq \frac{b+1}{q}\left(\frac{\sqrt{3n}}{2}bn + \sigma\left(32\sqrt{2/3}n + 6\sqrt{n}\right)\right)$$

- After multiplication (of ciphertexts with noise v_1, v_2)

$$\begin{aligned}\|v_{\mathtt{Mul}}\|^{\mathsf{can}} \leq (b+1)\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n\right)\left(\|v_1\|^{\mathsf{can}} + \|v_2\|^{\mathsf{can}}\right)\\ + 3\|v_1\|^{\mathsf{can}}\|v_2\|^{\mathsf{can}}\\ + \frac{b+1}{q}\left(\sqrt{3n} + \frac{8\sqrt{2}}{3}n + \frac{8}{\sqrt{3}}(\ell+1)\sigma w n + \frac{40}{3\sqrt{3}}n\sqrt{n}\right).\end{aligned}$$

In practice, $b \ll t$!

Regular circuits consisting of the following levels:



	n $\log q$	4096 116			8192 226			16384 435			32768 889		
U	A	0	3	10	0	3	10	0	3	10	0	3	10
2^{32}	D_O	0	0	0	1	1	1	1	1	1	2	2	2
	D_M	5	5	4	9	9	7	12	11	10	14	14	13
	D_I	5	5	4	8	8	7	11	10	10	13	13	12
	D_F	5	5	4	9	8	7	11	10	10	13	13	12
2^{64}	D_O	—	—	—	0	0	0	1	1	1	2	1	1
	D_M	5	5	4	8	8	7	11	11	10	13	13	12
	D_I	5	4	4	8	7	7	10	10	9	12	12	12
	D_F	5	5	4	8	8	7	10	10	9	12	12	12

Real and imaginary parts of input data are bounded by U.

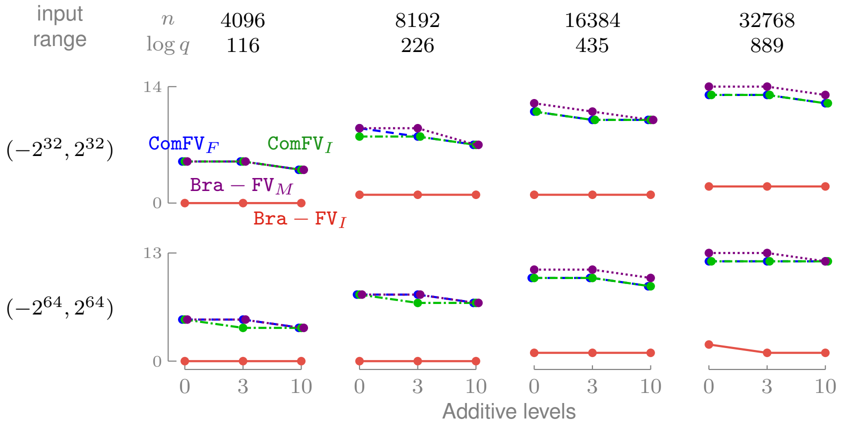
D_O: original `Bra` – `FV` with integer coefficient approximation in $\mathbb{Z}[\zeta_8]$.

D_M: `Bra` – `FV` with $t = X - b$ and separately encrypted real and imaginary parts of complex input. Needs twice more memory and additional operations!

D_I: `ComFV` with integer coefficient approximation in $\mathbb{Z}[\zeta_8]$.

D_F: `ComFV` with fractional encoding.

Regular circuit depth



- Bra − FV with R_{X-b} and separately encrypted real and imaginary parts (Bra − FV$_M$). Needs twice more memory and additional operations!
- Bra − FV with integer coefficient approximation (Bra − FV$_I$).
- ComFV with integer coefficient approximation (ComFV$_I$).
- ComFV with fractional encoding (ComFV$_F$).

CONCLUSION

+ **New encoding method** of complex numbers for FHE/SHE schemes.
+ **New plaintext space** allowing to encode **big complex numbers**.
+ Much **slower noise growth** in comparison to existing native $\mathtt{Bra} - \mathtt{FV}$ encodings of complex numbers.
+ Almost the **same depth** but **smaller memory usage** and **faster** complex number **operations** in comparison to "High-Precision" method [CLPX18].

Conclusion

+ **New encoding method** of complex numbers for FHE/SHE schemes.
+ **New plaintext space** allowing to encode **big complex numbers**.
+ Much **slower noise growth** in comparison to existing native $\mathtt{Bra}-\mathtt{FV}$ encodings of complex numbers.
+ Almost the **same depth** but **smaller memory usage** and **faster** complex number **operations** in comparison to "High-Precision" method [CLPX18].

- **Hard** to find an optimal b.
- **Limited packing** functionality.

Conclusion

+ **New encoding method** of complex numbers for FHE/SHE schemes.
+ **New plaintext space** allowing to encode **big complex numbers**.
+ Much **slower noise growth** in comparison to existing native $\mathtt{Bra-FV}$ encodings of complex numbers.
+ Almost the **same depth** but **smaller memory usage** and **faster** complex number **operations** in comparison to "High-Precision" method [CLPX18].

- **Hard** to find an optimal b.
- **Limited packing** functionality.

? Better methods to approximate complex numbers by cyclotomic integers.

? Polynomial ciphertext modulus

THANK YOU.

QUESTIONS?