Fast Blind Rotation for Bootstrapping FHEs

Binwu Xiang; Jiang Zhang; Yi Deng; Yiran Dai; Dengguo Feng

State Key Laboratory of Cryptology

CRYPTO 2023

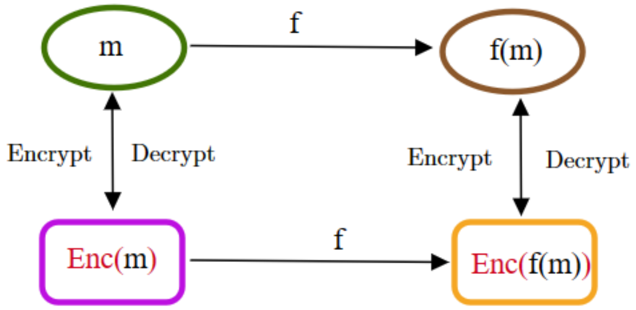
State Key Laboratory of Cryptology

Outline

- Preliminaries
- Motivation and Technical Contribution
- New NTRU-based GSW-like Scheme and blind rotation
- Experimental Results and Conclusion

Homomorphic Encryption

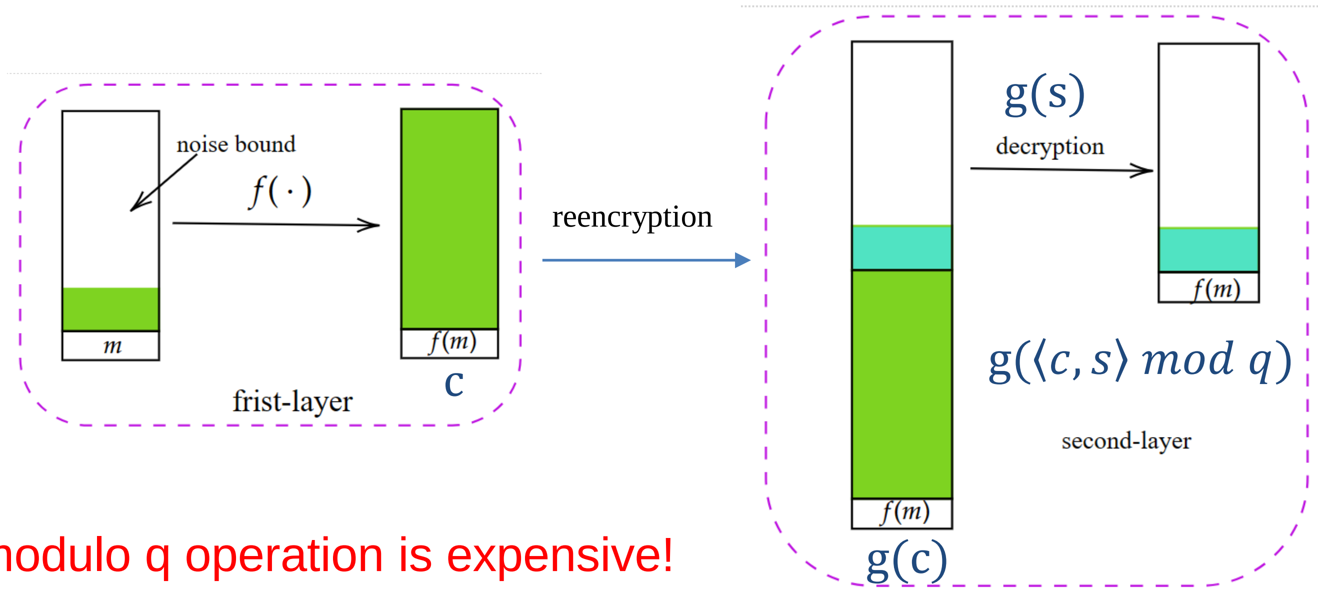
- ➢ Addition and multiplication over ciphertext space
  - $Enc(a+b)=Enc(a)+Enc(b)$ $Enc(a\cdot b)=Enc(a)\cdot Enc(b)$
- ➢ Classification by evaluation function
  - Partially Homomorphic Encryption
  - Somewhat Homomorphic Encryption
  - Fully Homomorphic Encryption



Fully Homomorphic Encryption

➢Two-layer framework:

- First layer: Noise-based somewhat HE
- Second-layer: Bootstrapping (Homomorphically computes the decryption of SHE)



The modulo q operation is expensive!

