Fast Blind Rotation for Bootstrapping FHEs

Binwu Xiang[1,2], Jiang Zhang[2], Yi Deng[1], Yiran Dai[1,2], Dengguo Feng[2]

[1]Chinese Academy of Sciences, [2]State Key Laboratory of Cryptology

Part I

Preliminaries

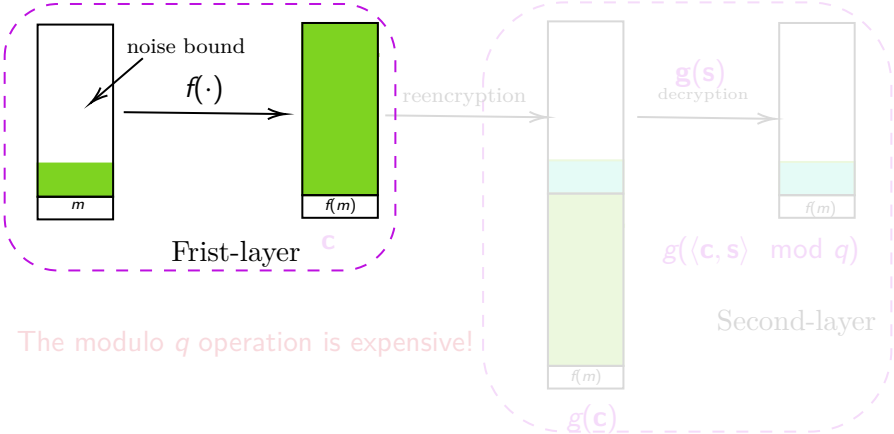
Fully Homomorphic Encryption

- Two-layer framework:
  1. Noise-based somewhat HE
  2. Homomorphically computes the decryption of SHE (Bootstrapping)

The modulo q operation is expensive!

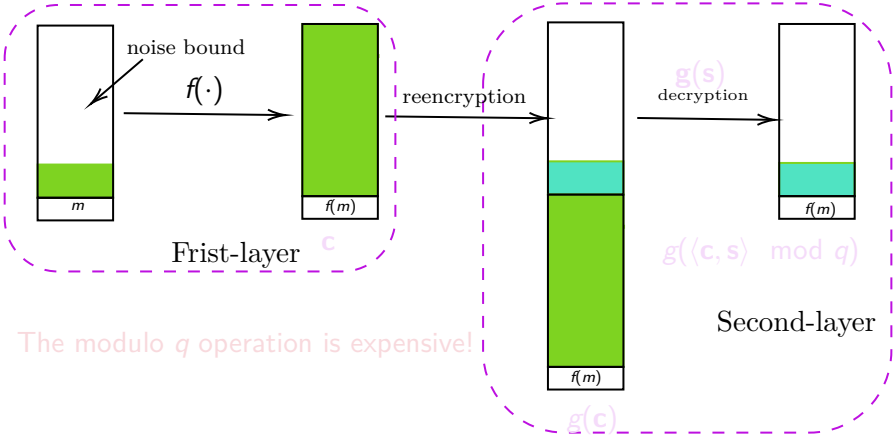
Fully Homomorphic Encryption

- Two-layer framework:
  1. Noise-based somewhat HE
  2. Homomorphically computes the decryption of SHE (Bootstrapping)



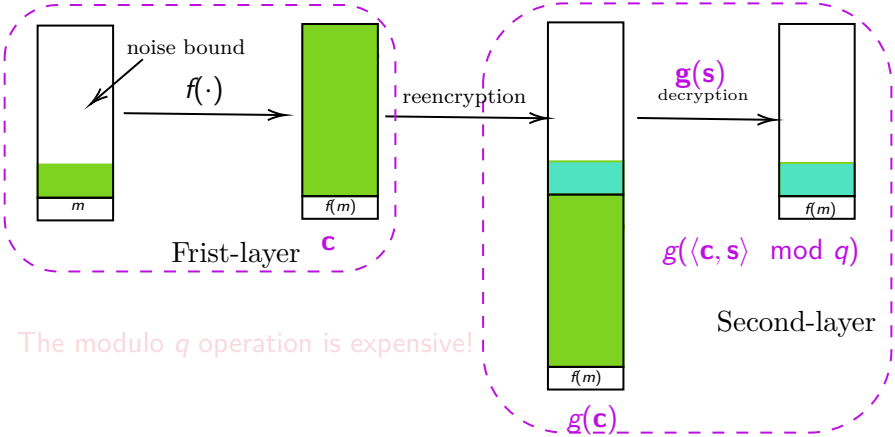
Fully Homomorphic Encryption

- Two-layer framework:
  1. Noise-based somewhat HE
  2. Homomorphically computes the decryption of SHE (Bootstrapping)



