

FINAL全同态加密方案的自举优化技术

赵秀凤 吴蒙* 宋巍涛

(信息工程大学密码工程学院 郑州 450001)

摘要: 自举是实现全同态加密的有效方法,同时也是影响全同态加密效率的关键环节。FINAL方案是2022年亚密委会提出的全同态密码方案,比TFHE方案自举速度快28%,可以进行高效的同态布尔运算。自举主要包括盲旋转算法和密钥转换算法。针对盲旋转算法,该文提出累加器压缩方法,即对基于容错学习(LWE)的加密方案的密钥生成引入块二进制分布,利用块二进制密钥特性,使得密钥的每个分块只需进行1次外积运算,减少盲旋转算法所需的外积数量。针对密钥转换算法,给出了密钥复用技术,即在生成NGS密钥时复用LWE密钥且复用部分不参与密钥转换的密钥生成,减小了密钥转换密钥规模,进而减少密钥转换算法运算次数,提高了密钥转换算法的效率。分析表明,在安全性相当的情况下,优化的FINAL方案自举所需要执行的外积数量和快速傅里叶变换的数量分别由610和3 940减少到305和1 970,数量上优化50%。密钥转换密钥规模由11 264减少到4 554,密钥转换中标量乘法以及标量加法的运算次数大约由 13.8×10^6 减少到 5.6×10^6 ,密钥转换的密钥规模和计算开销均优化约60%。

关键词: 全同态加密; FINAL; 自举; 盲旋转; 密钥转换

中图分类号: TN918.1

文献标识码: A

文章编号: 1009-5896(2025)07-2183-11

DOI: 10.11999/JEIT241036

CSTR: 32379.14.JEIT241036

1 引言

随着大数据和人工智能的快速发展,数据安全与隐私保护问题日益凸显。全同态加密作为一种允许对加密数据执行任意运算而无需解密的密码体制,被广泛应用于解决隐私计算问题,例如基于多方安全计算的联合统计、基于隐私信息检索的联合查询等。

2009年Gentry^[1]提出了首个同时支持密文状态下加法和乘法运算的全同态加密方案。这一突破性工作开辟了密码学研究新方向,随后涌现出一系列改进方案,推动了全同态加密这一领域的持续发展。典型的全同态加密方案有BGV^[2], B/FV^[3,4], GSW^[5], CKKS^[6]和FHEW/TFHE^[7,8],这些方案的安全性依赖于容错学习(Learning With Errors, LWE)问题及其环变体(Ring Learning With Errors, RLWE)。其中,TFHE方案与其它全同态加密方案相比,优势在于能够进行任意布尔门计算,且能够更快速的自举。除此之外,TFHE还在可编程的自举^[9-11]和多密钥的TFHE^[12,13]方面得到了进一步的发展。可编程自举在降低噪声的同时,还可以利用查找表(Look-Up Table, LUT)的方式同态运算一个单变量函数(包括非线性函数),提高了TF-

HE的计算效率。多密钥的TFHE方案可以对不同密钥下的密文进行同态计算,能够在多方且没有可信第三方的情况下聚合和分析数据。

近年来,针对全同态加密自举环节的性能优化研究有了一系列突破性成果。2022年,Bonte等人^[14]提出FINAL方案,其自举NGS方案是基于NTRU(Nthdegree Truncated polynomial Ring Unit)问题的GSW变体,自举效率优于TFHE方案。同年,Kluczniak^[15]提出了一种基于NTRU的TFHE变体的全同态加密方案NTRU- μ -um,支持对有限域上的任何函数进行同态计算且在效率上优于TFHE。Lee等人^[16]提出了利用环自同构和密钥转换来进行盲旋转的新技术,即LMKC+盲旋转算法,实现了更小的公钥规模和更快的盲旋转。2023年, Lee等人^[17]还在TFHE方案上提出多种自举优化技术,在相同的安全级别下,TFHE自举的执行时间以及自举密钥尺寸得到了优化。2023年美密会, Xiang等人^[18]基于NTRU假设提出了GSW类加密方案,该方案可对环自同构进行快速的密钥转换,并利用环自同构以及密钥转换提出了一种新的盲旋转算法,进而构建了新的自举算法。2024年欧密会, Ma等人^[19]提出了将BGV自举中的数论变换(Number Theoretic Transform, NTT)分解为多个线性子变换的具体方法,分解的子变换可以基于快速NTT算法高效地进行同态运算,同时还对自变换的运算进行创新优化,显著提高自举效率。在同一会议上, Wang等人^[20]重新设计了电路自举的工作流程同时改进了LMKC+盲旋转算法,提出了新的电路自举算法,性能上优于当前TFHE方法。

收稿日期: 2024-11-22; 改回日期: 2025-05-19; 网络出版: 2025-06-24

*通信作者: 吴蒙 18733233053@163.com

基金项目: 国家密码科学基金(2025NCSF02044), 先进计算与智能工程(国家级)实验室基金(2023-LYJJ-01-002)

Foundation Items: The National Cryptologic Science Fund of China (2025NCSF02044), The Fund of Laboratory for Advanced Computing and Intelligence Engineering (2023-LYJJ-01-002)

上述优化大多是对BGV, TFHE方案的自举优化, FINAL方案自举效率优于TFHE方案且当前对FINAL方案的自举优化工作较少。因此, 本文研究FINAL方案上的自举优化技术, FINAL方案自举主要由盲旋转和密钥转换两个模块组成, 通过对这两个模块进行优化, 提高FINAL方案自举的效率。

一方面, 提出了累加器压缩的盲旋转算法。采用块二进制分布生成LWE密钥, 利用该密钥每个分块的汉明重量最多为1的特性, 对累加器进行压缩, 使得每个分块只需进行1次外积。与原盲旋转算法相比, 外积次数以及快速傅里叶变换(Fast Fourier Transform, FFT)的次数优化50%。

另一方面, 针对密钥转换模块, 提出了密钥复用技术。具体来说, NGS密钥中的 n 维分量可以由LWE密钥表示。因此, 在实际生成NGS密钥时, 只需生成其中的 $(N-n)$ 维, 密钥转换密钥规模减小, 进而密钥转换算法的计算开销降低, 其中, n 为LWE密钥维数, N 为NGS密钥的维数。与原密钥转换算法相比, 存储开销以及计算开销降低近60%。

最后, 将具体参数带入优化方案进行具体的性能分析, 与原FINAL方案相比, 每次自举所需的外积及快速傅里叶变换次数减少50%, 密钥转换密钥的尺寸以及密钥转换的运算减少60%。

2 基础知识

2.1 符号

粗斜体小写字母代表向量, 粗斜体大写字母代表矩阵, \mathbb{Z} 表示整数环, $R := \mathbb{Z}[X]/\langle X^N + 1 \rangle$, $R_Q := \mathbb{Z}_Q[X]/\langle X^N + 1 \rangle$, 其中, N 是2的幂。 \mathbb{B} 表示集合 $\{0, 1\}$, $\mathbb{B}_N[X]$ 表示次数小于 N 且系数在 \mathbb{B} 中的多项式集合。 $\langle \mathbf{a}, \mathbf{s} \rangle$ 表示向量 \mathbf{a} 和 \mathbf{s} 的内积, $\mathbf{c} \odot \mathbf{c}$ 表示NGS标量密文 c 与NGS向量密文 \mathbf{c} 的外积。 $\lceil \cdot \rceil$ 表示舍入符号(四舍五入), $\lfloor \cdot \rfloor$ 表示向下取整符号。

对于分布 \mathcal{D} , 用 $x \leftarrow \mathcal{D}$ 表示 \mathcal{D} 的随机抽样。 \mathcal{D}_σ 表示均值为0, 方差为 σ^2 的高斯分布。对于集合 S , 定义 S 上的均匀分布为 $\mathcal{U}(S)$ 。 $\mathbf{B}_{l,k}$ 表示 \mathbb{B}^n 上的块二进制分布, 对于块二进制密钥 \mathbf{s} , 定义 l 为密钥分块尺寸, k 为分块数量, 即 $n = l \cdot k$ 。

定义 $\phi(f)$ 为多项式 f 的系数向量, $\Phi(f)$ 是关于 f 的反循环矩阵, 即若 $f = f_0 + f_1 X + \dots + f_{N-1} X^{N-1}$, 则 $\phi(f) = (f_0, f_1, \dots, f_{N-1})$

$$\Phi(f) = \begin{pmatrix} f_0 & f_1 & \dots & f_{N-2} & f_{N-1} \\ -f_{N-1} & f_0 & \dots & f_{N-3} & f_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -f_2 & -f_3 & \dots & f_0 & f_1 \\ -f_1 & -f_2 & \dots & -f_{N-1} & f_0 \end{pmatrix} \quad (1)$$

