High-Precision Exact FHE Made Simple, General, and Fast

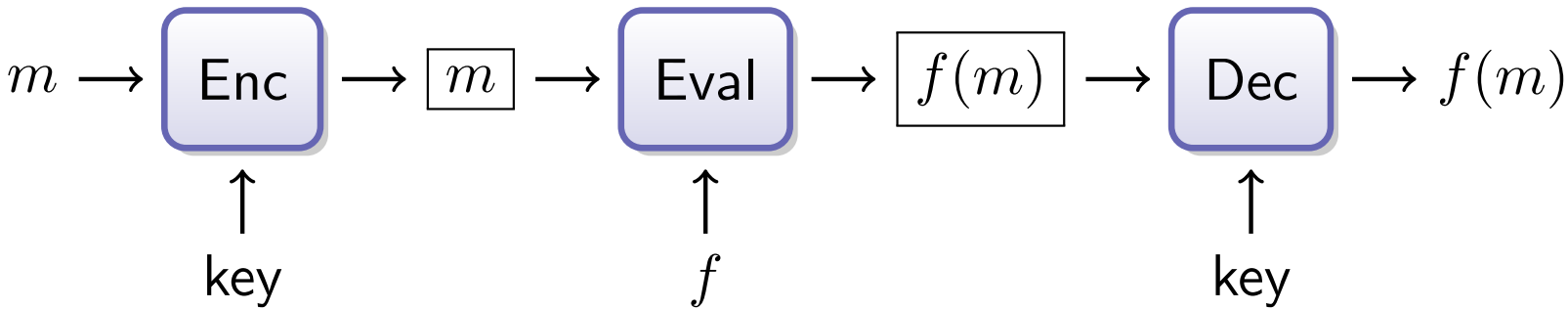
Chris Peikert
University of Michigan
Fhenix

Doron Zarchy
Fhenix

Guy Zyskind
Fhenix
University of Miami

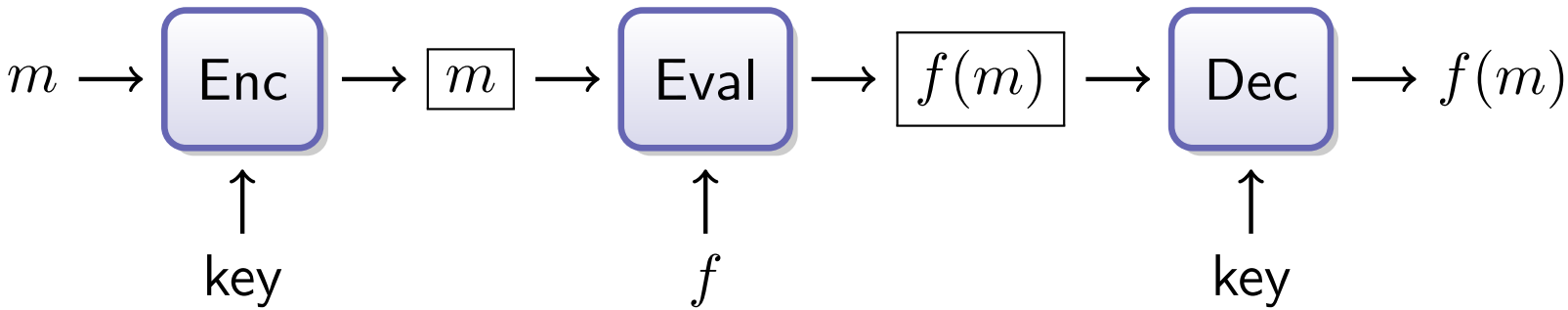
CRYPTO
20 Aug 2026

Fully Homomorphic Encryption [RAD'78,Gentry'09,...]



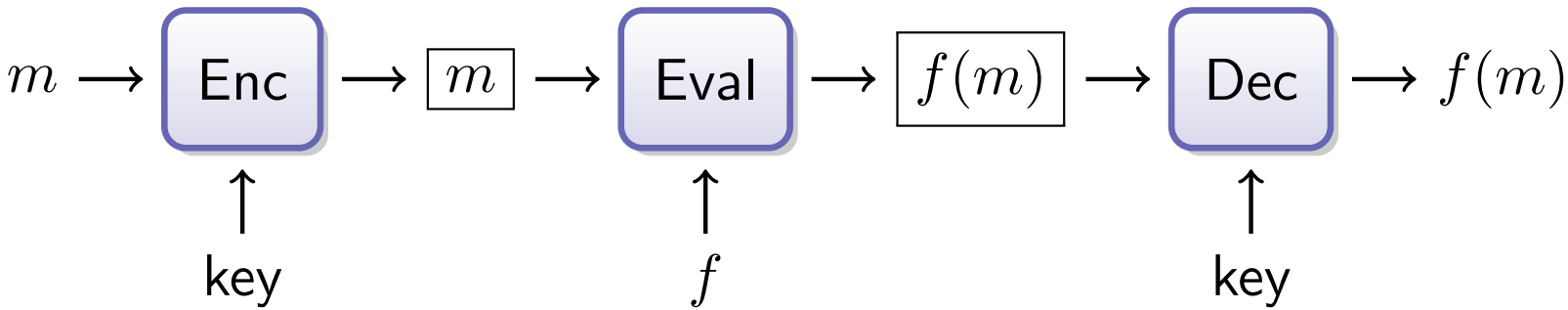
- Compute on data without seeing it

Fully Homomorphic Encryption [RAD'78,Gentry'09,...]



- Compute on data without seeing it
- For exact arithmetic, plaintexts live in a finite ring like $\mathbb{Z}_p$

Fully Homomorphic Encryption [RAD'78,Gentry'09,...]



- Compute on data without seeing it
- For exact arithmetic, plaintexts live in a finite ring like $\mathbb{Z}_p$
- High precision—i.e., large p—is a major pain point

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic
- BGV/BFV: each multiplication enlarges the error rate by $\gg p$ factor

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic
- BGV/BFV: each multiplication enlarges the error rate by $\gg p$ factor
- So depth-h computation needs initial error rate $\alpha < p^{-h}$

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic
- BGV/BFV: each multiplication enlarges the error rate by $\gg p$ factor
- So depth-h computation needs initial error rate $\alpha < p^{-h}$
- Security requires LWE dimension and integer precision $\Omega(\log(1/\alpha))$

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic
- BGV/BFV: each multiplication enlarges the error rate by $\gg p$ factor
- So depth-h computation needs initial error rate $\alpha < p^{-h}$
- Security requires LWE dimension and integer precision $\Omega(\log(1/\alpha))$

Resulting cost: $\Omega(h^2 \log^2 p)$ ciphertext sizes & runtimes

High Precision Is Expensive

- Example apps: encrypted 64-bit ALU, large finite-field arithmetic
- BGV/BFV: each multiplication enlarges the error rate by $\gg p$ factor
- So depth-h computation needs initial error rate $\alpha < p^{-h}$
- Security requires LWE dimension and integer precision $\Omega(\log(1/\alpha))$

Resulting cost: $\Omega(h^2 \log^2 p)$ ciphertext sizes & runtimes

For $p \geq 2^{64}$, this is a substantial concrete expense!

A Fragmented Landscape

[GC'14,CLPX'18,GV'25,BK'25,K'25,K'26,CPL'26,BFGGMS'26,...]