Fully Homomorphic Encryption

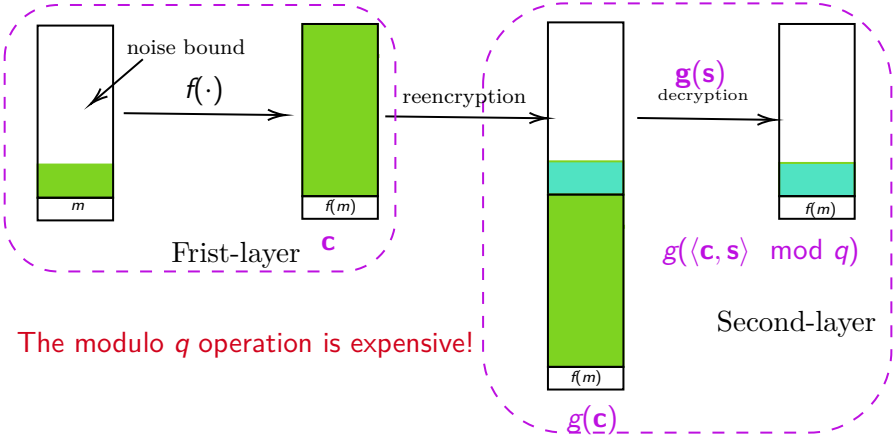
- Two-layer framework:
  1. Noise-based somewhat HE
  2. Homomorphically computes the decryption of SHE (Bootstrapping)



The modulo q operation is expensive!

Fully Homomorphic Encryption

- Two-layer framework:
  1. Noise-based somewhat HE
  2. Homomorphically computes the decryption of SHE (Bootstrapping)



The modulo q operation is expensive!

Blind Rotation

- Homomorphically decrypt an LWE ciphertext on the exponent.

Definition (Blind Rotation(case $q = 2N$))

- Input:
  - LWE ciphertext $(\mathbf{a}, b = \sum_{i=0}^{n-1} a_i s_i - \text{noised}(m)) \in \mathbb{Z}_q^{n+1}$;
  - $r(X) \in R_Q = \mathbb{Z}_Q[X]/(X^N + 1)$; evaluation key **EVK**;
- Output: $g(r(X) \cdot X^{\text{noised}(m)})$

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

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

- The constant term of $r(X) \cdot X^{\text{noised}(m)}$ is exactly m.

Blind Rotation

- Homomorphically decrypt an LWE ciphertext on the exponent.

Definition (Blind Rotation(case $q = 2N$))

- Input:
  - LWE ciphertext $(\mathbf{a}, b = \sum_{i=0}^{n-1} a_i s_i - \text{noised}(m)) \in \mathbb{Z}_q^{n+1}$;
  - $r(X) \in R_Q = \mathbb{Z}_Q[X]/(X^N + 1)$; evaluation key **EVK**;
- Output: $g(r(X) \cdot X^{\text{noised}(m)})$

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

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

- The constant term of $r(X) \cdot X^{\text{noised}(m)}$ is exactly m.

Blind Rotation

- Homomorphically decrypt an LWE ciphertext on the exponent.

Definition (Blind Rotation(case $q = 2N$))

- Input:
  - LWE ciphertext $(\mathbf{a}, b = \sum_{i=0}^{n-1} a_i s_i - \text{noised}(m)) \in \mathbb{Z}_q^{n+1}$;
  - $r(X) \in R_Q = \mathbb{Z}_Q[X]/(X^N + 1)$; evaluation key **EVK**;
- Output: $g(r(X) \cdot X^{\text{noised}(m)})$

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

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

- The constant term of $r(X) \cdot X^{\text{noised}(m)}$ is exactly m.

Blind Rotation

- Homomorphically decrypt an LWE ciphertext on the exponent.

Definition (Blind Rotation(case $q = 2N$))

- Input:
  - LWE ciphertext $(\mathbf{a}, b = \sum_{i=0}^{n-1} a_i s_i - \text{noised}(m)) \in \mathbb{Z}_q^{n+1}$;
  - $r(X) \in R_Q = \mathbb{Z}_Q[X]/(X^N + 1)$; evaluation key **EVK**;
- Output: $g(r(X) \cdot X^{\text{noised}(m)})$

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

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

- The constant term of $r(X) \cdot X^{\text{noised}(m)}$ is exactly m.

State of the art works

- RLWE-based blind rotation:
  - AP/FHEW [ASP14, DM15]
    - decompose $a_i = \sum_j a_{i,j} B^j$, all secret key distribution, large evaluation key
  - GINX/TFHE [GINX16, CGGI20]
    - decompose $s_i = \sum_{u \in U} s_{i,u} u$ for some public set U, limited secret key distribution, small evaluation key
  - Lee et al. [LMK+23]
    - ring automorphism, all secret key distribution, small evaluation key
- NTRU-based blind rotation:
  - Final[BIP+22], NTRU-vum[Klu22]
    - TFHE-like, limited secret key distribution, faster external product

State of the art works

- RLWE-based blind rotation:
  - AP/FHEW [ASP14, DM15]
    - decompose $a_i = \sum_j a_{i,j} B^j$, all secret key distribution, large evaluation key
  - GINX/TFHE [GINX16, CGGI20]
    - decompose $s_i = \sum_{u \in U} \bar{s}_{i,u} u$ for some public set U, limited secret key distribution, small evaluation key
  - Lee et al. [LMK+23]
    - ring automorphism, all secret key distribution, small evaluation key
- NTRU-based blind rotation:
  - Final[BIP+22], NTRU-vum[Klu22]
    - TFHE-like, limited secret key distribution, faster external product