Current State of the Art

Three approaches to solve the modulo q operation.

- Digit extraction ([BGV, BFV])
  - Convert $f(\mathbf{c},\mathbf{s}) = \sum_i c_i s_i \mod q$ to $f'(\mathbf{c},\mathbf{s}) = \sum_i c_i s_i \mod p^r$
  - Extract the bit decompositions of $\sum_i c_i s_i$ in base p
- Function approximation ([CKKS])
  - Use math function to approximate the modular function $f(\mathbf{c},\mathbf{s}) = \sum_i c_i s_i \mod q$
  - e.g. $\sum_i c_i s_i \mod q \approx \frac{q}{2\pi} \sin\left(\frac{2\pi}{q}\langle \mathbf{c},\mathbf{s}\rangle\right)$
- Limitations
  - Slow bootstrap (20 secs)
  - Large FHE parameters => huge BK (10 GB)

Blind Rotation

- Homomorphically decrypt an LWE ciphertext on the exponent.
- **Definition(q=2N)**

  Input : LWE ciphertext $(\mathbf{a}, b = \sum_{i=0}^{n-1} a_i s_i - \mathsf{noised}(m)) \in \mathbb{Z}_q^{n+1}$

  Rotation polynomial $r(X) \in \mathbb{Z}_Q[X]/X^N + 1$

  Evaluation key $\mathbf{EVK}$

  Output: $\boxed{g(r(X) \cdot X^{\mathsf{noised}(m)})}$

- The modulo q operation can be done for free in the exponent.

$$\boxed{X^{\mathsf{noised}(m)} = X^{\sum_{i=0}^{n-1} a_i s_i - b \mod q} = X^{-b} X^{\sum_{i=0}^{n-1} a_i s_i}}$$

- The constant term of $\boxed{r(X) \cdot X^{\mathsf{noised}(m)}}$ is exactly m.
- Fast bootstrap (ms) and small parameter size (MB)

State of the art works

- RLWE-based blind rotation:
  - ✓ AP/FHEW [EUROCRYPT 2015]
    - all secret keys distribution,
    - large boot key
  - ✓ GINX/TFHE [EUROCRYPT 2016]
    - limited secret key distribution
    - small boot key
  - ✓ Lee et al. [EUROCRYPT 2023]
    - all secret key distribution,
    - small boot key

- NTRU-based blind rotation:
  - ✓ FINAL [ASIACRYPT 2022]
  - ✓ NTRU-vum [CCS 2022]
    - TFHE-like;
    - limited secret key distribution;
    - small boot key

Outline

- Preliminaries
- Motivation and Technical Contribution
- New NTRU-based GSW-like Scheme and blind rotation
- Experimental Results and Conclusion

Motivation

- Recommended key distributions in [ACC21]: Uniform, Gaussian, and Ternary
- Final[BIP22], NTRU-vum[Klu22] use binary or ternary secrets for performance consideration.
- Potential Problem: small secrets are subject to special attacks

  [Alb17, AGVW17, SC19, EJK20]
- Design bootstrapping for large keys may be of independent interest.

Contributions

✓ We design a new NTRU-based GSW-like encryption

- Faster external product
- Faster key-switching and ring automorphism

✓ We propose a new blind rotation using NTRU and ring automorphism

- Performance asymptotically independent from the key distributions
- Comparison (evaluation key size , computational efficiency)

Key distribution	AP, GINX, LEE's	FINAL NTRU-vum
Ternery	better	equal
Uniform and Gussian	better	better

✓ We use our new blind rotation to bootstrap an LWE-based scheme

- 53% faster than TFHE and Smaller evaluation key (18MB)

Outline

- Preliminaries
- Motivation and Technical Contribution
- New NTRU-based GSW-like Scheme and blind rotation
- Experimental Results and Conclusion

NTRU-based GSW-like Scheme

- Scalar and Vector ciphertext
  - Scalar: $c = \mathrm{NTRU}_{Q,f}(u) := g/f + \Delta u/f \in R_Q$
  - Vector $\mathbf{c}' = \mathrm{NTRU}_{Q,f'}(v) := \mathbf{g}/f' + \left(B^0, B^1, \ldots, B^{d-1}\right) \cdot v \in R_Q^d$
- Homomorphic Multiplication

$$c \odot \mathbf{c}' = \langle \mathrm{BitDecom}_B(c), \mathbf{c}' \rangle = \sum_{i=0}^{d-1} c_i c_i' = \sum_{i=0}^{d-1} c_i g_i / f' + cv \in R_Q$$

