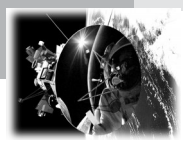

专栏

面向智算网络的SCLOUD+后量子密钥封装适配与BW格优化研究

潘洁[1]，郗卓宁[2]，赵占军[1]，卜忠贵[1]，侯慧芳[1]，叶兰[2]，陈曦[1]，薛曌[1]
（1. 中国移动通信集团设计院有限公司，北京 100080；
2. 中国移动通信集团有限公司，北京 100032）

摘　要：现有主流后量子密钥封装机制，如基于模块格的密钥封装机制（module-lattice-based key encapsulation mechanism，ML-KEM），依赖于结构化格的模上容错学习（module learning with errors，Module-LWE）问题，其代数结构可能导致归约漏洞。聚焦于非结构化LWE框架下的SCLOUD+方案，其基于巴恩斯-沃尔（Barnes-Wall，BW）格的递归构造特性，通过高维格编码增益实现公钥与密文尺寸的显著压缩。基于该方案提出一种面向BW格的维度特化全展开递归消除技术，通过编译期常量优化、分层硬编码策略及单指令多数据（single-instruction multiple-data，SIMD）友好内存布局，将BW格在128维场景下的解码时钟周期从147 798降至30 107，为SCLOUD+后量子密钥封装机制提供了核心支撑。此研究为智算网络提供了一种兼顾高效性与抗量子安全性的轻量级密钥封装机制（key encapsulation mechanism，KEM）范式，为分布式联邦学习等低时延应用场景奠定了关键技术基础。

关键词：智算网络；巴恩斯-沃尔格；有界距离解码；后量子密码；密钥封装
中图分类号：TN918.4
文献标志码：A
doi: 10.11959/j.issn.1000-0801.2025169

Research on SCLOUD+ post-quantum key encapsulation adaptation and BW lattice optimization for intelligent computing networks

PAN Jie[1], XI Zhuoning[2], ZHAO Zhanjun[1], BU Zhonggui[1],
HOU Huifang[1], YE Lan[2], CHEN Xi[1], XUE Zhao[1]
1. China Mobile Group Design Institute Co., Ltd., Beijing 100080, China
2. China Mobile Communications Group Co., Ltd., Beijing 100032, China

Abstract: The existing mainstream post-quantum key encapsulation mechanisms, such as the module-lattice-based key encapsulation mechanism (ML-KEM), rely on the module learning with errors (Module-LWE) problem associated with structured lattices. The algebraic structure of these mechanisms may lead to reduction vulnerabilities. The SCLOUD+ scheme, which was focused on within the unstructured LWE framework, achieved significant compres-

收稿日期：2025-03-21；修回日期：2025-07-14

sion of public key and ciphertext sizes through high-dimensional lattice coding gain, based on the recursive construction properties of the Barnes-Wall (BW) lattice. Moreover, a dimension-specific full-unfolding recursive elimination technique for BW lattices was proposed. Through compile-time constant optimization, hierarchical hard-coding strategies, and single-instruction, multiple-data (SIMD) friendly memory layouts, the decoding clock cycles of the BW lattice in a 128-dimensional scenario were reduced from 147 798 to 30 107, providing core support for the SCLOUD+ post-quantum key encapsulation mechanism. This research provided a lightweight key encapsulation mechanism (KEM) paradigm that balances efficiency and quantum-resistant security for intelligent computing networks, laying a crucial technical foundation for low-latency scenarios such as distributed federated learning.

Key words: intelligent computing network, Barnes-Wall lattice, bounded distance decoding, post-quantum cryptography, key encapsulation

0　引言

当前智算网络仍处于起步发展阶段，需要满足大模型带来的超大规模算力集群、超高网络吞吐、超低网络时延及高可靠性等业务需求，其网络协议包括路由协议、流量调度机制、拥塞管理技术和远程直接存储器访问（remote direct memory access，RDMA）传输优化等关键技术[1-2]。RDMA是一种为解决网络传输中服务器端数据处理时延而产生的技术，其允许计算机直接访问另一台计算机的内存，绕过操作系统内核和中央处理器（central processing unit，CPU），从而有效降低协议栈时延，提升CPU的效率。需要强调的是，在默认配置下，RDMA存在以下潜在风险：未加密传输，数据在网络上明文传输，易被窃听；缺乏身份验证，攻击者可伪装成合法节点发起中间人攻击；内存访问越权，恶意节点可能通过RDMA操作非法读写目标主机内存；网络隔离不足，共享RDMA网络的多租户环境存在数据泄露风险。为了提升RDMA的安全性，通常与传输层安全（transport layer security，TLS）协议或互联网络层安全（Internet protocol security，IPSec）协议相结合，如RFC 5042中提出的数据包传输层安全性（datagram transport layer security，DTLS）协议[3-6]。

然而，传统的TLS和IPSec面临量子计算带来的系统性安全威胁，同时量子算法（如Shor算法）可高效破解传统公钥密码体系，如RSA（Rivest-Shamir-Adleman）加密算法、椭圆曲线加密（elliptic curve cryptography，ECC）算法，导致智算网络中的身份认证、密钥协商及数据传输面临严峻风险[7-9]。在智算网络中，密钥协商算法需要频繁执行于边缘节点、云服务器及终端设备，其效率直接影响全局任务吞吐量与资源利用率，如在联邦学习时跨节点模型聚合需要多次密钥协商与数据加密[10]。因此，部署抗量子密码算法成为保障智算网络长期安全的必然选择。中国移动通信集团研究院在其《新型智算中心以太网物理层安全（PHYSec）架构白皮书（2024年）》中指出，智算网络对任务处理具有低时延与高吞吐的需求，要求后量子密码算法不仅要具备理论安全性，还需要满足工程化部署的效率要求[11]。