2.2 LWE问题

LWE问题^[21], 设 $n > 0$, $q > 1$ 是整数, 秘密向量 $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{A} \in \mathbb{Z}_q^{x \times n}$ 服从均匀分布, 已知样本 $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{x \times n} \times \mathbb{Z}_q^x$, 定义LWE样本为存在一个秘密向量 $\mathbf{s} \in \mathbb{Z}_q^n$, 使得 $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q$, 其中, 误差 $\mathbf{e} \leftarrow \mathcal{D}_\sigma^x$ 。

(n, q, σ, x) -LWE计算问题: 给定LWE样本的情况下, 求解秘密向量 \mathbf{s} 。

(n, q, σ, x) -LWE判定问题: 给定一个矩阵 $\mathbf{A} \in \mathbb{Z}_q^{x \times n}$ 和一个向量 \mathbf{b} 的情况下, 区分 \mathbf{b} 是LWE样本还是均匀随机样本。

2.3 LWE基本加密方案

LWE.SecretKeyGen(1^λ): 输入安全参数 λ , 输出密钥 $\mathbf{s} \in \mathbb{Z}_q^n$ 。

LWE.Enc(\mathbf{s}, m): 输入密钥 $\mathbf{s} = (s_0, s_1, \dots, s_{n-1})$, 消息 $m \in \{0, 1\}$, 输出密文 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$, 其中, $\mathbf{a} = (a_0, a_1, \dots, a_{n-1}) \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{e} \leftarrow \mathcal{D}_\sigma$, $b = -\langle \mathbf{a}, \mathbf{s} \rangle + \lceil q/4 \rceil \cdot m + \mathbf{e} \pmod q$ 。

LWE.Dec(\mathbf{s}, \mathbf{c}): 输入密钥 \mathbf{s} 和密文 $\mathbf{c} = (b, \mathbf{a})$, 输出明文 \mathbb{Z} 。

对于LWE加密, 定义 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$ 的相位为 $\varphi_s(b, \mathbf{a}) = b + \langle \mathbf{a}, \mathbf{s} \rangle \pmod q$ 。

2.4 NTRU问题

NTRU问题^[14], 设 $N > 0$, $Q > 1$ 是整数, $\sigma > 0$ 为实数, $g, f \leftarrow \mathcal{D}_\sigma^N$ 且 f 在 R_Q 上可逆。

(N, Q, σ) -NTRU计算问题是在给定 $h := g \cdot f^{-1} \pmod Q$ 的情况下恢复 f 和 g 。

(N, Q, σ) -NTRU判定问题是区分 h 分布和 R_Q 上的均匀分布。

2.5 NGS加密方案

NGS^[14]是基于NTRU的类GSW方案, 与GSW方案一样具有准加性噪声增长, 被用作自举的累加器。NGS加密方案具有两个加密函数, 一个加密的明文是 R_Q 上的3元多项式, 另一个加密的明文是 R_Q 上的向量, 该方案具体描述如下。

NGS.ParamGen(1^λ): 输入安全参数 λ , 输出 (N, Q, σ, B, ℓ) , 其中, B 是小工具向量的分解基, $\ell = \log_B(Q)$ 。

NGS.KeyGen(N, σ): 输入参数 N, σ , 输出 $\text{sk} = f$, sk 为NGS加密方案的私钥。其中, $f = 1 + 4f'$ 且 f^{-1} 在 R_Q 中存在, $f' \leftarrow \mathcal{D}_\sigma^N$ 。

NGS.EncS(sk, m): 输入 (sk, m) , 输出 $c = g/f + \Delta \cdot m \in R_Q$, 其中, m 是待加密的3元多项式明文, $g \leftarrow \mathcal{D}_\sigma^N$, $R := \mathbb{Z}[X]/\langle X^N + 1 \rangle$, c 为 m 的标量加密。

NGS.EncVec(sk, m): 输入 (sk, m) , 输出

$c = g/f + g \cdot m \in R_Q^\ell$, 其中, $g = (g_0, g_1, \dots, g_{\ell-1})$, $g_i \leftarrow \mathcal{D}_\sigma^N$, $0 \leq i \leq \ell - 1$, $m \in \mathbb{M} = \{\pm b \cdot X^k : b \in \{0, 1\}, k \in \mathbb{N}\}$ 是一个单项式, c 为 m 的向量加密。

定义1^[14] 设标量密文为 $c = g/f + \Delta \cdot \mu \in R_Q$, 其中, μ 是一个3元多项式, 向量密文为 $c = g/f + g \cdot v \in R_Q^\ell$, 其中 $v \in \mathbb{M}$, 定义 c 和 c 的外积

$$c \odot c = g^{-1}(c) \cdot c \in R_Q \quad (2)$$

即 $c \odot c = g^{-1}(c) \cdot g/f + g^{-1}(c)g \cdot v = (g^{-1}(c) \cdot g + g \cdot v)/f + \Delta \cdot \mu \cdot v$ 。

2.6 模转换

定义2^[14] 设 $Q, q \in \mathbb{Z}$ 且 $1 < q < Q$, 定义函数 $[\cdot]_{Q;q} : \mathbb{Z}_Q \rightarrow \mathbb{Z}_q$ 为 $[z]_{Q;q} = [q \cdot z/Q] + \text{Ber}$, $\text{Ber} \in \{0, 1\}$ 是一个伯努利随机变量, 其概率为 $\Pr[\text{Ber} = 1] = (q \cdot z/Q) - [q \cdot z/Q] \in \{0, 1\}$ 。将上述定义拓展到多项式的形式, 即

$$\text{ModSwitch}(c) = \sum_{i=0}^{N-1} [c_i]_{Q;q} X^i \in R_q \quad (3)$$

2.7 密钥转换

密钥转换是FINAL方案的重要模块之一, 其功能是将经过盲旋转和模转换后生成的NGS标量密文 $c = g/f + \varepsilon + \Delta \cdot m \in R_q$ 转换为LWE密文, 其中, ε 是模转换后的误差。密钥转换分为密钥转换密钥的生成算法和密钥转换算法。

密钥转换密钥生成算法: $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = \text{SwitchKeyGen}(s, f_0)$, 输入LWE密钥 s , NGS密钥的反循环矩阵的第1列向量 f_0 以及参数 $q, B_{\text{ksk}}, \sigma_e$, 输出密钥转换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = (A, b = A \cdot s + e + P \cdot f_0)$, 其中, $A \in \mathbb{Z}_q^{(N \cdot L) \times n}$, $f_0 = \text{col}_0(\Phi(f)) \in \mathbb{Z}^N$, $e \leftarrow \chi_{\sigma_e}^{N \cdot L}$, $L = \log_{B_{\text{ksk}}}(q)$, $P = I_N \otimes g_{q, B_{\text{ksk}}} \in \mathbb{Z}_q^{(N \cdot L) \times N}$, 即 P 的每个“对角元素”等于 $g_{q, B_{\text{ksk}}} \in \mathbb{Z}_q^L$ 。

密钥转换算法: $\text{KeySwitch}_{\text{NTRU} \rightarrow \text{LWE}}(c, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}})$, 输入密钥转换密钥和NGS密文, 输出LWE密文。

$\text{KeySwitch}_{\text{NTRU} \rightarrow \text{LWE}}(c, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}})$:

步骤1 分解 c 的系数向量得到 $y := g^{-1}(\phi(c)) \in \mathbb{Z}^{N \cdot L}$;

步骤2 $a \leftarrow y \cdot A$;

步骤3 $b \leftarrow y \cdot b$;

步骤4 输出 $c' = (a, b) \in \mathbb{Z}_q^{n+1}$ 。 $b = (y \cdot A \cdot s + y \cdot e + y \cdot P \cdot f_0) = a \cdot s + y \cdot e + g_0 + \varepsilon \cdot ((1, \theta) + 4\phi(f')) + 4 \cdot \varepsilon' \cdot \phi(m) \cdot \phi(f') + \Delta \cdot m_0$, 其中, $\varepsilon, \varepsilon' \in (-1/2, 1/2]$, m_0 是 m 的常数项, g_0 为 g 的常数项。 c' 是加密明文为 m_0 的LWE密文。