- External product
  - Let $f = f'$, we have $c \odot \mathbf{c}' = \left(\sum_{i=0}^{d-1} g_i c_i + gv\right) / f + \Delta \cdot uv/f$
- Key Switching
  - Let $v = f/f'$, we have $c \odot \mathbf{c}' = \left(\sum_{i=0}^{d-1} g_i c_i + g\right) / f' + \Delta \cdot u/f'$

Both External product and Key Switching only need d multiplications on R_Q

New blind rotation

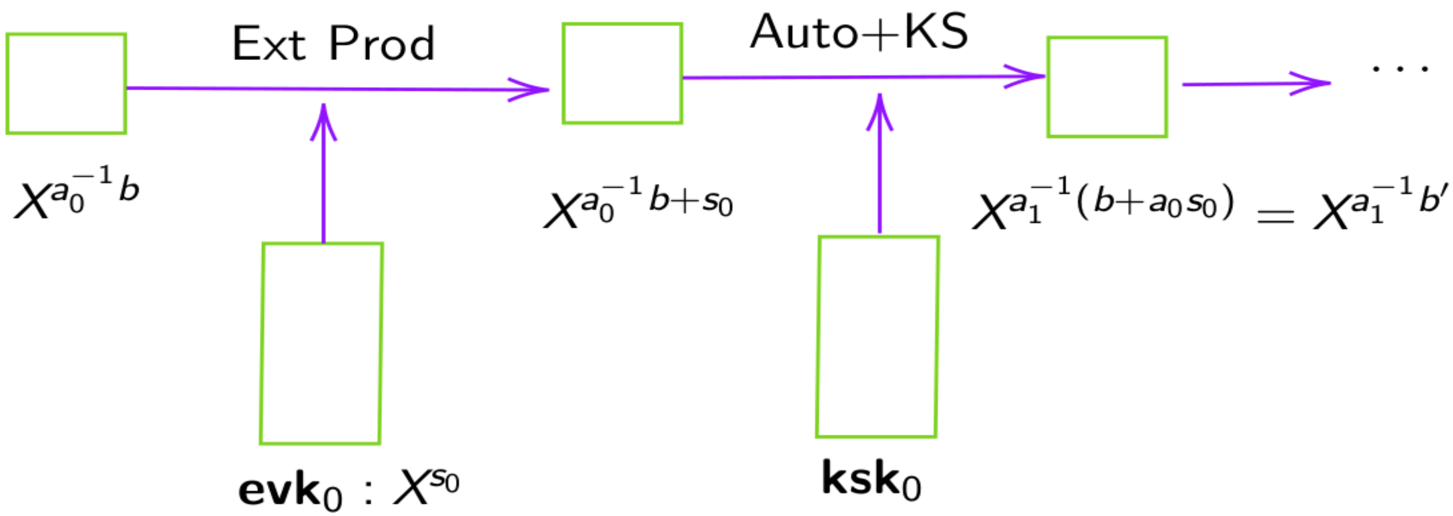
- Recall blind rotation: homomorphically decrypt the LWE ciphertext on the exponent

$$r(X)X^{\sum_{i=0}^{n-1} a_i s_i - b \mod q} = r(X)X^{-b}X^{\sum_{i=0}^{n-1} a_i s_i}$$

- Basic idea
  - ✓ Given a ciphertext $\mathrm{NTRU}_{f(x)}(X^{s_i})$, applying $X \to X^{a_i}$.
  - ✓ Perform once key-switching: convert the secret $f(X^{a_i})$ to $f(X)$.

$$\boxed{\mathrm{NTRU}_{f(X)}(X^{s_i})} \xrightarrow{\text{EvalAuto}} \boxed{\mathrm{NTRU}_{f(X^{a_i})}(X^{a_i s_i})} \xrightarrow{\text{KS}} \boxed{\mathrm{NTRU}_{f(X)}(X^{a_i s_i})}$$

New blind rotation



- The first scalar NTRU ciphertext requires a specific form.
  - Counteracting the impact of automorphism on b
- Proper automorphism ensures the same ciphertext structure .
  - Eg. $X \to X^{a_0 a_1^{-1}}$, in the next iteration, the initial message remains $X^{a_1^{-1}b'}$
- Cost: n external products and n key-switchings

Construction

➢ Problem I: automorphism exists only for odd numbers in $\mathbb{Z}_{2N}$

Transformation:

- ✓ Set $q = N$, X^2 has order q in $R = \mathbb{Z}[X]/X^N + 1$
- ✓ Define $S = \{2i + 1 : 0 \leq i \leq q - 1\} \subset \mathbb{Z}_{2N}$ is a a multiplicative subgroup of $\mathbb{Z}_{2N}$
- ✓ Easily check: $\forall w, \hat{w} \in S, w^{-1}, \hat{w}^{-1}, w\hat{w} \in S$

Set $w_i = 2a_i + 1 < 2N$, we have

$$\boxed{X^{2\sum_{i=0}^{n-1} a_i s_i} = X^{\sum_{i=0}^{n-1} w_i s_i} \cdot X^{-\sum_{i=0}^{n-1} s_i}}$$

- ✓ Only need extra one external product and one evaluation key.

Construction

➢Problem II: Accumulator initialization

- Previous Scheme (a polynomial can be viewed as a noiseless ciphertext)

$$\left(0, r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right) = \mathrm{RLWE}\left(r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right)$$

- This feature is not satisfied for our KDM-form encryption:

💡Design the evaluation key carefully.

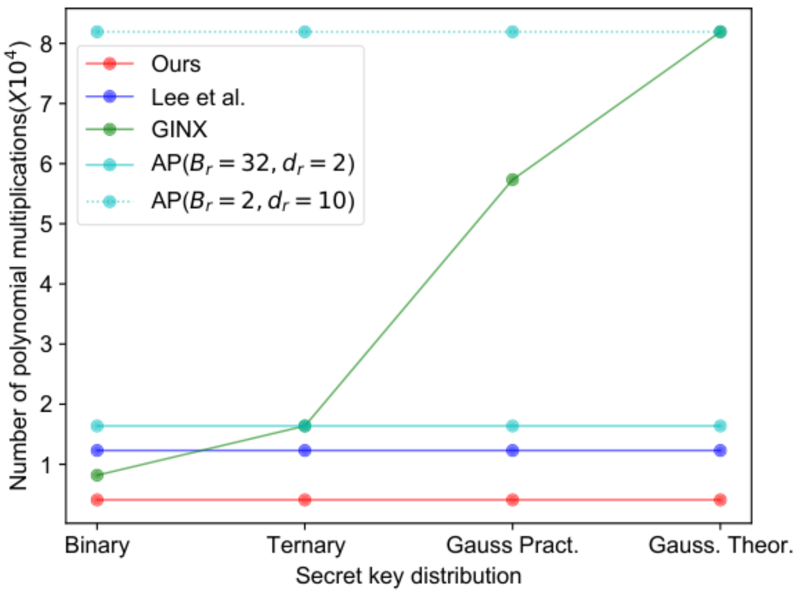
$$\mathbf{evk}_0 = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_0}/f), \qquad \mathbf{evk}_i = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_i}) \text{ for } 1 \le i < n,$$
$$\mathbf{evk}_n = \mathrm{NTRU}'_{Q,f,\tau}(X^{-\sum_{i=0}^{n-1} s_i}).$$

✓ External product still satisfied.

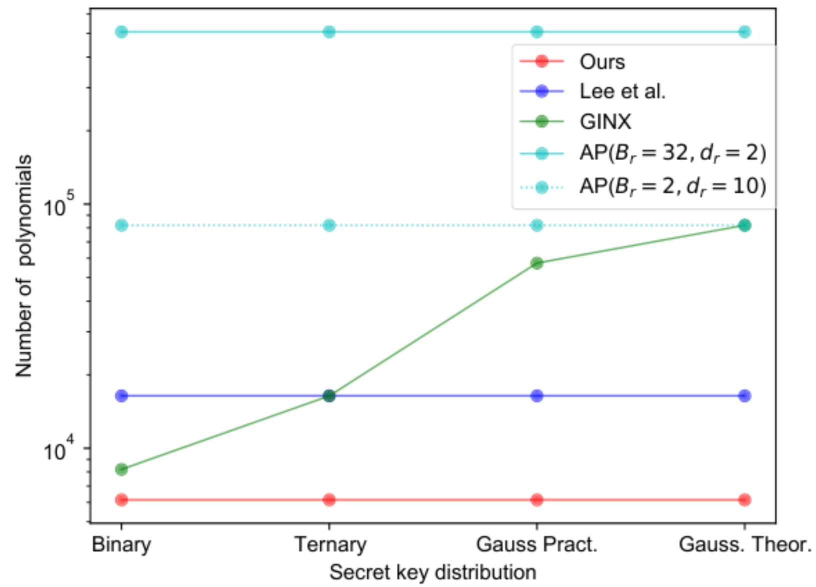
$$\boxed{u \in R_Q \odot \ \mathrm{NTRU}'(v/f) \to \mathrm{NTRU}(uv)}$$

Comparison

➤Comparison of blind rotations using different first-layer key distributions



(a) The number of ring multiplications for blind rotation.