State of the art works

- RLWE-based blind rotation:
  - AP/FHEW [ASP14, DM15]
    - decompose $a_i = \sum_j a_{i,j} B^j$, all secret key distribution, large evaluation key
  - GINX/TFHE [GINX16, CGGI20]
    - decompose $s_i = \sum_{u \in U} s_{i,u} u$ for some public set U, limited secret key distribution, small evaluation key
  - Lee et al. [LMK+23]
    - ring automorphism, all secret key distribution, small evaluation key
- NTRU-based blind rotation:
  - Final[BIP+22], NTRU-vum[Klu22]
    - TFHE-like, limited secret key distribution, faster external product

State of the art works

- RLWE-based blind rotation:
  - AP/FHEW [ASP14, DM15]
    - decompose $a_i = \sum_j a_{i,j} B^j$, all secret key distribution, large evaluation key
  - GINX/TFHE [GINX16, CGGI20]
    - decompose $s_i = \sum_{u \in U} s_{i,u} u$ for some public set U, limited secret key distribution, small evaluation key
  - Lee et al. [LMK+23]
    - ring automorphism, all secret key distribution, small evaluation key
- NTRU-based blind rotation:
  - Final[BIP+22], NTRU-vum[Klu22]
    - TFHE-like, limited secret key distribution, faster external product

State of the art works

- RLWE-based blind rotation:
  - AP/FHEW [ASP14, DM15]
    - decompose $a_i = \sum_j a_{i,j} B^j$, all secret key distribution, large evaluation key
  - GINX/TFHE [GINX16, CGGI20]
    - decompose $s_i = \sum_{u \in U} s_{i,u} u$ for some public set U, limited secret key distribution, small evaluation key
  - Lee et al. [LMK+23]
    - ring automorphism, all secret key distribution, small evaluation key
- NTRU-based blind rotation:
  - Final[BIP+22], NTRU-vum[Klu22]
    - TFHE-like, limited secret key distribution, faster external product

Motivation

- Recommended key distributions in [ACC+21]: Uniform, Gaussian, and Ternary distribution.
- Final[BIP+22], 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.

Motivation

- Recommended key distributions in [ACC+21]: Uniform, Gaussian, and Ternary distribution.
- Final[BIP+22], 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.

Motivation

- Recommended key distributions in [ACC+21]: Uniform, Gaussian, and Ternary distribution.
- Final[BIP+22], 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.

Motivation

- Recommended key distributions in [ACC+21]: Uniform, Gaussian, and Ternary distribution.
- Final[BIP+22], 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.

Part II

Technical Contributions

Technical Contribution

- 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
  - all secret key distribution, small evaluation key
- We use our new blind rotation to bootstrap an LWE-based scheme
  - Faster bootstrapping time(53% faster than TFHE)
  - Smaller evaluation key (18MB)
- Extra improvement by using large key distributions (17%)

Technical Contribution

- 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
  - all secret key distribution, small evaluation key
- We use our new blind rotation to bootstrap an LWE-based scheme
  - Faster bootstrapping time(53% faster than TFHE)
  - Smaller evaluation key (18MB)
- Extra improvement by using large key distributions (17%)

Technical Contribution

- 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
  - all secret key distribution, small evaluation key
- We use our new blind rotation to bootstrap an LWE-based scheme
  - Faster bootstrapping time(53% faster than TFHE)
  - Smaller evaluation key (18MB)
- Extra improvement by using large key distributions (17%)

Technical Contribution

- 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
  - all secret key distribution, small evaluation key
- We use our new blind rotation to bootstrap an LWE-based scheme
  - Faster bootstrapping time(53% faster than TFHE)
  - Smaller evaluation key (18MB)
- Extra improvement by using large key distributions (17%)

Part III

NTRU-based GSW-like Scheme

NTRU-based GSW-like Scheme

- Parameters (τ, Δ) depends on the encoding

$$(\tau, \Delta) = \begin{cases} \left(1, \left\lfloor \frac{Q}{t} \right\rceil\right), & \text{if noised}(m) = e + \left\lfloor \frac{q}{t} \right\rceil \cdot m; \\ (t, 1), & \text{if noised}(m) = t \cdot e + m; \\ (1, 1), & \text{if noised}(m) = e + m. \end{cases}$$

- scalar ciphertext: $\mathrm{NTRU}_{Q,f,\tau,\Delta}(u) := \tau \cdot g/f + \Delta \cdot u/f \in R_Q$
- vector ciphertext: $\mathrm{NTRU}'_{Q,f,\tau}(v) := (\tau \cdot g_0/f + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f + B^{d-1} \cdot v) \in R_Q^d$

NTRU-based GSW-like Scheme

- Parameters (τ, Δ) depends on the encoding

$$(\tau, \Delta) = \begin{cases} \left(1, \left\lfloor \frac{Q}{t} \right\rceil\right), & \text{if noised}(m) = e + \left\lfloor \frac{q}{t} \right\rceil \cdot m; \\ (t, 1), & \text{if noised}(m) = t \cdot e + m; \\ (1, 1), & \text{if noised}(m) = e + m. \end{cases}$$

- scalar ciphertext: $\mathrm{NTRU}_{Q,f,\tau,\Delta}(u) := \tau \cdot g/f + \Delta \cdot u/f \in R_Q$
- vector ciphertext: $\mathrm{NTRU}'_{Q,f,\tau}(v) := (\tau \cdot g_0/f + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f + B^{d-1} \cdot v) \in R_Q^d$