2.8 FINAL方案

本节给出FINAL方案^[4]的自举算法如算法1所示。

2.9 块二进制分布

设 l 和 k 为正整数。首先, 将 B_l 定义为 \mathbb{Z}^l 上的一个分布, 该分布等概率地对 \mathbb{Z}^l 中的一个标准单位向量或0向量进行采样, 即 $(0, 0, \dots, 0), (1, 0, \dots, 0), (0, 1, \dots, 0), \dots, (0, 0, \dots, 1) \in \mathbb{B}^l$, 该分布可以返回 $(l+1)$ 个向量, 且返回每个向量的概率为 $1/(l+1)$ 。然后, 通过连接 k 个 B_l 的实例来对 B_l 进行扩展, 定义了 \mathbb{B}^n 上的块二进制分布, 记作 $B_{l,k}$, 即 $(s_0, s_1, \dots, s_{k-1}) \in \mathbb{Z}^n$, 采样 s_0, s_1, \dots, s_{k-1} 相互独立于 B_l , 其中, $n = l \cdot k$ 。块二进制分布的向量在每个长度为 l 的块中最多有一个分量为1。

关于块二进制密钥分布的安全性, Goldwasser 等人^[22]证明了二进制密钥的LWE问题是安全的。此外, 他们指出, 安全性证明适用于具有足够大的最小熵的任意二进制密钥分布的LWE。

定理1^[22] 设 $n, q \geq 1$ 为整数, \mathcal{D} 为 $\{0, 1\}^n$ 上的任意分布且最小熵 $\geq k$, $\alpha, \beta > 0$, 使得 $\alpha/\beta = \text{negl}(n)$, 其中, α, β 是误差分布的标准差。则对于任意 $l \leq k - \omega \log_q n$, 存在一个从 $\text{DLWE}_{l,q,\alpha}(\mathcal{U}(\mathbb{Z}_q^l))$ 到 $\text{DLWE}_{n,q,\beta}(\mathcal{D})$ 的多项式时间的归约。

当最小熵大于证明要求时, 块二进制分布LWE的困难性得到保证。在实际应用中, 安全级别是通过具体攻击来估计的, 3.4节考虑到块二进制密钥分布, 采用攻击算法^[23-25]对本文方案进行安全评估。

3 FINAL自举优化

盲旋转算法和密钥转换算法是影响FINAL方案自举效率的关键, 两个算法都需要大量的多项式运算, 标量加法以及标量乘法。本节对FINAL方

算法1 自举算法

输入: 盲旋转密钥BRK,

加密消息为 m 的LWE密文 $c = (b, a) \in \mathbb{Z}_q^{n+1}$,

密钥转换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}$ 。

输出: 新的LWE密文 c' , 加密消息为 m 。

(1) $[2N \cdot c/q] = (\bar{b}, \bar{a}_0, \bar{a}_1, \dots, \bar{a}_{n-1})$

(2) $\text{ACC} \leftarrow [Q/8] \cdot X^{(N/2)+\bar{b}} \cdot \sum_{i=0}^{N-1} X^i$

(3) for $0 \leq j < n$ do

(4) $\text{ACC} \leftarrow \text{ACC} + [\text{ACC} \cdot (X^{\bar{a}_j} - 1)] \odot \text{BRK}_i$ (4)

(5) end for

(6) $\text{ACC} \leftarrow \text{ACC} + [Q/8] \cdot \sum_{i=0}^{N-1} X^i$

(7) $\text{ACC} \leftarrow \text{ModSwitch}(\text{ACC})$

(8) $c' \leftarrow \text{KeySwitch}_{\text{NTRU} \rightarrow \text{LWE}}(\text{ACC}, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}})$

(9) return c'

案的盲旋转和密钥转换两个算法进行了优化。针对盲旋转模块,采用块二进制分布^[20]对LWE密钥进行采样,利用该密钥特性,压缩盲旋转的累加器,减少盲旋转迭代次数。同时,针对密钥转换模块,引入密钥复用技术,对FINAL方案密钥转换密钥的存储开销以及密钥转换过程的计算开销进行了优化,并给出了优化的盲旋转以及密钥转换的算法描述和性能分析,最后给出优化后自举的算法描述以及安全性分析。

3.1 基于压缩累加器的盲旋转优化技术

本节对优化的盲旋转算法进行了具体描述以及性能分析。

3.1.1 盲旋转优化

设 $\mathbf{c} = (\mathbf{b}, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$ 为密钥 $\mathbf{s} \in \mathbb{B}^n$ 下的LWE密文。盲旋转过程旨在生成 $v \cdot X^{\varphi_{\mathbf{s}}(\bar{\mathbf{b}}, \bar{\mathbf{a}})}$ 的NGS标量加密,其中, $\varphi_{\mathbf{s}}(\bar{\mathbf{b}}, \bar{\mathbf{a}}) = \bar{\mathbf{b}} + \langle \mathbf{s}, \bar{\mathbf{a}} \rangle \pmod{2N}$ 。在原FINAL方案^[14]中,上述计算是通过将累加器ACC初始化为 $X^{\bar{\mathbf{b}}} \cdot v$ 的平凡NGS标量加密,然后再与 $X^{\bar{\mathbf{a}}_0 \cdot s_0}, X^{\bar{\mathbf{a}}_1 \cdot s_1}, \dots, X^{\bar{\mathbf{a}}_{n-1} \cdot s_{n-1}}$ 同态相乘实现。其中,每个同态乘法都被表示为一个NGS的标量密文与 $X^{\bar{\mathbf{a}}_i \cdot s_i} = 1 + (X^{\bar{\mathbf{a}}_i} - 1) \cdot \text{BRK}_i$ 的外积。

现在假设LWE密钥 $\mathbf{s} = (s_0, s_1, \dots, s_{n-1})$ 根据 $\mathbf{B}_{l,k}$ 进行采样,使每个块 $x \leftarrow \mathcal{D}$ 中最多有一个分量为1,其中, $I_j = \{jl, jl+1, \dots, j(l+1)-1\}$, $0 \leq j < k$ 。因为 $X^{\bar{\mathbf{a}}_i \cdot s_i} = 1 + (X^{\bar{\mathbf{a}}_i} - 1) \cdot s_i$, 故

$$X^{\sum_{i \in I_j} \bar{\mathbf{a}}_i \cdot s_i} = 1 + \sum_{i \in I_j} (X^{\bar{\mathbf{a}}_i} - 1) \cdot s_i \quad (5)$$

若存在 $i \in I_j$, 使得 $s_i = 1$, 则等式两边为 $X^{\bar{\mathbf{a}}_i}$; 反之,若对于所有 $i \in I_j$, $s_i = 0$, 等式两边等于1。

算法2是基于原FINAL方案的盲旋转的优化,利用式(5),将累加器式(4)替换为 $X^{\sum_{i \in I_j} \bar{\mathbf{a}}_i \cdot s_i}$ 与ACC的 k 次迭代相乘式(6),即NewBlindRotate算法需要 k 个外积,与原方案盲旋转相比减少了 $n-k$ 个外积,提高了盲旋转效率。

算法 2 NewBlindRotate(BRK, \mathbf{c})

输入: 盲旋转密钥BRK和一个LWE密文 $\mathbf{c} = (\mathbf{b}, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$

输出: NTRU(NGS)密文 $c = g/f + \epsilon + \Delta \cdot m \in R_q$

(1) $[2N \cdot \mathbf{c}/q] = (\bar{\mathbf{b}}, \bar{\mathbf{a}}_0, \bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_{n-1})$

(2) $\text{ACC} \leftarrow [Q/8] \cdot X^{(N/2)+\bar{\mathbf{b}}} \cdot \sum_{i=0}^{N-1} X^i$

(3) **for** $0 \leq j < k$ **do**

(4) $\text{ACC} \leftarrow \text{ACC} + \text{ACC} \odot [\sum_{i \in I_j} (X^{\bar{\mathbf{a}}_i} - 1) \cdot \text{BRK}_i]$ (6)

(5) **end for**

(6) **return** ACC

3.1.2 盲旋转性能分析

FINAL方案的外积运用到了工具向量的分解、标量乘法以及快速傅里叶变换,其中,快速傅里叶变换是自举中代价最大的部分,用于多项式乘法前后,将多项式转变成FFT形式,提高多项式乘法的效率,或将FFT形式转换为多项式的向量形式,保持结构的统一。

首先分析原FINAL方案中自举所需要的FFT转换以及标量乘法的数量。盲旋转的每次迭代需要计算

$$\text{ACC} \leftarrow \text{ACC} + [\text{ACC} \cdot (X^{\bar{\mathbf{a}}_j} - 1)] \odot \text{BRK}_i \quad (7)$$

$\text{ACC} \cdot (X^{\bar{\mathbf{a}}_j} - 1)$ 以及 $\mathbf{g}^{-1}(\text{ACC} \cdot (X^{\bar{\mathbf{a}}_j} - 1))$ 的计算不需要快速傅里叶变换,盲旋转密钥 BRK_i 可以进行FFT预处理,因此只需要考虑计算 $\mathbf{g}^{-1}(\text{ACC} \cdot (X^{\bar{\mathbf{a}}_j} - 1)) \cdot \text{BRK}_i$ 以及将FFT形式的计算结果转变为向量形式,故,每次外积运算需要 $\ell+1$ 次FFT变换, $\ell \cdot N$ 次标量乘法,其中, ℓ 是 BRK_i 的长度。又因为需要原FINAL方案的盲旋转需要进行 n 次外积,所以总共需要 $(\ell+1) \cdot n$ 次FFT变换, $\ell \cdot N \cdot n$ 次标量乘法。