Approach	**Main advantage**	**Main compromise(s)**
Bitwise TFHE	Arbitrary p	$\Omega(\log^2 p)$ mult, slow add
Special modulus ideals	Simple, fast	Rare moduli, coupled rings
CKKS-based methods	Arbitrary p	Complex, mult $\supseteq$ bootstr
Special non-cyclo ring	Fast for $p = 2^k$	no SIMD, slow bootstrap

Our Contribution: **Decomposed** BFV(/BGV/GSW)

- **Simple and General:** any modulus p and any ring (or plain LWE)

Our Contribution: **Decomposed** BFV(/BGV/GSW)

- **Simple and General:** any modulus p and any ring (or plain LWE)
- **Compatible:** retain standard rings/arithmetic, homomorphic ops, packing/SIMD, automorphisms, and bootstrapping options (+ self-host)

Our Contribution: **Decomposed** BFV(/BGV/GSW)

- **Simple and General:** any modulus p and any ring (or plain LWE)
- **Compatible:** retain standard rings/arithmetic, homomorphic ops, packing/SIMD, automorphisms, and bootstrapping options (+ self-host)
- **Efficient:** error rates and growth, ciphertext sizes, and runtimes can match classic BFV/BGV with an exponentially smaller modulus $\approx \log p$

Our Contribution: **Decomposed** BFV(/BGV/GSW)

- **Simple and General:** any modulus p and any ring (or plain LWE)
- **Compatible:** retain standard rings/arithmetic, homomorphic ops, packing/SIMD, automorphisms, and bootstrapping options (+ self-host)
- **Efficient:** error rates and growth, ciphertext sizes, and runtimes can match classic BFV/BGV with an exponentially smaller modulus $\approx \log p$

Key Idea

Represent $\mathbb{Z}_p$ isomorphically in higher dimension, w/ much smaller representatives

Our Contribution: **Decomposed** BFV(/BGV/GSW)

- **Simple and General:** any modulus p and any ring (or plain LWE)
- **Compatible:** retain standard rings/arithmetic, homomorphic ops, packing/SIMD, automorphisms, and bootstrapping options (+ self-host)
- **Efficient:** error rates and growth, ciphertext sizes, and runtimes can match classic BFV/BGV with an exponentially smaller modulus $\approx \log p$

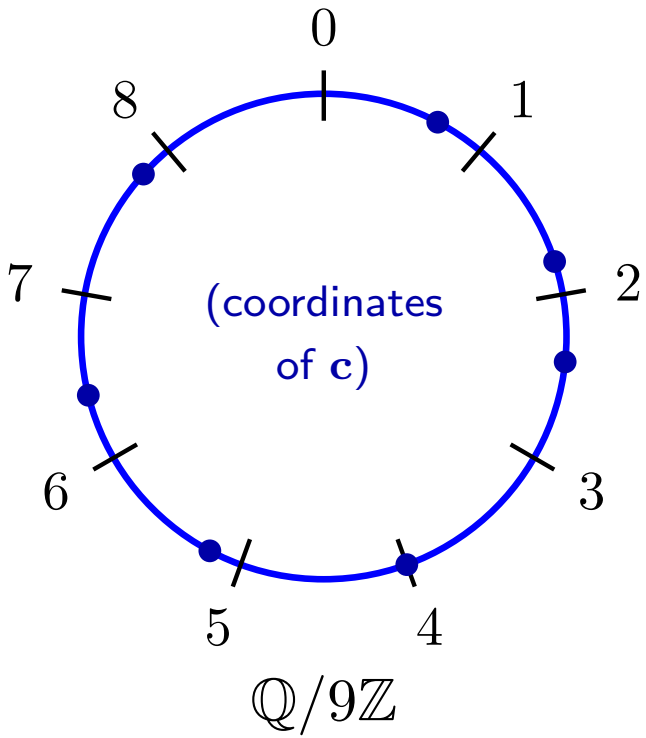
Key Idea

Represent $\mathbb{Z}_p$ isomorphically in higher dimension, w/ much smaller representatives

	$\log_2 b$	$\log_2$(growth)	depth	CPU add	CPU/GPU mul
BFV $(d = 1)$	64	75	2	< 0.05	$11.4/0.26$
dBFV $(d = 4)$	16	31	6	< 0.10	$66.7/1.30$
dBFV $(d = 8)$	8	25	8	0.20	$196.3/2.82$
TFHE-rs	2	–	–	596.7	$16084/–$

$p = 2^{64}$; BFV/dBFV use $n = 2^{13}$; times in ms on one CPU thread / one A100.
Preliminary, unoptimized dBFV implementation; TFHE-rs uses $n = 2^{11}$.

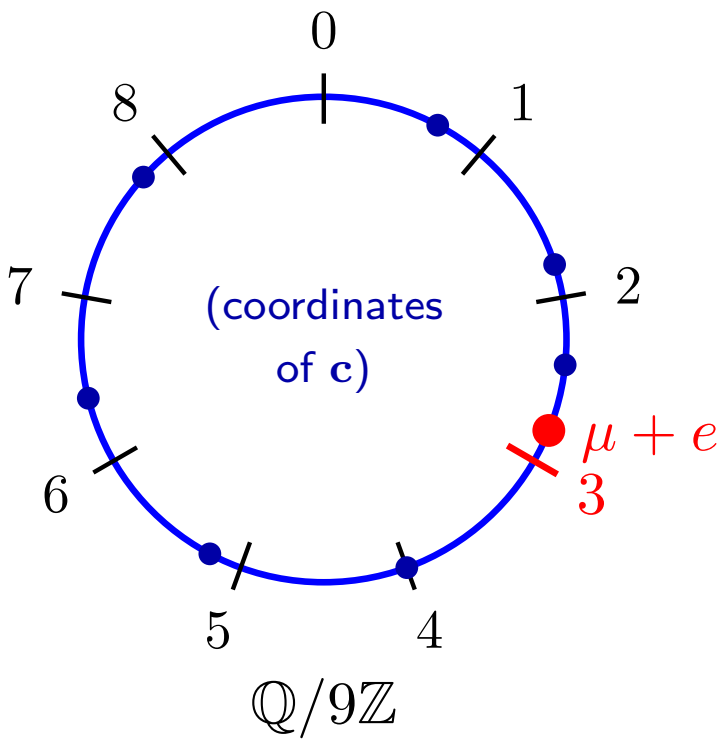
Warmup: Simplified BFV at the Right Scale



A ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a (typically affine-linear) polynomial

$$\mathbf{c}(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}]$$

Warmup: Simplified BFV at the Right Scale



A ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a (typically affine-linear) polynomial

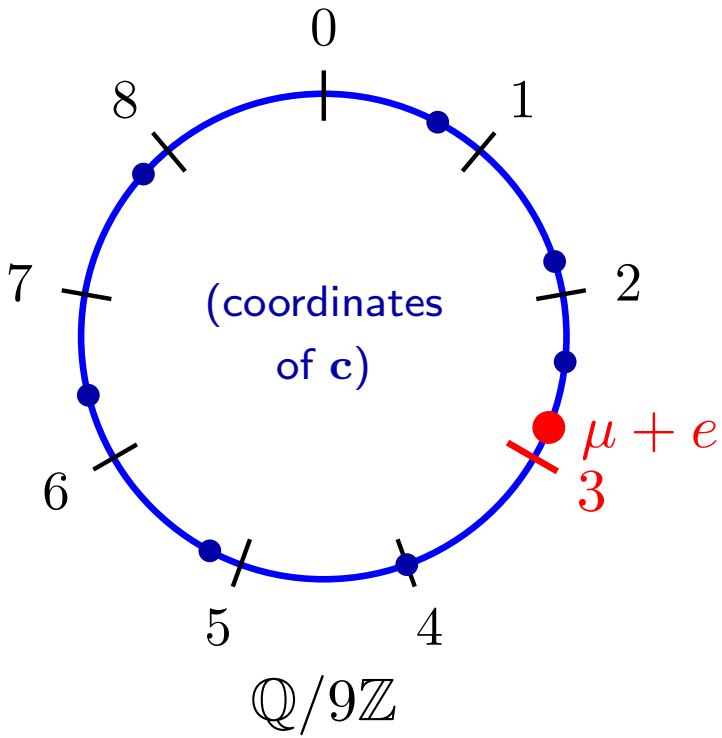
$$\mathbf{c}(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}]$$

such that

$$\mathbf{c}(\mathbf{s}) = \mu + e \approx \mu \pmod{p\mathbb{Z}}, \quad |e| \ll \tfrac{1}{2}.$$

Evaluate at the short secret $\mathbf{s} \in \mathbb{Z}^n$, then round to the nearest integer modulo p.