(b) The number of ring elements in the evaluation key for blind rotation.

Bootstrapping for LWE

- Overview of bootstrapping in FHEW-scheme.

$$\mathsf{LWE}_{q,\mathbf{s}}(m) \xrightarrow{\mathsf{BREval}} \boxed{\mathsf{ACC}} \xrightarrow{\mathsf{Ext}} \mathsf{LWE}_{Q,\mathbf{s}'}(m) \xrightarrow{\mathsf{ModSwitch}} \mathsf{LWE}_{q,\mathbf{s}}(m)$$

$$\mathsf{RLWE}_{Q,s'}(r(X)X^{\mathsf{noised}(m)})$$

- Our bootstrapping

$$\mathsf{LWE}_{q,\mathbf{s}}(m) \xrightarrow{\mathsf{BREval}} \boxed{\mathsf{ACC}} \xrightarrow{\mathsf{Ext}} \mathsf{LWE}_{Q,\mathbf{s}'}(m) \xrightarrow{\mathsf{ModSwitch}} \mathsf{LWE}_{q,\mathbf{s}}(m)$$

$$\mathsf{NTRU}_{Q,s'}(r(X)X^{\mathsf{noised}(m)})$$

How to transform NTRU ciphertext into LWE ciphertext?

Bootstrapping for LWE

- Ext: Extract LWE sample form NTRU ciphertext
  - For our NTRU ciphertext $c = (g+m)/f$
  - coefficient vectors: $\mathbf{c} = (c_0, \ldots, c_{N-1})$, $\mathbf{f} = (f_0, \ldots, f_{N-1})$
  - Treat it as an RLWE ciphertext $(a, b = as + e)$ with

$$\boxed{a = c, b = 0, s = f}$$

  - We can extract $\mathrm{LWE}_{Q,\mathbf{f}}(m) = (\hat{\mathbf{c}} = (c_0, -c_{N-1}, \ldots, -c_1), 0)$

Outline

- Preliminaries
- Motivation and Technical Contribution
- New NTRU-based GSW-like Scheme and blind rotation
- Experimental Results and Conclusion

Recommended Parameters

Parameters	Key distrib.	n	q	N	Q	B	Q_{ks}	B_{ks}
STD128 [39]	Ternary	512	1024	1024	2^{27}	2^7	2^{14}	2^7
P128T	Ternary	512	1024	1024	$995329 \approx 2^{19.9}$	2^4	2^{14}	2^7
P128G	Gaussian	465	1024	1024	$995329 \approx 2^{19.9}$	2^4	2^{14}	2^7
STD192 [39]	Ternary	1024	1024	2048	2^{37}	2^{13}	2^{19}	28
P192T	Ternary	1024	1024	2048	$44421121 \approx 2^{25.4}$	2^9	2^{19}	28
P192G	Gaussian	870	1024	2048	$44421121 \approx 2^{25.4}$	2^9	2^{17}	28

Experimental Results

Algorithms	Parameters	Key distrib.	Timings (ms)	EVK (MB)	KSK (MB)	Boots. key (MB)
FHEW/AP [22,6]	STD128 [39]	Ternary	359	1674	224	1898
TFHE/GINX [19,6]	STD128 [39]	Ternary	234	54	224	278
Ours	P128T	Ternary	112	18.65	224	242.65
	P128G	Gaussian	100	17.90	203.44	221.34
FHEW/AP [22,6]	STD192 [39]	Ternary	1200	6682	532	7214
TFHE/GINX [19,6]	STD192 [39]	Ternary	859	222	532	754
Ours	P192T	Ternary	320	38.10	532	570.10
	P192G	Gaussian	273	34.30	404.41	438.71

- Extra 10% improvement over P128T by using Gaussian key
- Extra 17% improvement over P192T by using Gaussian key

Thanks!

GitHub home page:

https://github.com/SKLC-FHE/CHIFHE

State Key Laboratory of Cryptology