优化的盲旋转方案只需要进行 k 次同态乘法的迭代运算,即需要进行 k 次外积计算,计算

$$\text{ACC} \leftarrow \text{ACC} + \text{ACC} \odot \left[\sum_{i \in I_j} (X^{\bar{\mathbf{a}}_i} - 1) \cdot \text{BRK}_i \right] \quad (8)$$

也可以写成计算

$$\text{ACC} \leftarrow \text{ACC} + \left[\sum_{i \in I_j} (X^{\bar{\mathbf{a}}_i} - 1) \cdot (\text{ACC} \odot \text{BRK}_i) \right] \quad (9)$$

对于所有的 $i \in I_j$, $\text{ACC} \odot \text{BRK}_i = \mathbf{g}^{-1}(\text{ACC}) \cdot \text{BRK}_i$, 因此本文在 $i \in I_j$ 时,只需要对 $\mathbf{g}^{-1}(\text{ACC})$ 进行1次FFT转换, BRK_i 进行FFT转换预处理,即在进行计算时只需要重用 $\mathbf{g}^{-1}(\text{ACC})$ 的FFT形式以及调用 BRK_i 的FFT形式。故,每进行1次同态乘法迭代运算仍需要 $\ell+1$ 次FFT变换,但需要的标量乘法次数变为 $\ell \cdot N \cdot l$, 其中, l 为块二进制密钥的分块尺寸,即 $k \cdot l = n$ 。又因为优化后的盲旋转方案只需要进行 k 次同态乘法的迭代运算,因此,优化后的盲旋转总共需要 $(\ell+1) \cdot k$ 次FFT变换,需要 $\ell \cdot N \cdot l \cdot k = \ell \cdot N \cdot n$ 次标量乘法,与原方案对比见表1。

表 1 盲旋转算法的性能分析

方案	FFT	标量乘法
FINAL ^[14]	$(\ell+1) \cdot n$	$\ell \cdot N \cdot n$
本文方案	$(\ell+1) \cdot k$	$\ell \cdot N \cdot n$

3.2 基于密钥复用的密钥转换优化技术

本节对优化的密钥转换算法进行了具体描述以及性能分析。

3.2.1 密钥转换优化

FINAL方案的密钥转换操作旨在将一个密钥 f 下的NTRU/NGS密文 $c = g/f + \epsilon + \Delta \cdot m \in R_q$ 转换为一个密钥 s 下的LWE密文 $(b', a') = (b', a_0', \dots, a_{n-1}')$ ，使得 $b' + \langle s, a' \rangle \approx [q/4] \cdot m_0 \text{mod} q$ ， m_0 是 m 的常数项。为了加快密钥转换速度，使用LWE密钥 $s = (s_0, \dots, s_{n-1})$ 作为NGS密钥 f_0 的一部分，即

$$\begin{aligned} f_0 \cdot \text{col}_0(\Phi(f)) &= (1 + 4s_0, 4s_1, \dots, 4s_{n-1}, 4f'_n, \dots, \\ &\quad 4f'_{N-1}) \\ &= (4s, f') + (1, 0, \dots, 0) \\ &= 1 + 4f'_0 \in \mathbb{Z}^N \end{aligned} \quad (10)$$

使得优化的密钥转换算法可以省略 n 维的运算，获得约 $(N - n)/N$ 的速度提升。

优化前后的密钥转换对比如图1所示，其中， f_0 为NGS密钥 f 的反循环矩阵的第1列。优化的密钥转换算法具体描述如下：

NewSwitchKeyGen(s, f'): 输入LWE密钥 s ，NGS密钥的反循环矩阵第1列向量的后 $(N - n)$ 维向量 f' 以及参数 $q, B_{\text{ksk}}, \sigma_e$ ，输出密钥转换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = (A_1, b_1 = A_1 \cdot s + e_1 + P_1 \cdot f')$ ，与

原方案密钥转换密钥对比，密钥尺寸减少了 $(N - n)/N$ ，如图2所示。其中， $A_1 \in \mathbb{Z}_q^{((N-n) \cdot L) \times n}$ ， $e_1 \leftarrow \chi_{\sigma_e}^{(N-n) \cdot L}$ ， $s_i \leftarrow B_{l,k}$ ， $P_1 = I_{(N-n)} \otimes g_{q, B_{\text{ksk}}}$ ， $f'_j \leftarrow \mathcal{U}(\{0, 1, -1\})$ ， $n \leq j < N$ ， $(A, b) \in \mathbb{Z}_q^{(N \cdot L) \times (n+1)}$ 为原FINAL方案密钥转换密钥。

NewKeySwitch_{NTRU→LWE}($\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, c$): 输入密钥转换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = (A_1, b_1)$ 和NGS密文 c ，输出LWE密文。

将 c 的系数向量分为 $\phi(c)_0 \in \mathbb{Z}_q^n$ 和 $\phi(c)_1 \in \mathbb{Z}_q^{N-n}$ 两部分，如图3所示，即 $\phi(c) = (\phi(c)_0, \phi(c)_1)$ 。

NewKeySwitch_{NTRU→LWE}($\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, c$):
 步骤1 分解 c 的部分系数向量得到 $y_1 := g^{-1}(\phi(c)_1) \in \mathbb{Z}^{(N-n) \cdot L}$;
 步骤2 $a' \leftarrow y_1 \cdot A_1$;
 步骤3 $b' \leftarrow y_1 \cdot b_1$;
 步骤4 输出 $c' = (a, b) = (a', b') + (0, 4\phi(c)_0 \cdot s + c_0) \in \mathbb{Z}_q^{n+1}$ 。

若优化后密钥转换 $(b - a \cdot s)$ 的值与原方案的值相等，则能正确解密，优化后密钥转换的正确性分析为

$$b = y_1 \cdot b_1 + 4\phi(c)_0 \cdot s + c_0 = y_1 \cdot A_1 \cdot s + y_1 \cdot e_1 + y_1 \cdot P_1 \cdot f' + 4\phi(c)_0 \cdot s + c_0 \quad (11)$$

$$b - a \cdot s = y_1 \cdot e_1 + \phi(c)_1 \cdot f' + 4\phi(c)_0 \cdot s + c_0 = y_1 \cdot e_1 + \phi(c) \cdot f_0 \quad (12)$$

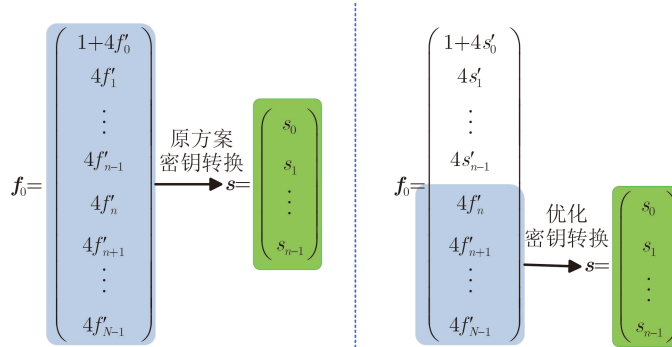


图1 优化前后的密钥转换

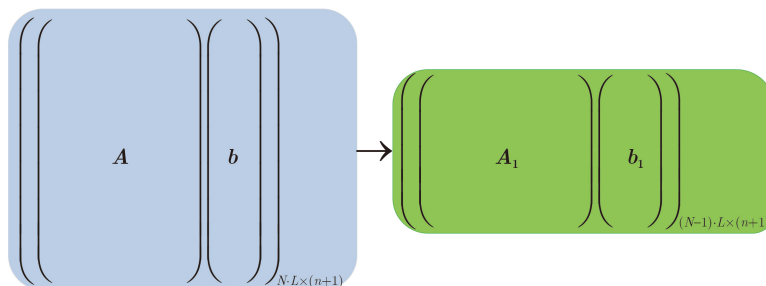


图2 优化前后的密钥转换密钥

$$\phi(c) = (\underbrace{c_0 \quad c_1 \quad \dots \quad c_{n-1}}_{\phi(c)_0} \quad \underbrace{c_n \quad c_{n+1} \quad \dots \quad c_{N-1}}_{\phi(c)_1})$$