Warmup: Simplified BFV at the Right Scale



A ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a (typically affine-linear) polynomial

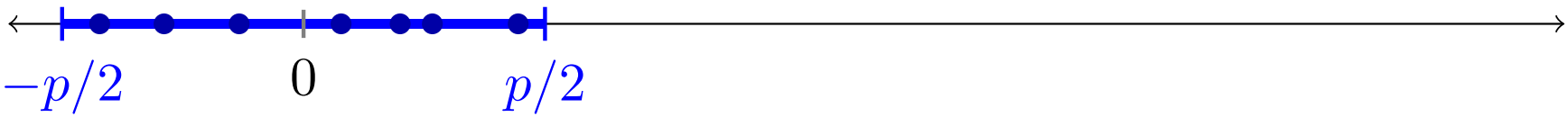
$$\mathbf{c}(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}]$$

such that

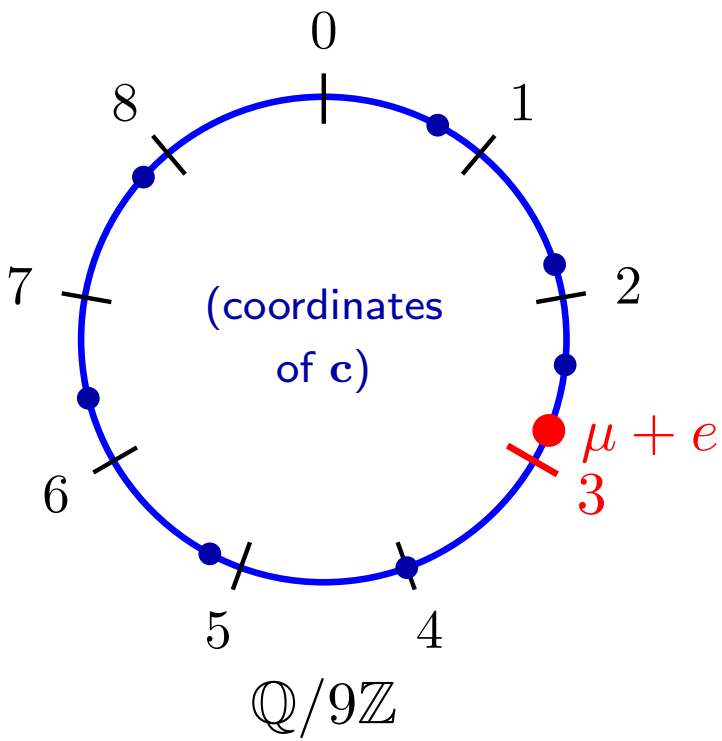
$$\mathbf{c}(\mathbf{s}) = \mu + e \approx \mu \pmod{p\mathbb{Z}}, \quad |e| \ll \tfrac{1}{2}.$$

Evaluate at the short secret $\mathbf{s} \in \mathbb{Z}^n$, then round to the nearest integer modulo p.

lifted ciphertext coordinates $\widehat{c_i} \in [-p/2, p/2)$:



Warmup: Simplified BFV at the Right Scale



A ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a (typically affine-linear) polynomial

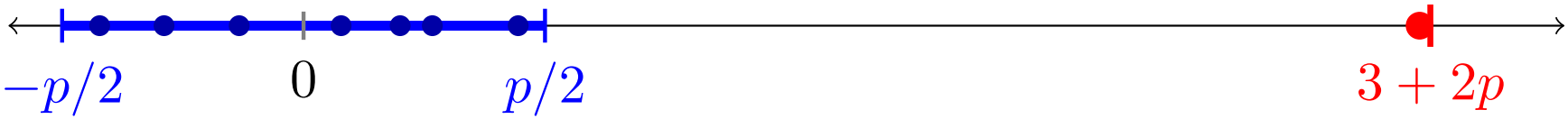
$$\mathbf{c}(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}]$$

such that

$$\mathbf{c}(\mathbf{s}) = \mu + e \approx \mu \pmod{p\mathbb{Z}}, \quad |e| \ll \tfrac{1}{2}.$$

Evaluate at the short secret $\mathbf{s} \in \mathbb{Z}^n$, then round to the nearest integer modulo p.

lifted ciphertext coordinates $\widehat{c}_i \in [-p/2, p/2)$:



$\widehat{\mathbf{c}}(\mathbf{s}) \approx \widehat{\mu}$

$3 + 2p$

$\widehat{\mu} = \mu \pmod{p}$

Warmup: BFV Homomorphic Operations

Addition: trivial

Just add the ciphertext polynomials:

$$\mathbf{c}_+(\vec{S}) := \mathbf{c}(\vec{S}) + \mathbf{c}'(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}], \quad \text{so} \quad \mathbf{c}_+(\mathbf{s}) \approx \mu + \mu' \pmod{p\mathbb{Z}}.$$

Warmup: BFV Homomorphic Operations

Addition: trivial

Just add the ciphertext polynomials:

$$\mathbf{c}_+(\vec{S}) := \mathbf{c}(\vec{S}) + \mathbf{c}'(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}], \quad \text{so} \quad \mathbf{c}_+(\mathbf{s}) \approx \mu + \mu' \pmod{p\mathbb{Z}}.$$

Multiplication: lift, then multiply

Lift to $\widehat{\mathbf{c}}(\vec{S}), \widehat{\mathbf{c}}'(\vec{S}) \in \mathbb{Q}[\vec{S}]$, where $\widehat{\mathbf{c}}(\mathbf{s}) = \widehat{\mu} + e$ and $\widehat{\mathbf{c}}'(\mathbf{s}) = \widehat{\mu}' + e'$, then set

$$\widehat{\mathbf{c}}_\times(\vec{S}) := \widehat{\mathbf{c}}(\vec{S}) \cdot \widehat{\mathbf{c}}'(\vec{S}).$$

Warmup: BFV Homomorphic Operations

Addition: trivial

Just add the ciphertext polynomials:

$$\mathbf{c}_+(\vec{S}) := \mathbf{c}(\vec{S}) + \mathbf{c}'(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}], \quad \text{so} \quad \mathbf{c}_+(\mathbf{s}) \approx \mu + \mu' \pmod{p\mathbb{Z}}.$$

Multiplication: lift, then multiply

Lift to $\widehat{\mathbf{c}}(\vec{S}), \widehat{\mathbf{c}}'(\vec{S}) \in \mathbb{Q}[\vec{S}]$, where $\widehat{\mathbf{c}}(\mathbf{s}) = \widehat{\mu} + e$ and $\widehat{\mathbf{c}}'(\mathbf{s}) = \widehat{\mu}' + e'$, then set

$$\widehat{\mathbf{c}}_\times(\vec{S}) := \widehat{\mathbf{c}}(\vec{S}) \cdot \widehat{\mathbf{c}}'(\vec{S}).$$