NTRU-based GSW-like Scheme

- Parameters (τ, Δ) depends on the encoding

$$(\tau, \Delta) = \begin{cases} \left(1, \left\lfloor \frac{Q}{t} \right\rceil\right), & \text{if noised}(m) = e + \left\lfloor \frac{q}{t} \right\rceil \cdot m; \\ (t, 1), & \text{if noised}(m) = t \cdot e + m; \\ (1, 1), & \text{if noised}(m) = e + m. \end{cases}$$

- scalar ciphertext: $\mathrm{NTRU}_{Q,f,\tau,\Delta}(u) := \tau \cdot g/f + \Delta \cdot u/f \in R_Q$
- vector ciphertext: $\mathrm{NTRU}'_{Q,f,\tau}(v) := (\tau \cdot g_0/f + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f + B^{d-1} \cdot v) \in R_Q^d$

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

NTRU-based GSW-like Scheme

- Let $c = \tau \cdot g/f + \Delta \cdot u/f$, $\mathbf{c}' = (\tau \cdot g_0/f' + B^0 \cdot v, \cdots, \tau \cdot g_{d-1}/f' + B^{d-1} \cdot v)$
- Basic operation
  - $\mathrm{BitDecom}_B(c) = (c_0, c_1, \ldots, c_{d-1}) \in R_B^d$ such that $c = \sum_{i=0}^{d-1} c_i \cdot B^i$
  - $$c \odot \mathbf{c}' = \sum_{i=0}^{d-1} c_i c_i' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i \right) / f' + \tau \cdot gv/f + \Delta \cdot uv/f$$
- External product
  - Let $f = f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + gv \right) / f + \Delta \cdot uv/f$
  - d multiplications on R_q
- Key Switching
  - Let $v = f/f'$, $c \odot \mathbf{c}' = \tau \cdot \left(\sum_{i=0}^{d-1} g_i c_i + g \right) / f' + \Delta \cdot u/f'$
  - d multiplications on R_q

Part IV

New blind rotation

Basic idea

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

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

Basic idea

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

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

Basic idea

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

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

Basic idea

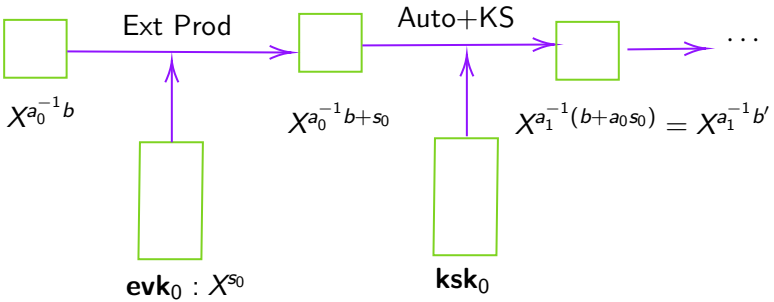
- Recall blind rotation: homomorphically decrypt the LWE ciphertext on the exponent

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

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

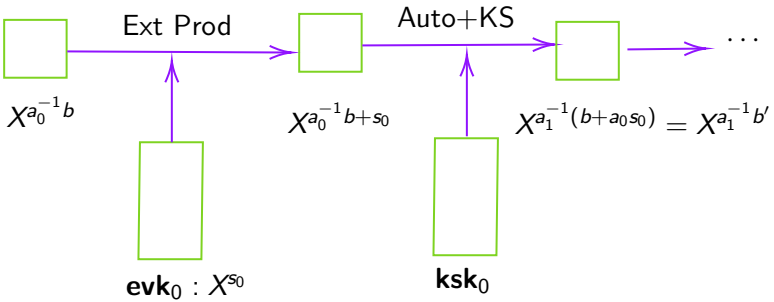
$$\mathrm{NTRU}_{f(X)}(X^{s_i}) \xrightarrow{\text{EvalAuto}} \mathrm{NTRU}_{f(X^{a_i})}(X^{a_i s_i}) \xrightarrow{\text{KS}} \mathrm{NTRU}_{f(X)}(X^{a_i s_i})$$

Basic idea



- The first scalar NTRU ciphertext requires a specific form.
- Proper automorphism ensures uniform ciphertext structure (eg. $X \to X^{a_0 a_1^{-1}}$).
- n external products and n key-switchings

Basic idea



- The first scalar NTRU ciphertext requires a specific form.
- Proper automorphism ensures uniform ciphertext structure (eg. $X \to X^{a_0 a_1^{-1}}$).
- n external products and n key-switchings