图3 c的系数向量

而原方案 $b - \mathbf{a} \cdot \mathbf{s} = \mathbf{y} \cdot \mathbf{e} + \phi(c) \cdot \mathbf{f}_0$ ，因此，新的密钥转换算法可以正常解密，即 \mathbf{c}' 是 m 常数项的LWE加密密文。

3.2.2 性能分析

本节从密钥转换的密钥规模以及密钥转换的计算两个角度对其性能进行分析，如表2所示。

首先，对密钥转换的密钥规模进行分析，原FINAL方案密钥转换密钥为

$$\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = (\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} + \mathbf{P} \cdot \mathbf{f}_0) \quad (13)$$

其中， $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{(N \cdot L) \times (n+1)}$ ，不妨将之视为 $N \cdot L$ 个LWE密文。优化的密钥转换方案的密钥变为

$$\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} = (\mathbf{A}_1, \mathbf{b}_1 = \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1 + \mathbf{P}_1 \cdot \mathbf{f}') \quad (14)$$

其中， $(\mathbf{A}_1, \mathbf{b}_1) \in \mathbb{Z}_q^{((N-n) \cdot L) \times (n+1)}$ ，即为 $(N-n) \cdot L$ 个LWE密文。

其次，对密钥转换的计算进行分析，原密钥转换算法需要计算 $(\mathbf{a}, \mathbf{b}) = (\mathbf{y} \cdot \mathbf{A}, \mathbf{y} \cdot \mathbf{b}) \in \mathbb{Z}_q^{n+1}$ ，其中， $\mathbf{y} := \mathbf{g}^{-1}(\phi(c)) \in \mathbb{Z}^{N \cdot L}$ ，计算需要运行 $[N \cdot L \cdot (n+1)]$ 次标量乘法， $[(N \cdot L - 1) \cdot (n+1)]$ 次标量加法。优化的密钥转换算法则计算 $(\mathbf{a}, \mathbf{b}) = (\mathbf{y}_1 \cdot \mathbf{A}_1, \mathbf{y}_1 \cdot \mathbf{b}_1) + (\mathbf{0}, 4\phi(c)_0 \cdot \mathbf{s} + c_0)$ ，其中， $\mathbf{y}_1 := \mathbf{g}^{-1}(\phi(c)_1) \in \mathbb{Z}^{(N-n) \cdot L}$ ，计算需要运行 $[(N-n) \cdot L \cdot (n+1) + n]$ 次标量乘法， $[(N-n) \cdot L - 1] \cdot (n+1) + n$ 次标量加法。

3.3 自举优化

FINAL方案的自举优化主要由盲旋转优化算法NewBlindRotate以及密钥转换优化算法NewKeySwitch_{NTRU→LWE}两部分构成，本节给出了FINAL自举优化后的方案描述以及具体安全性分析。

3.3.1 方案描述

Setup(1^λ): 以安全参数 λ 为输入，生成如下参数：LWE密钥维数 n ，NGS密钥维数 N ，小工具分解基 B 和维数 ℓ ，密钥转换密钥的分解基 B_{ksk} 和分解维数 L 。

KeyGen(1^λ):

(1) 采样 $s_i \leftarrow B_{l,k}$ ， $0 \leq i < n$ ，整数 $l, k > 0$ ，且 $l \cdot k = n$ ，输出LWE密钥 $\mathbf{s} = (s_0, s_1, \dots, s_{n-1})$ 。

(2) 采样 $f'_i \leftarrow \mathcal{U}(\{0, 1, -1\})_{n \leq i < N}$ ， $f' = (s_0, -f'_{N-1}, \dots, -f'_n, -s_{n-1}, \dots, -s_1)$ ，NGS密钥 $f = 1 + 4f'$ 且 f^{-1} 在 R_Q 上存在， $\text{sk} := f = (1 + 4s_0, -4f'_{N-1}, \dots, -4f'_n, -4s_{n-1}, \dots, -4s_1)$ ，则 $\mathbf{f}_0 = \text{col}_0(\Phi(f)) = (1 + 4s_0, 4s_1, \dots, 4s_{n-1}, 4f'_n, \dots, 4f'_{N-1}) = 1 + (4\mathbf{s}, \mathbf{f}')$ ，输出 sk 。

(3) $\text{BRK}_i \leftarrow \text{NGS.EncVec}(\text{sk}, s_i)$ ，输出盲旋转密钥 $\text{BRK} = \{\text{BRK}_i\}_{0 \leq i < n}$ 。

(4) $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}} \leftarrow \text{NewSwitchKeyGen}(\mathbf{s}, \mathbf{f}')$ ，其中， $\mathbf{f}' \in \mathbb{Z}^{N-n}$ ，输出 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}$ 。

LWE.Enc(\mathbf{s}, m): 输入密钥 $\mathbf{s} = (s_0, s_1, \dots, s_{n-1})$ ， $m \in \{0, 1\}$ ， $\mathbf{a} = (a_0, a_1, \dots, a_{n-1}) \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ ， $\mathbf{e} \leftarrow \mathcal{D}_\sigma$ ，输出 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$ ，其中， $b = -\langle \mathbf{a}, \mathbf{s} \rangle + [q/4] \cdot m + \text{emod}q$ 。

LWE.Dec(\mathbf{s}, \mathbf{c}): 输入密钥 \mathbf{s} 和密文 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$ ，输出 $[4 \cdot (b + \langle \mathbf{a}, \mathbf{s} \rangle) / q] \bmod q \bmod 2$ 。

NewBoot($\text{BRK}, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, \mathbf{c}$): 输入盲旋转密钥 BRK ，密钥转换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}$ 和LWE密文 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$ ，输出密文 $\mathbf{c}' \in \mathbb{Z}_q^{n+1}$ 。

NewHomNAND($\text{BRK}, \text{KSK}, \mathbf{c}_1, \mathbf{c}_2$): 输入盲旋转密钥 BRK ，密钥交换密钥 $\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}$ 和LWE密文 $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{Z}_q^{n+1}$ ，输出 $\text{NewBoot}(\text{BRK}, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, \mathbf{c}_{\text{NAND}}) = ([5 \cdot q/8], 0) - \mathbf{c}_1 - \mathbf{c}_2 \in \mathbb{Z}_q^{n+1}$ 。

对优化的自举算法NewBoot($\text{BRK}, \text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, \mathbf{c}$)的正确性分析如下：

当 $\varphi_s(\mathbf{c}_i) \approx (q/4) \cdot m_i \bmod q$ 且 $m = m_1 \bar{\wedge} m_2$ 时， $\varphi_s(\mathbf{c}_{\text{NAND}}) \approx (q/2) \cdot m \bmod q$ 。自举将NAND门运算后生成的密文 \mathbf{c}_{NAND} ，转换为 $\mathbf{c}'_{\text{NAND}}$ ，其中， $\varphi_s(\mathbf{c}'_{\text{NAND}}) \approx (q/4) \cdot m \bmod q$ 。即只需证明输入一个LWE密文 \mathbf{c} ， $\varphi_s(\mathbf{c}) \approx (q/2) \cdot m \bmod q$ ，使得输出的密文 \mathbf{c}' 满足 $\varphi_s(\mathbf{c}') \approx (q/4) \cdot m \bmod q$ ，则NewBoot算法正确。

算法3第1行中ACC的相位近似为 $\mathbf{v} \cdot X^{\bar{b} + \langle \mathbf{s}, \bar{\mathbf{a}} \rangle}$ ，其中， $(\bar{b}, \bar{a}_0, \bar{a}_1, \dots, \bar{a}_{n-1}) = 2N \cdot \mathbf{c} / q$ ， $\mathbf{v} = Q/8 \cdot X^{(N/2)}$ 。 $\sum_{i=0}^{N-1} X^i$ 为测试向量。若 $m = 1$ ，则 $N/2 < \bar{b} + \sum_{i=0}^{n-1} \bar{a}_i \cdot s_i \leq (3 \cdot N)/2$ ， $\mathbf{v} \cdot X^{\bar{b} + \langle \mathbf{s}, \bar{\mathbf{a}} \rangle}$ 的常数项接近 $Q/8$ 。相反，若 $m = 0$ ， $\mathbf{v} \cdot X^{\bar{b} + \langle \mathbf{s}, \bar{\mathbf{a}} \rangle}$ 的常数项接近 $-Q/8$ 。因此， $\text{ACC} \leftarrow \text{ACC} + [Q/8] \cdot \sum_{i=0}^{N-1} X^i$ 的相位常数项接近 $Q/4$ 或 0 。在算法2的第4行的模转换后，ACC的相位常数项接近 $q/4$ 或 0 。最后，进行

表2 密钥转换的性能分析

方案	标量乘法	标量加法	密钥转换密钥
FINAL ^[14]	$N \cdot L \cdot (n+1)$	$(N \cdot L - 1) \cdot (n+1)$	$N \cdot L$
本文方案	$(N-n) \cdot L \cdot (n+1) + n$	$((N-n) \cdot L - 1) \cdot (n+1) + n$	$(N-n) \cdot L$

算法 3 NewBoot(BRK, ksk_{NTRU}→LWE, c)

输入: 盲旋转密钥BRK, 密钥转换密钥ksk_{NTRU}→LWE, LWE密文 $\mathbf{c} = (b, \mathbf{a}) \in \mathbb{Z}_q^{n+1}$.