Correct:
$$\begin{aligned}\widehat{\mathbf{c}}_\times(\mathbf{s}) &= (\widehat{\mu} + e) \cdot \left(\widehat{\mu}' + e'\right) \\ &= \widehat{\mu} \cdot \widehat{\mu}' + (e \cdot \widehat{\mu}' + \widehat{\mu} \cdot e' + e \cdot e') \\ &\approx \widehat{\mu} \cdot \widehat{\mu}' = \mu \cdot \mu' \pmod{p\mathbb{Z}}.\end{aligned}$$

Warmup: BFV Homomorphic Operations

Addition: trivial

Just add the ciphertext polynomials:

$$\mathbf{c}_+(\vec{S}) := \mathbf{c}(\vec{S}) + \mathbf{c}'(\vec{S}) \in (\mathbb{Q}/p\mathbb{Z})[\vec{S}], \quad \text{so} \quad \mathbf{c}_+(\mathbf{s}) \approx \mu + \mu' \pmod{p\mathbb{Z}}.$$

Multiplication: lift, then multiply

Lift to $\widehat{\mathbf{c}}(\vec{S}), \widehat{\mathbf{c}}'(\vec{S}) \in \mathbb{Q}[\vec{S}]$, where $\widehat{\mathbf{c}}(\mathbf{s}) = \widehat{\mu} + e$ and $\widehat{\mathbf{c}}'(\mathbf{s}) = \widehat{\mu}' + e'$, then set

$$\widehat{\mathbf{c}}_\times(\vec{S}) := \widehat{\mathbf{c}}(\vec{S}) \cdot \widehat{\mathbf{c}}'(\vec{S}).$$

Correct:
$$\begin{aligned} \widehat{\mathbf{c}}_\times(\mathbf{s}) &= (\widehat{\mu} + e) \cdot (\widehat{\mu}' + e') \\ &= \widehat{\mu} \cdot \widehat{\mu}' + (e \cdot \widehat{\mu}' + \widehat{\mu} \cdot e' + e \cdot e') \\ &\approx \widehat{\mu} \cdot \widehat{\mu}' = \mu \cdot \mu' \pmod{p\mathbb{Z}}. \end{aligned}$$

Error blowup

Typically, $|\widehat{\mu}|, |\widehat{\mu}'| \approx p \cdot \|\mathbf{s}\|_2$, so multiplication amplifies noise e, e' by $\gg p$.

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$

✗ Addition needs nonlinear carry logic

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

- ✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$
- ✗ Addition needs nonlinear carry logic
- ✗ Multiplication needs $\Omega(d^2)$ $\mathbb{Z}_b$-products

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

- ✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$
- ✗ Addition needs nonlinear carry logic
- ✗ Multiplication needs $\Omega(d^2)$ $\mathbb{Z}_b$-products

'Relaxed' polynomial representations

- View the plaintext digits as coefficients of a polynomial:

$$\boldsymbol{\mu}(B) = \mu_0 + \mu_1 B + \cdots + \mu_{d-1} B^{d-1}, \qquad \mu = \boldsymbol{\mu}(b).$$

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

- ✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$
- ✗ Addition needs nonlinear carry logic
- ✗ Multiplication needs $\Omega(d^2)$ $\mathbb{Z}_b$-products

'Relaxed' polynomial representations

- View the plaintext digits as coefficients of a polynomial:

$$\boldsymbol{\mu}(B) = \mu_0 + \mu_1 B + \cdots + \mu_{d-1} B^{d-1}, \qquad \mu = \boldsymbol{\mu}(b).$$

- Add and multiply the plaintexts as polynomials

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

- ✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$
- ✗ Addition needs nonlinear carry logic
- ✗ Multiplication needs $\Omega(d^2)$ $\mathbb{Z}_b$-products

'Relaxed' polynomial representations

- View the plaintext digits as coefficients of a polynomial:

$$\boldsymbol{\mu}(B) = \mu_0 + \mu_1 B + \cdots + \mu_{d-1} B^{d-1}, \qquad \mu = \boldsymbol{\mu}(b).$$

- Add and multiply the plaintexts as polynomials
- Allow coefficients larger than b, but still $O(b)$; do not compute carries

Base-b Decomposition: Two Approaches

- **Natural idea:** express μ as d base-b digits, where $b \approx p^{1/d}$.

Arithmetic with Carries (e.g., TFHE)

- ✔ Encrypt each digit in plaintext ring $\mathbb{Z}_b$: error growth $\approx b$
- ✗ Addition needs nonlinear carry logic
- ✗ Multiplication needs $\Omega(d^2)$ $\mathbb{Z}_b$-products

'Relaxed' polynomial representations

- View the plaintext digits as coefficients of a polynomial:

$$\boldsymbol{\mu}(B) = \mu_0 + \mu_1 B + \cdots + \mu_{d-1} B^{d-1}, \qquad \mu = \boldsymbol{\mu}(b).$$

- Add and multiply the plaintexts as polynomials
- Allow coefficients larger than b, but still $O(b)$; do not compute carries
- Inspired by [CCKS'23] approx FHE, but technically very different.

Plaintext Decomposition as a Ring Isomorphism

- Fix a base b and define the $\mathbb{Z}[B]$-ideal

$$\mathcal{L} := \{\mathbf{z}(B) \in \mathbb{Z}[B] : \mathbf{z}(b) = 0 \pmod{p}\} \supseteq p\mathbb{Z}.$$

("Gadget lattice": kernel of powers-of-b vector $(1, b, b^2, \ldots)$ [MicPei12])

Plaintext Decomposition as a Ring Isomorphism

- Fix a base b and define the $\mathbb{Z}[B]$-ideal

$$\mathcal{L} := \{\mathbf{z}(B) \in \mathbb{Z}[B] : \mathbf{z}(b) = 0 \pmod{p}\} \supseteq p\mathbb{Z}.$$

("Gadget lattice": kernel of powers-of-b vector $(1, b, b^2, \ldots)$ [MicPei12])

- Evaluation at b gives a ring isomorphism:

$$\boxed{\mathbb{Z}[B]/\mathcal{L} \;\cong\; \mathbb{Z}_p}$$

$$\boldsymbol{\mu}(B) + \mathcal{L} \longmapsto \boldsymbol{\mu}(b) \bmod p,$$
$$\mu + \mathcal{L} \longmapsfrom \mu.$$

Plaintext Decomposition as a Ring Isomorphism

- Fix a base b and define the $\mathbb{Z}[B]$-ideal

$$\mathcal{L} := \{\mathbf{z}(B) \in \mathbb{Z}[B] : \mathbf{z}(b) = 0 \pmod{p}\} \supseteq p\mathbb{Z}.$$

("Gadget lattice": kernel of powers-of-b vector $(1, b, b^2, \ldots)$ [MicPei12])

- Evaluation at b gives a ring isomorphism:

$$\boxed{\mathbb{Z}[B]/\mathcal{L} \;\cong\; \mathbb{Z}_p}$$

$$\boldsymbol{\mu}(B) + \mathcal{L} \longmapsto \boldsymbol{\mu}(b) \bmod p,$$
$$\mu + \mathcal{L} \longmapsfrom \mu.$$

- A decomposition of μ is any representative $\boldsymbol{\mu}(B) \in \mu + \mathcal{L}$.