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i+1 : 1 \leq i \leq q-1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i+1 : 1 \leq i \leq q-1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i+1 : 1 \leq i \leq q-1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i + 1 : 1 \leq i \leq q - 1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i + 1 : 1 \leq i \leq q - 1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i+1 : 1 \leq i \leq q-1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i + 1 : 1 \leq i \leq q - 1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i+1 : 1 \leq i \leq q-1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem I: Each $a_i \in \mathbb{Z}_q$ can be not coprime to 2N (probability 1/2)
- Transformation:
  - Set $q = N$,X^2 has order q
  - Define $S = \{2i + 1 : 1 \leq i \leq q - 1\} \subset \mathbb{Z}_{2N}$, $S \cup \{1\}$ is a multiplicative subgroup of $\mathbb{Z}_{2N}$
  - Easily check:$\forall w, \hat{w} \in S \cup \{1\}$ $w^{-1}, \hat{w}^{-1}, w\hat{w} \in S \cup \{1\}$
- $X^{2a_i s_i} = X^{(2a_i+1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$, where $w_i = 2a_i + 1 < 2N$
- $r(X^2) \cdot X^{-2b} X^{2(\sum_{i=0}^{n-1} a_i s_i)} = r(X^2) \cdot X^{-2b} X^{\sum_{i=0}^{n-1} w_i s_i} X^{-\sum_{i=0}^{n-1} s_i}$
- Extra once external product and one evaluation key.
- General case $(q|N)$:$X^{\frac{2N}{q} a_i s_i} = X^{(\frac{2N}{q} a_i + 1)s_i - s_i} = X^{w_i s_i} X^{-s_i}$

Construction

- Problem II: Accumulator initialization
  - RLWE $\left(r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right) = \left(0, r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right)$ is a noiseless ciphertext.
  - $r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}$ cannot be publicly created in our case.
- Design the evaluation key carefully
  $\mathbf{evk}_0 = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_0}/f)$, $\mathbf{evk}_i = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_i})$ for $1 \leq i < n$,
  $\mathbf{evk}_n = \mathrm{NTRU}'_{Q,f,\tau}(X^{-\sum_{i=0}^{n-1} s_i})$.
- External product still satisfied.

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

Construction

- Problem II: Accumulator initialization
  - RLWE $\left(r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right) = \left(0, r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right)$ is a noiseless ciphertext.
  - $r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}$ cannot be publicly created in our case.
- Design the evaluation key carefully
  $\mathbf{evk}_0 = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_0}/f)$, $\mathbf{evk}_i = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_i})$ for $1 \leq i < n$,
  $\mathbf{evk}_n = \mathrm{NTRU}'_{Q,f,\tau}(X^{-\sum_{i=0}^{n-1} s_i})$.
- External product still satisfied.

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

Construction

- Problem II: Accumulator initialization
  - RLWE $\left(r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right) = \left(0, r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right)$ is a noiseless ciphertext.
  - $r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}$ cannot be publicly created in our case.
- Design the evaluation key carefully

  $\mathbf{evk}_0 = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_0}/f)$, $\quad \mathbf{evk}_i = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_i})$ for $1 \leq i < n$,

  $\mathbf{evk}_n = \mathrm{NTRU}'_{Q,f,\tau}(X^{-\sum_{i=0}^{n-1} s_i})$.
- External product still satisfied.

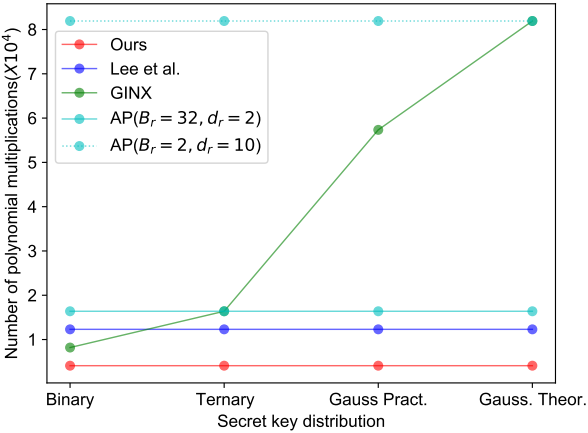
$$u \in R_Q \odot \mathrm{NTRU}'(v/f) \to \mathrm{NTRU}(uv)$$

Construction

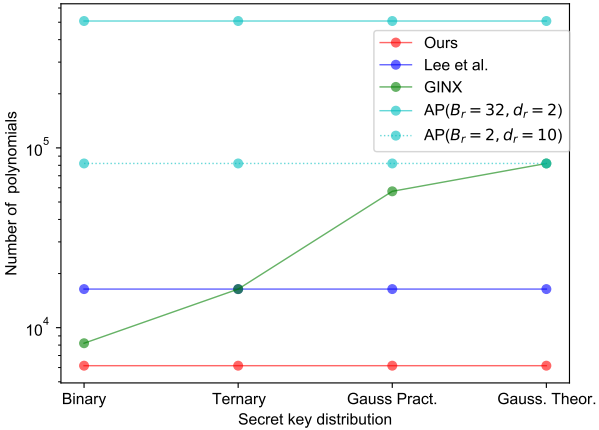
- Problem II: Accumulator initialization
  - RLWE $\left(r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right) = \left(0, r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}\right)$ is a noiseless ciphertext.
  - $r(X^{\frac{2N}{q}}) \cdot X^{-\frac{2N}{q}b}$ cannot be publicly created in our case.
- Design the evaluation key carefully
  $\mathbf{evk}_0 = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_0}/f)$, $\mathbf{evk}_i = \mathrm{NTRU}'_{Q,f,\tau}(X^{s_i})$ for $1 \leq i < n$,
  $\mathbf{evk}_n = \mathrm{NTRU}'_{Q,f,\tau}(X^{-\sum_{i=0}^{n-1} s_i})$.
- External product still satisfied.