输出: LWE密文 $\mathbf{c}' \in \mathbb{Z}_q^{n+1}$.

(1) $\text{ACC} \leftarrow \text{NewBlindRotate}(\text{BRK}, \mathbf{c})$

(2) $\text{ACC} \leftarrow \text{AAC} + \lceil Q/8 \rceil \cdot \sum_{i=0}^{N-1} X^i$

(3) $\text{ACC} \leftarrow \text{ModSwitch}(\text{ACC})$

(4) $\mathbf{c}' \leftarrow \text{NewKeySwitch}_{\text{NTRU} \rightarrow \text{LWE}}(\text{ksk}_{\text{NTRU} \rightarrow \text{LWE}}, \text{ACC})$

第4行中的密钥转换操作时, 本文获得了输出的密文 \mathbf{c}' , 由于密钥转换算法的性质, 它的相位仍然接近 $q/4$ 或 0 . 因此, 新的自举算法NewBoot满足正确性.

3.3.2 具体安全分析

为了证明优化方案的安全性, 本节从原始攻击、对偶攻击以及中间相遇攻击对优化方案的安全性进行了分析. 原始攻击(primal attack)和对偶攻击(dual attack)是目前LWE问题实际安全性分析中最常用、效果最好的两种攻击方法. 其中, 密钥的选取对攻击结果有很大的影响. 因此, 为了应对块二进制密钥带来的挑战, 本文引入了对这两种攻击的优化策略^[14].

(1) 原始攻击: 原始攻击将LWE问题转换成求解格中唯一最短向量问题. 简单来说, 给定LWE样本 $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$, 定义格 $\Lambda = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} : (\mathbf{A} | -\mathbf{I}_m | -\mathbf{b})\mathbf{x} = 0 \pmod{q}\}$, 格 Λ 的维数为 d , 体积为 q^m , $\mathbf{v} = (\mathbf{s}, \mathbf{e}, 1)$ 是格的一个短向量, 其中, $d = m + n + 1$. 通过几何级数假设来模拟BKZ算法, 找到格 Λ 的一个基, 其Gram-Schmidt范数为 $\mathbf{b}_i^* = \delta^{d-2i-1} \cdot \text{Vol}(\Lambda)^{1/d}$, 其中, $\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{1/2(b-1)}$ [26]. 若向量 \mathbf{v} 在由最后 b 个Gram-Schmidt向量张成的子空间上的投影长度小于 \mathbf{b}_{d-b}^* , 则短向量 \mathbf{v} 可以被检测得到. 该投影的范数期望为 $\sigma_s \sqrt{b}$, 原始攻击可以攻击成功当且仅当

$$\sigma_s \sqrt{b} \leq \delta^{2b-d-1} \cdot \text{Vol}(\Lambda)^{1/d} \quad (15)$$

本文中, 密钥的选取为块二进制分布, 可以将之视为具有较低汉明重量的2元分布, 在给定分块尺寸为 l , 分块数量为 k 的情况下, 密钥的汉明重量为 $n/(l+1)$, 密钥的标准差为 $\sigma_s = \sqrt{l}/(l+1)$. FINAL方案中误差 \mathbf{e} 分布的标准差为 $\sigma_e = 4.39 > \sigma_s$, 因此需要重新定义格

$$\Lambda' = \left\{ \mathbf{v} = \begin{pmatrix} cx \\ y \\ \sigma_e z \end{pmatrix} \in \mathbb{R}^{m+n+1} : (\mathbf{A} | -\mathbf{I}_m | -\mathbf{b}) \mathbf{u} = 0 \pmod{q}, \mathbf{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{Z}^{m+n+1} \right\} \quad (16)$$

则下面矩阵的列向量构成格 Λ' 的一组基

$$\mathbf{B} = \begin{pmatrix} c\mathbf{I}_n & 0 & 0 \\ -\mathbf{A} & q\mathbf{I}_m & \mathbf{b} \\ 0 & 0 & \sigma_e \end{pmatrix} \in \mathbb{R}^{(m+n+1) \times (m+n+1)} \quad (17)$$

格的体积 $\det(\Lambda') = \sigma_e \cdot (\sigma_e/\sigma_s)^n \cdot q^m$. 原始攻击可以攻击成功当且仅当

$$\sigma_e \sqrt{b} \leq \delta^{2b-d-1} \cdot \text{Vol}(\Lambda')^{1/d} \quad (18)$$

又因为 $\sigma_s = \sqrt{l}/(l+1)$, 故原始攻击可以攻击成功当且仅当

$$\sigma_e \sqrt{b} \leq \delta^{2b-d-1} \cdot \left(\sigma_e \cdot \frac{l+1}{\sqrt{l}} \right)^n \cdot q^m \quad (19)$$

(2) 对偶攻击: 给定LWE样本 $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$, 令 $\mathbf{s} = \begin{pmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \end{pmatrix}$, $\mathbf{A} = (\mathbf{A}_0 | \mathbf{A}_1)$, 其中, $\mathbf{s}_0 \in \mathbb{Z}_q^{n-k}$, $\mathbf{s}_1 \in \mathbb{Z}_q^k$, $\mathbf{A}_1 \in \mathbb{Z}_q^{m \times k}$, $\mathbf{A}_0 \in \mathbb{Z}_q^{m \times (n-k)}$, 则 $\mathbf{b} + \mathbf{A}_0 \mathbf{s}_0 = -\mathbf{A}_1 \mathbf{s}_1 + \mathbf{e}$.

对偶攻击是在格 $\mathcal{L} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^m \times \mathbb{Z}^n \mid \mathbf{A}^T \mathbf{x} = \mathbf{y} \pmod{q}\}$ 中找到一个短向量 $\mathbf{w} = (\mathbf{x}, \mathbf{y})$. 假设找到了一个长度为 ι 的向量 $\mathbf{v} = (\mathbf{x}, \mathbf{y})$, 并计算 $\mathbf{z} = \mathbf{v}^T \cdot \mathbf{b} = \mathbf{v}^T \mathbf{A} \mathbf{s} + \mathbf{v}^T \mathbf{e} \pmod{q}$, 若 (\mathbf{A}, \mathbf{b}) 符合LWE分布, 则其标准差为 $\iota\sigma$ (否则符合模 q 上在均匀分布). 这两种分布的最大的统计距离为

$$\varepsilon = 4\exp(-2\pi^2\tau^2), \tau = \frac{\iota\sigma}{q} \quad (20)$$

即给定一个长度为 ι 的向量, 构造一个区分优势为 ε 的DLWE的区分器.

向量的长度可以由BKZ算法得出, 即 $\iota = b_0$. 已知对偶格 \mathcal{L} 的维数为 $d = m + n$, 格的体积为 q^n , 可以得到长度 $\iota = \delta^{d-1} q^{n/d}$. 因此, 可以得到的区分优势 $\varepsilon \leq 4\exp(-2\pi^2\tau^2)$, 即

$$\ln(\varepsilon/4) \leq -2\pi^2\tau^2 \quad (21)$$

本文中, 密钥分布视为稀疏的, 向量长度则变为 $\iota' = \delta^d (\sigma_e/\sigma_s)^{m/d} \cdot q^{n/d}$, 最大的统计距离为

$$\varepsilon = 4\exp(-2\pi^2\tau'^2), \tau' = \frac{\sigma_s \cdot \iota'}{q} \quad (22)$$

采用二步格约简算法^[27], 第1步是用 b' 代替 b 来进行BKZ约简, 得到更优的约简基, 第2步是对上一步输出的约简基中的前 b_0 个向量生成的格进行筛选. 该算法在代价计算公式为

$$(d-b)c_{\text{po}}^2 \cdot \text{GT}(b') + c_{\text{po}} \cdot \text{GT}(b_0) \quad (23)$$

其中, b 为BKZ块长, $b' = b - d_{4f}$ 是筛选维数, $c_{\text{po}} = 1/(1 - 2^{-0.292}) = 5.46$, $d_{4f} = \frac{b \cdot \ln(4/3)}{\ln(b/(2\pi e))}$.