Plaintext Decomposition as a Ring Isomorphism

- Fix a base b and define the $\mathbb{Z}[B]$-ideal

$$\mathcal{L} := \{\mathbf{z}(B) \in \mathbb{Z}[B] : \mathbf{z}(b) = 0 \pmod{p}\} \supseteq p\mathbb{Z}.$$

("Gadget lattice": kernel of powers-of-b vector $(1, b, b^2, \ldots)$ [MicPei12])

- Evaluation at b gives a ring isomorphism:

$$\boxed{\mathbb{Z}[B]/\mathcal{L} \;\cong\; \mathbb{Z}_p}$$

$$\boldsymbol{\mu}(B) + \mathcal{L} \longmapsto \boldsymbol{\mu}(b) \bmod p,$$
$$\mu + \mathcal{L} \longmapsfrom \mu.$$

- A decomposition of μ is any representative $\boldsymbol{\mu}(B) \in \mu + \mathcal{L}$.

The Payoff

For $d = \lceil \log_b p \rceil$, every plaintext has many decompositions

$$\boldsymbol{\mu}(B) = \mu_0 + \mu_1 B + \cdots + \mu_{d-1} B^{d-1}, \qquad |\mu_i| \approx b \approx p^{1/d}.$$

Short Representatives Modulo $\mathcal{L}$

Toy example: let $b = 3$ and $p = b^2 = 9$.

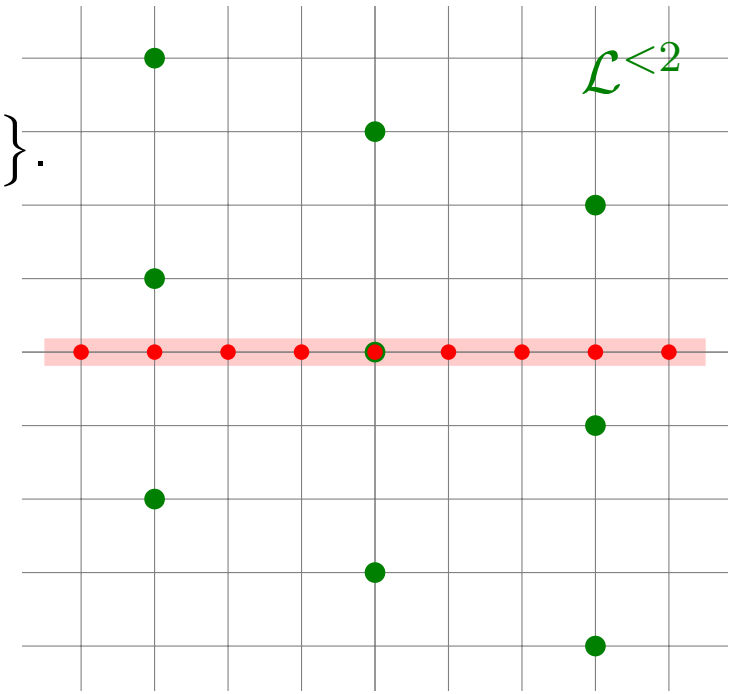


$\mathbb{Q}, \mathbb{Z}$ representatives mod $p\mathbb{Z}$

Short Representatives Modulo $\mathcal{L}$

Toy example: let $b = 3$ and $p = b^2 = 9$.
Define

$$\mathcal{L}^{<2} := \left\{ z(B) \in \mathbb{Z}^{<2}[B] : z(3) = 0 \pmod 9 \right\}.$$



$\mathbb{Q}, \mathbb{Z}$ representatives mod $p\mathbb{Z}$

Short Representatives Modulo $\mathcal{L}$

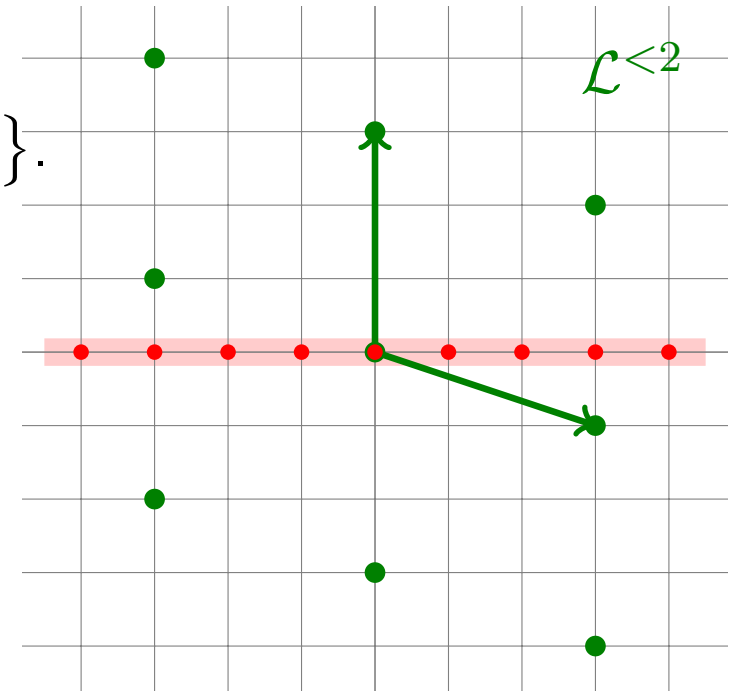
Toy example: let $b = 3$ and $p = b^2 = 9$.
Define

$$\mathcal{L}^{<2} := \left\{ z(B) \in \mathbb{Z}^{<2}[B] : z(3) = 0 \pmod 9 \right\}.$$

In coefficient form, a short basis is

$$\mathbf{b}_1 = (3, -1), \qquad \mathbf{b}_2 = (0, 3).$$

$\mathbb{Q}, \mathbb{Z}$ representatives mod $p\mathbb{Z}$



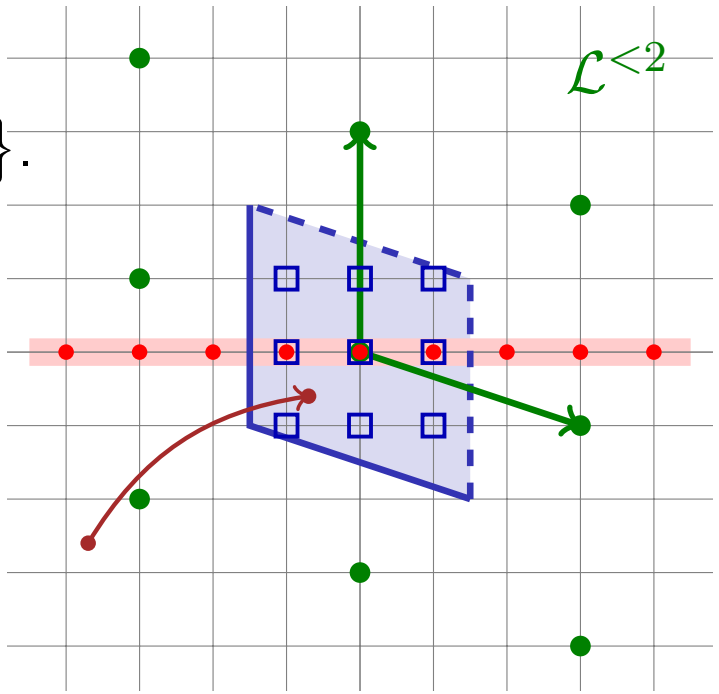
Short Representatives Modulo $\mathcal{L}$

Toy example: let $b = 3$ and $p = b^2 = 9$.
Define

$$\mathcal{L}^{<2} := \left\{ z(B) \in \mathbb{Z}^{<2}[B] : z(3) = 0 \pmod 9 \right\}.$$