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

Comparison


Ours
Lee et al.
GINX
AP($B_r = 32, d_r = 2$)
AP($B_r = 2, d_r = 10$)
Number of polynomial multiplications($X10^4$)
1
2
3
4
5
6
7
8
Binary
Ternary
Gauss Pract.
Gauss. Theor.
Secret key distribution

Comparison


Ours
Lee et al.
GINX
AP($B_r = 32, d_r = 2$)
AP($B_r = 2, d_r = 10$)
Number of polynomials
10^5
10^4
Binary
Ternary
Gauss Pract.
Gauss. Theor.
Secret key distribution

Part V

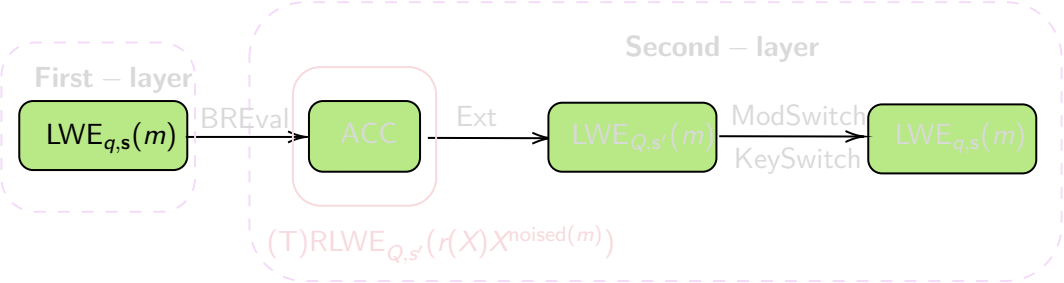
Bootstrapping

Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting

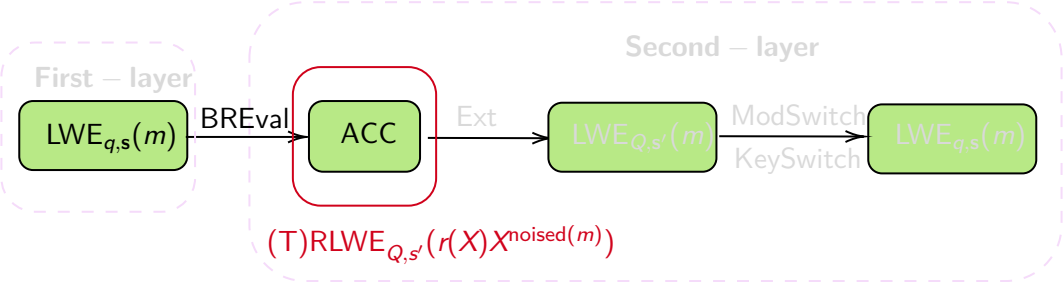
Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting



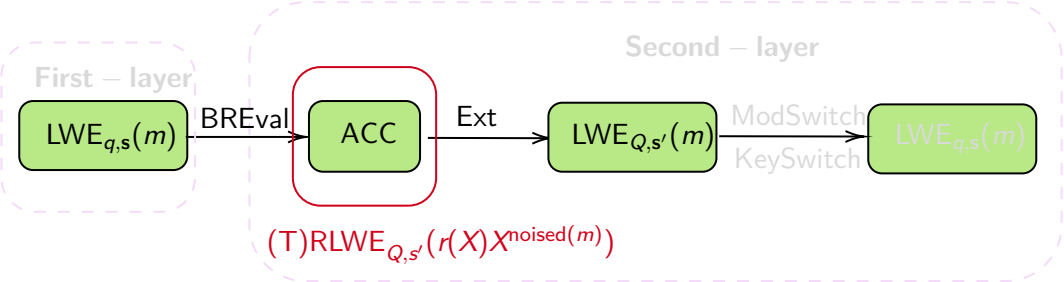
Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting



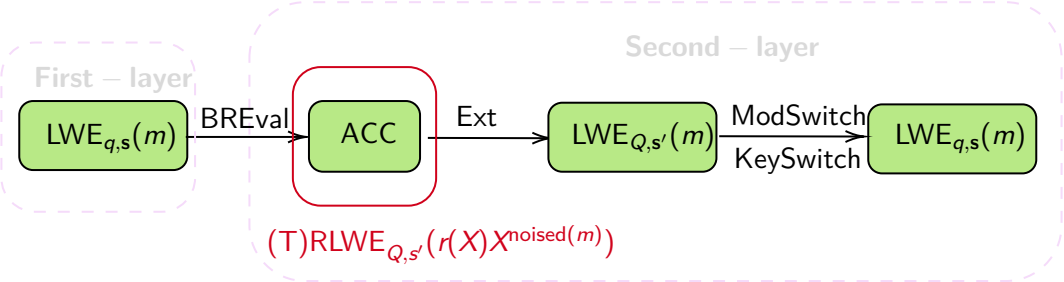
Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting



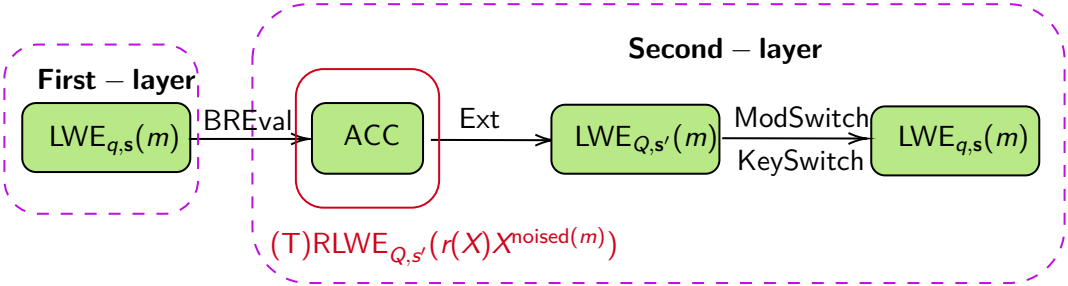
Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting



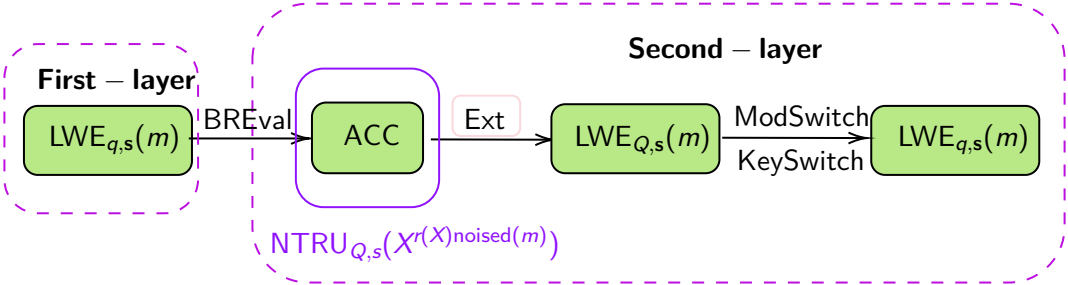
Bootstrapping for LWE

- Overview of bootstrapping in RLWE/RGSW setting



Bootstrapping for LWE

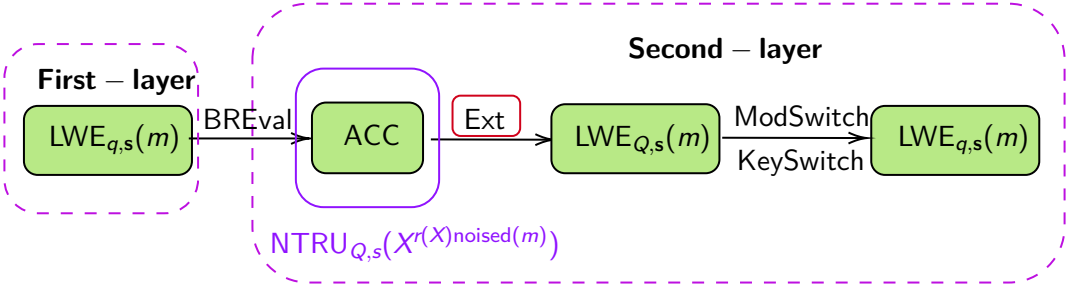
- Our bootstrapping



- Ext: NTRU $\to$ LWE
  - $c = (g + m)/f$, coefficient vectors: $\mathbf{c} = (c_0, \ldots, c_{N-1})$, $\mathbf{f} = (f_0, \ldots, f_{N-1})$
  - extract $\mathrm{LWE}_{Q,\mathbf{f}}(m) = (\hat{\mathbf{c}} = (c_0, -c_{N-1}, \ldots, -c_1), 0)$ from $c = \mathrm{NTRU}_{Q,f}(m)$

Bootstrapping for LWE

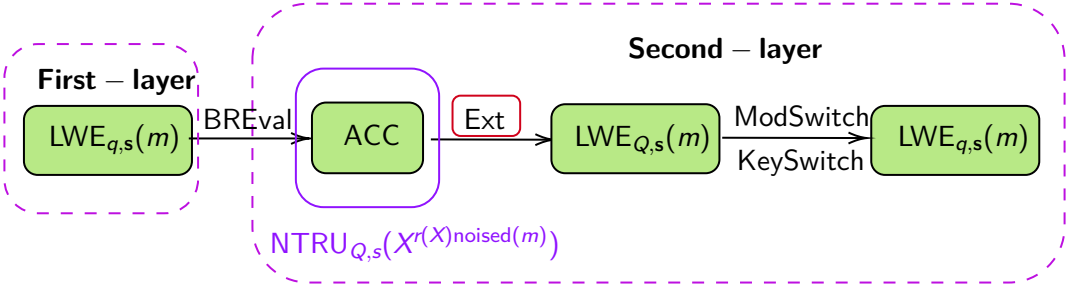
- Our bootstrapping



- Ext: NTRU $\rightarrow$ LWE
  - $c = (g + m)/f$, coefficient vectors: $\mathbf{c} = (c_0, \ldots, c_{N-1})$, $\mathbf{f} = (f_0, \ldots, f_{N-1})$
  - extract $\mathrm{LWE}_{Q,\mathbf{f}}(m) = (\hat{\mathbf{c}} = (c_0, -c_{N-1}, \ldots, -c_1), 0)$ from $c = \mathrm{NTRU}_{Q,f}(m)$

Bootstrapping for LWE

- Our bootstrapping



- Ext: NTRU $\rightarrow$ LWE
  - $c = (g + m)/f$, coefficient vectors: $\mathbf{c} = (c_0, \ldots, c_{N-1})$, $\mathbf{f} = (f_0, \ldots, f_{N-1})$
  - extract $\mathrm{LWE}_{Q,\mathbf{f}}(m) = (\hat{\mathbf{c}} = (c_0, -c_{N-1}, \ldots, -c_1), 0)$ from $c = \mathrm{NTRU}_{Q,f}(m)$

Part VI

Experimental Results

Experimental Results

Table: Parameters for bootstrapping LWE-based first-layer ciphertexts.