(3)中间相遇攻击: May最近的研究^[24]提出了对标准MitM算法在LWE问题上的优化。鉴于攻击效果与密钥空间尺寸紧密相关,且对于块二进制分布的LWE密钥具有相对较少候选密钥的特性,优化的MitM算法十分适用于本文优化方案的安全性分析。

标准MitM算法的一个主要策略是:将秘密向量 \mathbf{s} 分割为 $\mathbf{s} = \mathbf{s}_0 + \mathbf{s}_1$,其中, $\mathbf{s}_0 \in \{0,1\}^n$, $\mathbf{s}_1 \in \{0,1\}^n$,使得 \mathbf{s}_{01} 和 \mathbf{s}_{11} 等于(近似等于) \mathbf{s}_1 的 $1/2$ 。

因为 $b_i + \langle \mathbf{s}, \mathbf{a}_i \rangle$ 很小,所以对于 $\mathbf{s} = \mathbf{s}_0 + \mathbf{s}_1$, $b_i + \langle \mathbf{s}_0, \mathbf{a}_i \rangle$ 近似等于 $-\langle \mathbf{s}_1, \mathbf{a}_i \rangle$ 。该算法因此建立了两个集合

$$\mathcal{R}_0 = \{b_i + \langle \mathbf{x}_0, \mathbf{a}_i \rangle \mid \mathbf{x}_0 \in \mathbb{B}^n \wedge \mathbf{x}_{01} = \mathbf{s}_1/2\} \quad (24)$$

$$\mathcal{R}_1 = \{-\langle \mathbf{x}_1, \mathbf{a}_i \rangle \mid \mathbf{x}_1 \in \mathbb{B}^n \wedge \mathbf{x}_{11} = \mathbf{s}_1/2\} \quad (25)$$

中间相遇攻击期望两个集合的碰撞发生在正确的对 $(\mathbf{s}_0, \mathbf{s}_1)$ 上。即碰撞可以将秘密 \mathbf{s} 恢复为 $\mathbf{s}_0 + \mathbf{s}_1$ 。基于此思想,标准MitM可以在 $\mathcal{S}^{0.5}$ 时间内恢复秘密向量 \mathbf{s} ,其中, \mathcal{S} 是密钥空间 \mathcal{S} 的基数。

May^[24]优化标准MitM的一个关键思想是对目标向量 \mathbf{s}_0 和 \mathbf{s}_1 进行归纳分割。准确地说,这两个向量被分成 $\mathbf{s}_i = \mathbf{s}_{i,0} + \mathbf{s}_{i,1}$,其中, $\mathbf{s}_{i,j}$ 是一个汉明权重更小的向量,也可以看作是基于树搜索的泛化,密钥被反复分解,直到它们具有足够小的汉明权重,并且可以很容易地找到。

优化的MitM算法,可以在时间(渐近) $\mathcal{S}^{0.25}$ 得到秘密向量 \mathbf{s} ,但是这个渐近结果忽略了猜测部分。根据文献^[24],当该算法应用于NTRU, BLISS和GLP等具体实例时,包括猜测部分的总成本在 $[\mathcal{S}^{0.28}, \mathcal{S}^{0.3}]$ 范围内。因此,本文估计MitM算法的时间复杂度为 $\mathcal{S}^{0.28}$ 。将块二进制分布 $\mathbf{B}_{l,k}$ 的密钥空间 \mathcal{S} 的尺寸 $(l+1)^k$ 代入复杂度中,即可以在 $O(2^{0.28k \cdot \log_2(l+1)})$ 时间内恢复密钥。

表3给出了不同参数下不同攻击方案的安全级数,由该表可知,密钥选取的块尺寸越大,优化FINAL方案的安全级数越低。当选取块尺寸 $l=2$ 时,方案的安全级数与原FINAL方案安全级数相当。

4 FINAL优化方案性能分析

本节给出具体参数下FINAL优化方案的性能。根据3.6节的安全性分析,不妨设LWE密钥选取的块二进制密钥分布为 $\mathbf{B}_{2,305}$,即 $l=2, k=305$,原FINAL方案参数设置见表4。

根据3.3节中分析,原方案的盲旋转所需的FFT次数为 $(\ell+1) \cdot n$,标量乘法次数为 $\ell \cdot N \cdot n$ 。在表4的参数设置下,FFT次数为 $(\ell_1+1) \cdot n_1 + (\ell_2+1) \cdot n_2 = 3\,940$,标量乘法次数为 $(\ell_1 n_1 + \ell_2 n_2) \cdot N = 3\,409\,920$ 。优化方案需要进行的FFT次数为 $(\ell_1+1) \cdot (n_1/l) + (\ell_2+1) \cdot (n_1/l) = 1\,970$,标量乘法次数不变。具体性能对比见表5。

结合3.5节给出的密钥转换性能分析,在本节参数下的密钥转换的具体性能比较见表6。

与原FINAL方案相比,每次自举所需的外积及快速傅里叶变换(FFT)次数优化50%,密钥转换密钥的尺寸以及密钥转换的运算优化约60%。

5 结论

本文通过对FINAL方案进行分析以及优化,提出了计算开销以及存储开销都优于FINAL方案的优化方案,并对该优化方案进行了安全性分析以及性能分析。当参数选择为 $l=2, k=305, q=92\,683, N=1\,024, n=610$ 时,通过对FINAL方案的LWE密钥采用块二进制分布采样,即密钥相邻两个分量组成一个块结构,将密钥分为305个块。在此基础上,对FINAL方案自举的盲旋转累加器进行压缩,将其中需要的外积和快速傅里叶变换的数量分别由630和3\,940减少到305和1\,970,即都优

表3 安全性分析

n	N	l	k	q	MitM	Primal	Dual
610	1 024	1	610	92 683	170	125	134
610	1 024	2	305	92 683	135	124	131
610	1 024	3	204	92 683	113	123	129
610	1 024	4	153	92 683	98	122	128
610	1 024	5	122	92 683	88	121	126

表4 原FINAL方案的参数设置

方案	n	q	N	Q	(B_1, n_1)	(B_2, n_2)	B_{ksk}	ℓ_1	ℓ_2
FINAL ^[14]	610	92 683	1 024	912 829	(8,140)	(16,470)	3	7	5

表 5 盲旋转算法的性能比较

方案	FFT	标量乘法
FINAL ^[14]	3 940(×1)	3 409 920(×1)
本文方案	1 970(×0.5)	3 409 920(×1)

表 6 密钥转换算法的性能比较

方案	标量乘法	标量加法	密钥转换密钥
FINAL ^[14]	6 882 304	6 881 693	11 264
本文方案	2 783 104	2 782 493	4 554

化了50%。二是通过密钥复用技术,利用LWE密钥 s 来生成NGS的密钥 f ,进而提出新的密钥转换方案,该密钥转换方案与原方案相比,密钥规模由11 264减少到4 554,标量乘法以及标量加法的运算次数由 13.8×10^6 减少到 5.6×10^6 ,即都优化约60%。