In coefficient form, a short basis is

$$\mathbf{b}_1 = (3, -1), \qquad \mathbf{b}_2 = (0, 3).$$



$\mathbb{Q}, \mathbb{Z}$ representatives mod $p\mathbb{Z}$

$\mathbb{Q}^{<2}[B], \mathbb{Z}^{<2}[B]$ representatives mod $\mathcal{L}^{<2}$

The Isomorphism Extends to *Noisy Short* Decompositions

- Let $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu} + \mathbf{e}$ and $\widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu}' + \mathbf{e}'$ be noisy 'short' decompositions (suppressing their argument B):

The Isomorphism Extends to *Noisy Short* Decompositions

- Let $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu} + \mathbf{e}$ and $\widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu}' + \mathbf{e}'$ be noisy 'short' decompositions (suppressing their argument B):

$$\widetilde{\boldsymbol{\mu}} + \widetilde{\boldsymbol{\mu}}' = (\boldsymbol{\mu} + \boldsymbol{\mu}') + (\mathbf{e} + \mathbf{e}'),$$

$$\widetilde{\boldsymbol{\mu}} \cdot \widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu} \cdot \boldsymbol{\mu}' + (\mathbf{e}\boldsymbol{\mu}' + \boldsymbol{\mu}\mathbf{e}' + \mathbf{e}\mathbf{e}').$$

The Isomorphism Extends to *Noisy Short* Decompositions

- Let $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu} + \mathbf{e}$ and $\widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu}' + \mathbf{e}'$ be noisy 'short' decompositions (suppressing their argument B):

$$\widetilde{\boldsymbol{\mu}} + \widetilde{\boldsymbol{\mu}}' = (\boldsymbol{\mu} + \boldsymbol{\mu}') + (\mathbf{e} + \mathbf{e}'),$$

$$\widetilde{\boldsymbol{\mu}} \cdot \widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu} \cdot \boldsymbol{\mu}' + (\mathbf{e}\boldsymbol{\mu}' + \boldsymbol{\mu}\mathbf{e}' + \mathbf{e}\mathbf{e}').$$

The Win

- Short decompositions have size about $d\,b \approx d\,p^{1/d}$, not p.
  So multiplication grows noise by about this factor.

The Isomorphism Extends to *Noisy Short* Decompositions

- Let $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu} + \mathbf{e}$ and $\widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu}' + \mathbf{e}'$ be noisy 'short' decompositions (suppressing their argument B):

$$\widetilde{\boldsymbol{\mu}} + \widetilde{\boldsymbol{\mu}}' = (\boldsymbol{\mu} + \boldsymbol{\mu}') + (\mathbf{e} + \mathbf{e}'),$$

$$\widetilde{\boldsymbol{\mu}} \cdot \widetilde{\boldsymbol{\mu}}' = \boldsymbol{\mu} \cdot \boldsymbol{\mu}' + (\mathbf{e}\boldsymbol{\mu}' + \boldsymbol{\mu}\mathbf{e}' + \mathbf{e}\mathbf{e}').$$

The Win

- Short decompositions have size about $d\,b \approx d\,p^{1/d}$, not p.
  So multiplication grows noise by about this factor.
- The decompositions can be noncanonical; only shortness is needed.

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$,

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$, satisfying

$$\boxed{\mathbf{c}(B, \mathbf{s}) = \boldsymbol{\mu}(B) + \mathbf{e}(B) \approx \mu \pmod{\mathcal{L}}.}$$

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$, satisfying

$$\boxed{\mathbf{c}(B, \mathbf{s}) = \boldsymbol{\mu}(B) + \mathbf{e}(B) \approx \mu \pmod{\mathcal{L}}.}$$

1. Evaluate at the secret key $\mathbf{s}$ to get $\approx \boldsymbol{\mu}(B) \bmod \mathcal{L}$

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$, satisfying

$$\boxed{\mathbf{c}(B, \mathbf{s}) = \boldsymbol{\mu}(B) + \mathbf{e}(B) \approx \mu \pmod{\mathcal{L}}.}$$

1. Evaluate at the secret key $\mathbf{s}$ to get $\approx \boldsymbol{\mu}(B) \bmod \mathcal{L}$
2. Round the coefficients to recover $\boldsymbol{\mu}(B) \bmod \mathcal{L}$

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$, satisfying

$$\boxed{\mathbf{c}(B, \mathbf{s}) = \boldsymbol{\mu}(B) + \mathbf{e}(B) \approx \mu \pmod{\mathcal{L}}.}$$

1. Evaluate at the secret key $\mathbf{s}$ to get $\approx \boldsymbol{\mu}(B) \bmod \mathcal{L}$
2. Round the coefficients to recover $\boldsymbol{\mu}(B) \bmod \mathcal{L}$
3. Evaluate at $B = b$ to recover $\mu \in \mathbb{Z}_p$

Decomposed BFV (dBFV)

A dBFV ciphertext encrypting $\mu \in \mathbb{Z}_p$ is a bivariate polynomial

$$\mathbf{c}(B, \vec{S}) \in (\mathbb{Q}[B]/\mathcal{L})[\vec{S}],$$

typically affine linear in the secret-key variable $\vec{S}$, satisfying

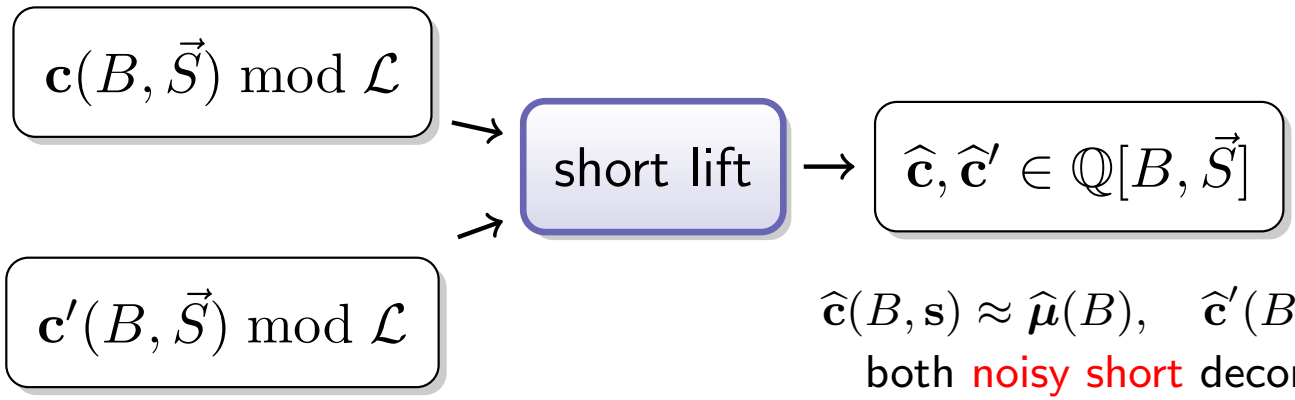
$$\boxed{\mathbf{c}(B, \mathbf{s}) = \boldsymbol{\mu}(B) + \mathbf{e}(B) \approx \mu \pmod{\mathcal{L}}.}$$

1. Evaluate at the secret key $\mathbf{s}$ to get $\approx \boldsymbol{\mu}(B) \bmod \mathcal{L}$
2. Round the coefficients to recover $\boldsymbol{\mu}(B) \bmod \mathcal{L}$
3. Evaluate at $B = b$ to recover $\mu \in \mathbb{Z}_p$

Fresh ciphertexts and security

- Fresh ciphertexts are mod-$\mathcal{L}$-LWE samples with error rate $\propto 1/p^{1/d}$.
- Hardness follows by a nearly lossless reduction from the corresponding standard plain/Module/Ring-LWE problem.