Parameters	Key distrib.	n	q	N	Q	B	Q_{ks}	B_{ks}
STD128 [MP21]	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 [MP21]	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

Table: Timings and key sizes for bootstrapping ($\lambda = 128$)

Algorithms	Parameters	Key distrib.	Timings (ms)	EVK (MB)	KSK (MB)	Boots. key (MB)
FHEW/AP	STD128 [MP21]	Ternary	359	1674	224	1898
TFHE/GINX	STD128 [MP21]	Ternary	234	54	224	278
Ours	P128T	Ternary	112	18.65	224	242.65
	P128G	Gaussian	100	17.90	203.44	221.34

Extra 10% improvement over P128T

Experimental Results

Table: Timings and key sizes for bootstrapping ($\lambda = 192$)

Algorithms	Parameters	Key distrib.	Timings (ms)	EVK (MB)	KSK (MB)	Boots. key (MB)
FHEW/AP	STD192 [MP21]	Ternary	1200	6682	532	7214
TFHE/GINX	STD192 [MP21]	Ternary	859	222	532	754
Ours	P192T	Ternary	320	38.10	532	570.10
	P192G	Gaussian	273	34.30	404.41	438.71

Extra 17% improvement over P192T

Conclusion

- An NTRU-based GSW-like scheme with faster external product and key switching
- A new blind rotation technique based on our scheme and ring automorphism
  - Arbitrary secret distribution, small evaluation key
- Faster bootstrapping than TFHE/FHEW
  - Improve parameter by using Gaussian distribution
  - Large secret distribution offers better security

References I

Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, et al.
Homomorphic encryption standard.
In *Protecting Privacy through Homomorphic Encryption*, pages 31–62. Springer, 2021.

Martin R Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer.
Revisiting the expected cost of solving uSVP and applications to LWE.
In *ASIACRYPT 2017*, pages 297–322. Springer, 2017.

Martin R Albrecht.
On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL.
In *EUROCRYPT 2017*, pages 103–129. Springer, 2017.

Jacob Alperin-Sheriff and Chris Peikert.
Faster bootstrapping with polynomial error.
In *CRYPTO 2014*, pages 297–314. Springer, 2014.

References II

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan.
(leveled) fully homomorphic encryption without bootstrapping.
In *ITCS 2012*, pages 309–325. ACM, 2012.

Charlotte Bonte, Ilia Iliashenko, Jeongeun Park, Hilder VL Pereira, and Nigel P Smart.
Final: Faster fhe instantiated with NTRU and LWE.
pages 188–215, 2022.

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène.
Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE.
In *ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*, pages 377–408. Springer, 2017.

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène.
TFHE: Fast fully homomorphic encryption over the torus.
Journal of Cryptology, 33(1):34–91, 2020.

References III

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song.
Homomorphic encryption for arithmetic of approximate numbers.
In *ASIACRYPT 2017*, pages 409–437. Springer, 2017.

Léo Ducas and Daniele Micciancio.
FHEW: bootstrapping homomorphic encryption in less than a second.
In *EUROCRYPT 2015*, volume 9056, pages 617–640. Springer, 2015.

Thomas Espitau, Antoine Joux, and Natalia Kharchenko.
On a dual/hybrid approach to small secret LWE: A dual/enumeration technique for learning with errors and application to security estimates of fhe schemes.
In *INDOCRYPT 2020*, pages 440–462. Springer, 2020.

Junfeng Fan and Frederik Vercauteren.
Somewhat practical fully homomorphic encryption.
Cryptology ePrint Archive, 2012.

References IV

Nicolas Gama, Malika Izabachene, Phong Q Nguyen, and Xiang Xie.
Structural lattice reduction: generalized worst-case to average-case reductions and homomorphic cryptosystems.
In *EUROCRYPT 2016*, pages 528–558. Springer, 2016.

Kamil Kluczniak.
NTRU-ν-um: Secure fully homomorphic encryption from NTRU with small modulus.
pages 1783–1797, 2022.

Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo.
Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption.
pages 227–256, 2023.

References V

Daniele Micciancio and Yuriy Polyakov.
Bootstrapping in FHEW-like cryptosystems.
In *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 17–28, 2021.

Hilder Vitor Lima Pereira.
Bootstrapping fully homomorphic encryption over the integers in less than one second.
In *PKC 2021*, pages 331–359. Springer, 2021.

Yongha Son and Jung Hee Cheon.
Revisiting the hybrid attack on sparse secret LWE and application to HE parameters.
In *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 11–20, 2019.