参 考 文 献

- [1] GENTRY C. Fully homomorphic encryption using ideal lattices[C]. The Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, USA, 2009: 169–178. doi: [10.1145/1536414.1536440](https://doi.org/10.1145/1536414.1536440).
- [2] BRAKERSKI Z, GENTRY C, and VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[J]. *ACM Transactions on Computation Theory (TOCT)*, 2014, 6(3): 13. doi: [10.1145/2633600](https://doi.org/10.1145/2633600).
- [3] BRAKERSKI Z. Fully homomorphic encryption without modulus switching from classical GapSVP[C]. The 32nd Annual Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 2012: 868–886. doi: [10.1007/978-3-642-32009-5_50](https://doi.org/10.1007/978-3-642-32009-5_50).
- [4] FAN Junfeng and VERCAUTEREN F. Somewhat practical fully homomorphic encryption[EB/OL]. <https://eprint.iacr.org/2012/144>, 2012.
- [5] GENTRY C, SAHAI A, and WATERS B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]. The 33rd Annual Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 2013: 75–92. doi: [10.1007/978-3-642-40041-4_5](https://doi.org/10.1007/978-3-642-40041-4_5).
- [6] CHEON J H, KIM A, KIM M, *et al.* Homomorphic encryption for arithmetic of approximate numbers[C]. The 23rd International Conference on the Theory and Applications of Cryptology and Information Security Advances in Cryptology, Hong Kong, China, 2017: 409–437. doi: [10.1007/978-3-319-70694-8_15](https://doi.org/10.1007/978-3-319-70694-8_15).
- [7] DUCAS L and MICCIANCIO D. FHEW: Bootstrapping homomorphic encryption in less than a second[C]. The 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology, Sofia, Bulgaria, 2015: 617–640. doi: [10.1007/978-3-662-46800-5_24](https://doi.org/10.1007/978-3-662-46800-5_24).
- [8] CHILLOTTI I, GAMA N, GEORGIEVA M, *et al.* TFHE: Fast fully homomorphic encryption over the torus[J]. *Journal of Cryptology*, 2020, 33(1): 34–91. doi: [10.1007/s00145-019-09319-x](https://doi.org/10.1007/s00145-019-09319-x).
- [9] CARPOV S, IZABACHÈNE M, and MOLLIMARD V. New techniques for multi-value input homomorphic evaluation and applications[C]. The Cryptographers' Track at the RSA Conference 2019 Topics in Cryptology, San Francisco, USA, 2019: 106–126. doi: [10.1007/978-3-030-12612-4_6](https://doi.org/10.1007/978-3-030-12612-4_6).
- [10] CHILLOTTI I, LIGIER D, ORFILA J B, *et al.* Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE[C]. The 27th International Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology, Singapore, Singapore, 2021: 670–699. doi: [10.1007/978-3-030-92078-4_23](https://doi.org/10.1007/978-3-030-92078-4_23).
- [11] GUIMARÃES A, BORIN E, and ARANHA D F. Revisiting the functional bootstrap in TFHE[J]. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021, 2021(2): 229–253. doi: [10.46586/tches.v2021.i2.229-253](https://doi.org/10.46586/tches.v2021.i2.229-253).
- [12] CHEN Hao, CHILLOTTI I, and SONG Y. Multi-key homomorphic encryption from TFHE[C]. The 25th International Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology, Kobe, Japan, 2019: 446–472. doi: [10.1007/978-3-030-34621-8_16](https://doi.org/10.1007/978-3-030-34621-8_16).
- [13] KWAK H, MIN S, and SONG Y. Towards practical multi-key TFHE: Parallelizable, key-compatible, quasi-linear complexity[C]. The 27th IACR International Conference on Practice and Theory of Public-Key Cryptography Public-Key Cryptography, Sydney, Australia, 2024: 354–385. doi: [10.1007/978-3-031-57728-4_12](https://doi.org/10.1007/978-3-031-57728-4_12).
- [14] BONTE C, ILIASHENKO I, PARK J, *et al.* FINAL: Faster FHE instantiated with NTRU and LWE[C]. The 28th International Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology, Taipei, China, 2022: 188–215. doi: [10.1007/978-3-031-22966-4_7](https://doi.org/10.1007/978-3-031-22966-4_7).
- [15] KLUCZNIAK K. NTRU-v-um: Secure fully homomorphic encryption from NTRU with small modulus[C]. The 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, USA, 2022: 1783–1797. doi: [10.1145/3548606.3560700](https://doi.org/10.1145/3548606.3560700).
- [16] LEE Y, MICCIANCIO D, KIM A, *et al.* Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption[C]. The 42nd Annual

- International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology, Lyon, France, 2023: 227–256. doi: [10.1007/978-3-031-30620-4_8](https://doi.org/10.1007/978-3-031-30620-4_8).
- [17] LEE C, MIN S, SEO J, *et al.* Faster TFHE bootstrapping with block binary keys[C]. The 2023 ACM Asia Conference on Computer and Communications Security, Melbourne, Australia, 2023: 2–13. doi: [10.1145/3579856.3595804](https://doi.org/10.1145/3579856.3595804).
- [18] XIANG Binwu, ZHANG Jiang, DENG Yi, *et al.* Fast blind rotation for bootstrapping FHEs[C]. The 43rd Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 2023: 3–36. doi: [10.1007/978-3-031-38551-3_1](https://doi.org/10.1007/978-3-031-38551-3_1).
- [19] MA Shihe, HUANG Tairong, WANG Anyu, *et al.* Accelerating BGV bootstrapping for large p using null polynomials over \mathbb{Z}_{p^e} [C]. The 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology, Zurich, Switzerland, 2024: 403–432. doi: [10.1007/978-3-031-58723-8_14](https://doi.org/10.1007/978-3-031-58723-8_14).
- [20] WANG Ruida, WEN Yundi, LI Zhihao, *et al.* Circuit bootstrapping: Faster and smaller[C]. The 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology, Zurich, Switzerland, 2024: 342–372. doi: [10.1007/978-3-031-58723-8_12](https://doi.org/10.1007/978-3-031-58723-8_12).
- [21] PEIKERT C. A decade of lattice cryptography[J]. *Foundations and Trends® in Theoretical Computer Science*, 2016, 10(4): 283–424. doi: [10.1561/04000000074](https://doi.org/10.1561/04000000074).
- [22] GOLDWASSER S, KALAI Y, PEIKERT C, *et al.* Robustness of the learning with errors assumption[C]. Proceedings of the Innovations in Computer Science–ICS 2010, Beijing, China, 2010: 230–240.
- [23] ALBRECHT M R. On dual lattice attacks against small-secret LWE and parameter choices in HELib and SEAL[C]. The 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology, Paris, France, 2017: 103–129. doi: [10.1007/978-3-319-56614-6_4](https://doi.org/10.1007/978-3-319-56614-6_4).
- [24] MAY A. How to meet ternary LWE keys[C]. The 41st Annual International Cryptology Conference on Advances in Cryptology, 2021: 701–731. doi: [10.1007/978-3-030-84245-1_24](https://doi.org/10.1007/978-3-030-84245-1_24).
- [25] ALBRECHT M R, PLAYER R, and SCOTT S. On the concrete hardness of Learning with Errors[J]. *Journal of Mathematical Cryptology*, 2015, 9(3): 169–203. doi: [10.1515/jmc-2015-0016](https://doi.org/10.1515/jmc-2015-0016).
- [26] CHEN Yuanmi and NGUYEN P Q. BKZ 2.0: Better lattice security estimates[C]. The 17th International Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology, Seoul, South Korea, 2011: 1–20. doi: [10.1007/978-3-642-25385-0_1](https://doi.org/10.1007/978-3-642-25385-0_1).
- [27] GUO Qian and JOHANSSON T. Faster dual lattice attacks for solving LWE with applications to CRYSTALS[C]. The 27th International Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology, Singapore, Singapore, 2021: 33–62. doi: [10.1007/978-3-030-92068-5_2](https://doi.org/10.1007/978-3-030-92068-5_2).
- 赵秀凤: 女, 博士, 副教授, 研究方向为全同态密码、格密码、密码协议等。
- 吴蒙: 男, 硕士生, 研究方向为全同态加密。
- 宋巍涛: 男, 博士, 副教授, 研究方向为全同态密码、云计算等。

责任编辑: 余蓉

Bootstrapping Optimization Techniques for the FINAL Fully Homomorphic Encryption Scheme

ZHAO Xiufeng WU Meng SONG Weitao

(College of Cryptography Engineering Information Engineering University, Zhengzhou 450001, China)

Abstract:

Objective Bootstrapping is a fundamental process in Fully Homomorphic Encryption (FHE) that directly affects its practical efficiency. The FINAL scheme, presented at ASIACRYPT 2022, achieves a 28% improvement in bootstrapping speed compared with TFHE, demonstrating high suitability for homomorphic Boolean operations. Nevertheless, further improvements are required to reduce its computational overhead and storage demands. This study aims to optimize the bootstrapping phase of FINAL by lowering its computational complexity and key size while preserving the original security level.

Methods This study proposes two key optimizations. Accumulator compression for blind rotation: A blockwise binary distribution is incorporated into the Learning With Errors (LWE) key generation process. By organizing

the key into blocks, each requiring only a single external product, the number of external product operations during blind rotation is reduced. Key reuse strategy for key switching: The LWE key is partially reused during the generation of the Number-theoretic Gadget Switching (NGS) key. The reused portion is excluded from the key switching key, thereby reducing both the key size and the number of associated operations.

Results and Discussions Under equivalent security assumptions, the optimized FINAL scheme yields substantial efficiency gains. For blind rotation, the number of external product operations is reduced by 50% (from 610 to 305), and the number of Fast Fourier Transform (FFT) operations is halved (from 3,940 to 1,970) (Table 5). For key switching, the key size is reduced by 60% (from 11,264 to 4,554), and the computational complexity decreases from 13.8×10^6 to 5.6×10^6 scalar operations (Table 6).

Conclusions The proposed optimizations substantially improve the efficiency of the FINAL scheme's bootstrapping phase. Blind rotation benefits from structured key partitioning, reducing the number of core operations by half. Key switching achieves comparable reductions in both storage requirements and computational cost through partial key reuse. These enhancements improve the practicality of FHE for real-world applications that demand efficient evaluation of Boolean circuits. Future directions include hardware acceleration and adaptive parameter tuning.

Key words: Fully Homomorphic Encryption (FHE); FINAL; Bootstrapping; Blind rotation; Key switching