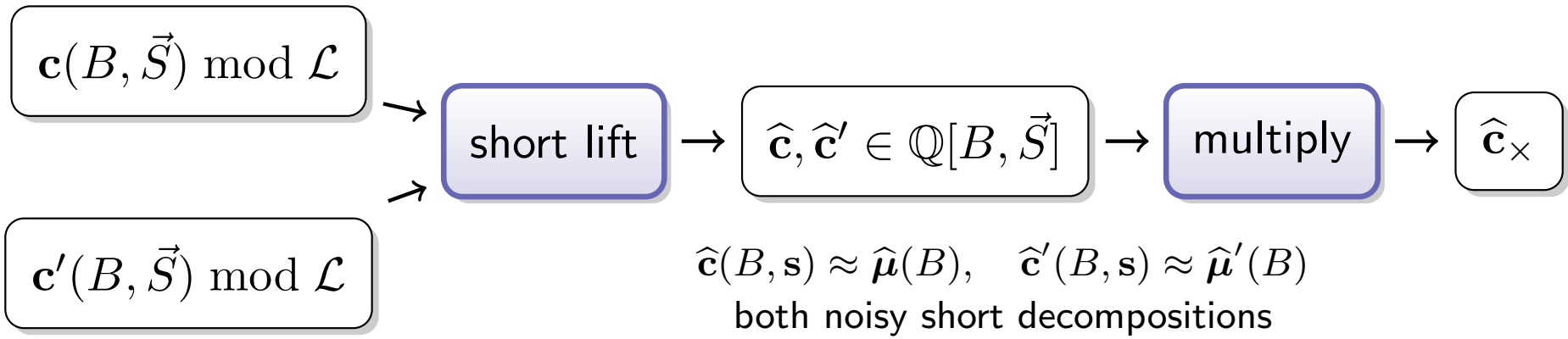
Homomorphic Multiplication in dBFV



$\widehat{\mathbf{c}}(B, \mathbf{s}) \approx \widehat{\boldsymbol{\mu}}(B), \quad \widehat{\mathbf{c}}'(B, \mathbf{s}) \approx \widehat{\boldsymbol{\mu}}'(B)$

both noisy short decompositions

Homomorphic Multiplication in dBFV

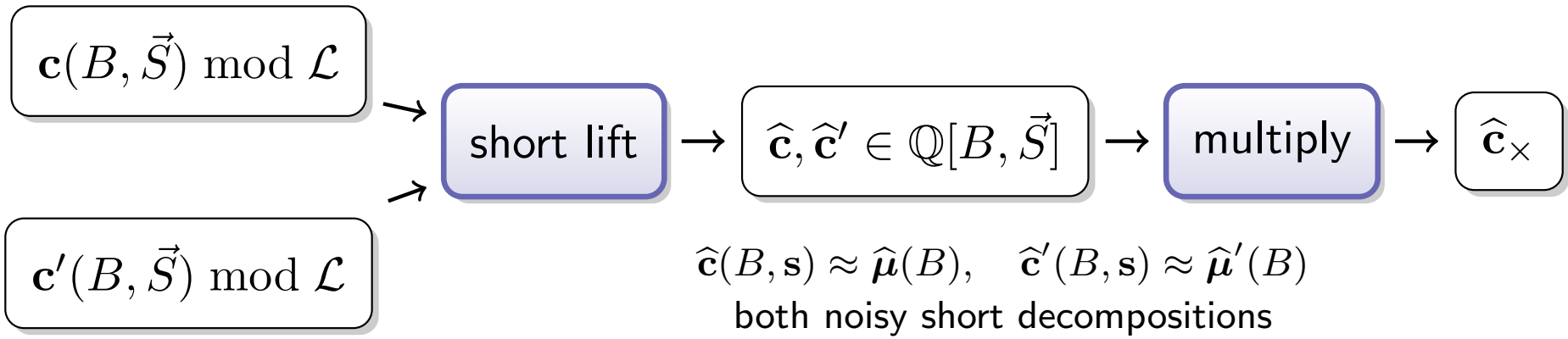


$$\widehat{\mathbf{c}}(B, \mathbf{s}) \approx \widehat{\boldsymbol{\mu}}(B), \quad \widehat{\mathbf{c}}'(B, \mathbf{s}) \approx \widehat{\boldsymbol{\mu}}'(B)$$

both noisy short decompositions

$$\widehat{\mathbf{c}}_\times(B, \mathbf{s}) \approx \widehat{\boldsymbol{\mu}}(B) \cdot \widehat{\boldsymbol{\mu}}'(B) = \mu \cdot \mu' \pmod{\mathcal{L}}$$

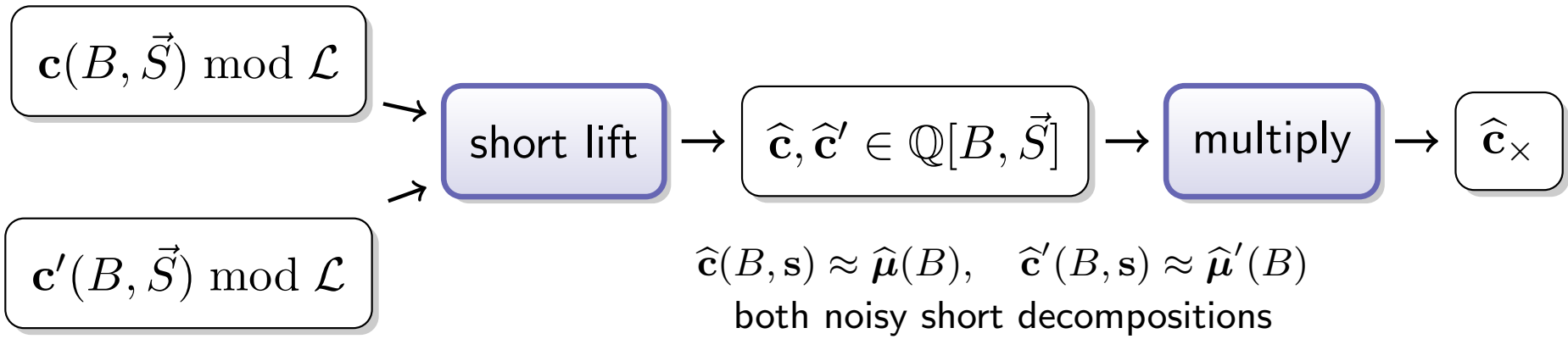
Homomorphic Multiplication in dBFV



Then restore to fresh-ciphertext form:

1. reduce modulo $\mathcal{L}$,

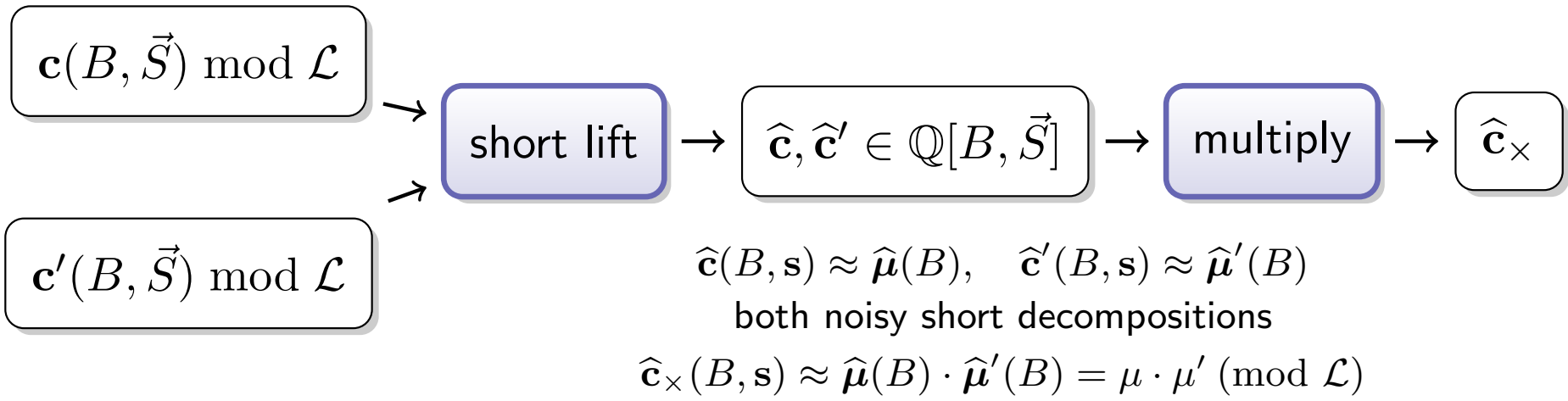
Homomorphic Multiplication in dBFV



Then restore to fresh-ciphertext form:

1. reduce modulo $\mathcal{L}$,
2. reduce degree in B,

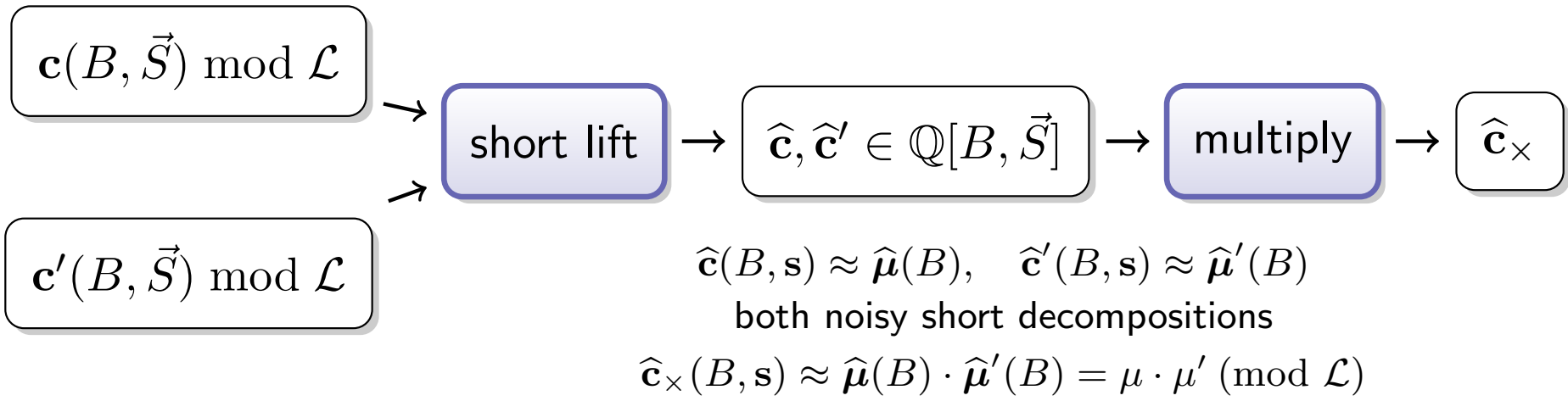
Homomorphic Multiplication in dBFV



Then restore to fresh-ciphertext form:

1. reduce modulo $\mathcal{L}$,
2. reduce degree in B,
3. relinearize in $\vec{S}$ by standard key switching.

Homomorphic Multiplication in dBFV



Then restore to fresh-ciphertext form:

1. reduce modulo $\mathcal{L}$,
2. reduce degree in B,
3. relinearize in $\vec{S}$ by standard key switching.

Degree reduction in B

- For $p = b^d$, simply use $B^d = 0 \pmod{\mathcal{L}}$.
- In general, high powers of B reduce to short combos of low powers.

Putting It All Together

Using d-digit ciphertexts with base $b \approx p^{1/d}$:

	ordinary BFV	dBFV
Fresh LWE error rate	$\approx 1/p$	$\approx 1/p^{1/d}$
Multiplicative growth	$\approx p$	$\approx d\,p^{1/d}$

Putting It All Together

Using d-digit ciphertexts with base $b \approx p^{1/d}$:

	ordinary BFV	dBFV
Fresh LWE error rate	$\approx 1/p$	$\approx 1/p^{1/d}$
Multiplicative growth	$\approx p$	$\approx d\,p^{1/d}$

Conclusions

- Taking $d \approx \log p$, dBFV's error profile matches ordinary BFV for an exponentially smaller modulus $\approx \log p$.

Putting It All Together

Using d-digit ciphertexts with base $b \approx p^{1/d}$:

	ordinary BFV	dBFV
Fresh LWE error rate	$\approx 1/p$	$\approx 1/p^{1/d}$
Multiplicative growth	$\approx p$	$\approx d\,p^{1/d}$

Conclusions

- Taking $d \approx \log p$, dBFV's error profile matches ordinary BFV for an exponentially smaller modulus $\approx \log p$.
- End-to-end sizes and times can be quasi-linear in $\log p$.

Putting It All Together

Using d-digit ciphertexts with base $b \approx p^{1/d}$:

	ordinary BFV	dBFV
Fresh LWE error rate	$\approx 1/p$	$\approx 1/p^{1/d}$
Multiplicative growth	$\approx p$	$\approx d\,p^{1/d}$

Conclusions

- Taking $d \approx \log p$, dBFV's error profile matches ordinary BFV for an exponentially smaller modulus $\approx \log p$.
- End-to-end sizes and times can be quasi-linear in $\log p$.
- Compatibility with arbitrary moduli and rings, SIMD packing, and bootstrapping.

More in the Paper

- **dBGV and dGSW:** the same decomposed viewpoint beyond BFV

More in the Paper

- **dBGV and dGSW:** the same decomposed viewpoint beyond BFV
- **Tight hardness for mod-$\mathcal{L}$-LWE:** nearly lossless reductions from ordinary plain/Module/Ring-LWE

More in the Paper

- **dBGV and dGSW:** the same decomposed viewpoint beyond BFV
- **Tight hardness for mod-$\mathcal{L}$-LWE:** nearly lossless reductions from ordinary plain/Module/Ring-LWE
- **(Self-)bootstrapping options:** generic coefficient-wise bootstrapping, plus low-error-growth dBFV/dGSW hosts

More in the Paper

- **dBGV and dGSW:** the same decomposed viewpoint beyond BFV
- **Tight hardness for mod-$\mathcal{L}$-LWE:** nearly lossless reductions from ordinary plain/Module/Ring-LWE
- **(Self-)bootstrapping options:** generic coefficient-wise bootstrapping, plus low-error-growth dBFV/dGSW hosts
- **Custom comparison circuit:** comparison and sign on relaxed digits, using fewer, parallel programmable bootstraps

More in the Paper

- **dBGV and dGSW:** the same decomposed viewpoint beyond BFV
- **Tight hardness for mod-$\mathcal{L}$-LWE:** nearly lossless reductions from ordinary plain/Module/Ring-LWE
- **(Self-)bootstrapping options:** generic coefficient-wise bootstrapping, plus low-error-growth dBFV/dGSW hosts
- **Custom comparison circuit:** comparison and sign on relaxed digits, using fewer, parallel programmable bootstraps

IACR ePrint 2025/2321

Thanks! Questions?