后量子密码算法是一类能够抵抗量子计算机攻击的密码算法。随着量子计算技术的发展，传统的基于数学难题（如整数分解、离散对数等）的密码算法面临被量子算法快速破解的风险[12]。后量子密码算法基于不同的数学结构，如格、编码理论、多变量多项式等，旨在提供在量子计算环境下的安全保障[13]。常见的后量子密码算法包括基于格的密码算法（如本文研究的基于容错学习（learning with errors，LWE）框架的算法）、基于编码的密码算法、基于哈希的密码算法，以

及基于多变量公钥的密码算法等[14]。后量子密码的重要性在于，它为未来量子计算时代的信息安全提供了关键的保护手段，对于保障智算网络等重要信息基础设施的安全运行具有不可或缺的作用。

在密码学中，基于结构化格的方案，如环上容错学习（ring learning with errors，Ring-LWE）问题、模上容错学习（module learning with errors，Module-LWE）问题，通过引入代数结构显著提升了效率，如美国国家标准与技术研究院（National Institute of Standards and Technology，NIST）推出的标准算法——基于模块格的密钥封装机制（module-lattice-based key encapsulation mechanism，ML-KEM），但其安全性存在潜在风险[15]。结构化格的困难性假设如分圆数域理想中的最短向量问题（shortest vector problem in ideals of cyclotomic number fields，Ideal-SVP）和模块最短向量问题（module shortest vector problem，Module-SVP），与经典格的最短向量问题（shortest vector problem，SVP）和最短独立向量问题（shortest independent vectors problem，SIVP）的安全性基础不同，且其抗量子能力尚存争议。已有研究表明，某些代数结构可能被量子算法更高效地破解，如文献[16]证明了特定分圆域上的近似Ideal-SVP可在量子多项式时间内求解。尽管这类攻击尚未直接扩展到Module-LWE等更广泛的结构化问题，但代数结构的安全性假设仍缺乏与经典格SVP的严格等价性，导致其长期抗量子性存疑。相比之下，非结构化LWE方案（如FrodoKEM）的安全性直接归约于经典格SVP的平均情况困难性，虽效率较低，但被视为更保守的安全选择[17]。

为规避结构化格的安全隐患，非结构化LWE框架因其保守的安全假设成为高安全性场景的优选，但其效率与密钥尺寸远逊于结构化LWE方案，制约了其实际应用。引入纠错编码压缩通信开销是一个关键方向，如BCH（Bose-Chaudhuri-Hocquenghem）码或低密度奇偶校验（low-density parity-check，LDPC）码，尽管格编码在LWE方案中应用广泛，但此前方案多采用4～16维格码，如D_4、E_8[18]。高维格码（如24维Leech格）虽能提供更高信噪比和纠错能力，但因缺乏高效的标签化与解标签化流程（即消息与格点坐标的高效映射），难以实用化，如文献[18]的Leech格方案因标签化技术缺失而被弃用。SCLOUD+作为一种基于非结构化LWE的KEM方案，通过引入巴恩斯-沃尔（Barnes-Wall，BW）格的高维编码增益，实现了公钥与密文尺寸的显著压缩，较FrodoKEM减少了30%～40%，同时保持了与传统LWE问题等效的安全性[19-21]。BW格的递归构造特性，如最小距离随维度指数增长、层次化格点分布，使其天然适配密钥封装中的纠错编码需求。然而，SCLOUD+依赖递归式有界距离解码（bounded distance decoding，BDD）算法，其递归调用导致的函数栈开销、动态内存分配低效及分支预测失败等问题，使得单次解码时延高达147 798时钟周期，难以满足智算网络微秒级响应的实时需求。因此，优化BW格解码效率，既是提升SCLOUD+实用性的关键，也是智算网络实现“高效-安全”平衡的核心突破口。针对上述问题，本文提出一种全展开递归消除技术，通过以下创新实现性能突破。

（1）编译期常量优化：将旋转因子$\varphi=1+i$及其逆$\varphi^{-1}=0.5(1-i)$预计算为静态常量，消除运行时的复数运算开销。

（2）分层硬编码策略：针对BW格典型维度（32、64、128）生成零函数调用的扁平化代码路径，彻底消除递归栈操作。

（3）单指令多数据（single-instruction multiple-data，SIMD）友好内存布局：通过连续对齐的内存分配加配合C语言的O3编译选项提升吞吐量。

实验结果表明，优化后算法在128维解码任务中实现4.5倍以上加速，显著优于现有方案。

本研究的贡献包括以下方面。

- 将全展开优化策略应用于BW格解码，为后量子密码硬件加速提供新范式。
- 提出静态内存分配与向量化计算协同优化方法，在一定程度上缓解高维场景实时性限制。
- 通过SCLOUD+验证方案兼容性，为后量子密码算法设计提供关键技术支撑，为分布式智算网络计算场景提供抗量子攻击保障。

1　关键技术

1.1　智算网络通信架构

传统基于TCP/IP的通信架构在数据传输过程中普遍面临协议处理开销大、时延敏感场景适应性不足等问题，具体表现为协议栈处理产生的多层级时延、用户态与内核态间反复的数据复制操作，以及复杂的流量控制机制。为应对上述性能瓶颈，RDMA技术通过架构创新实现了通信效率的突破，采用操作系统内核旁路机制消除上下文切换损耗，结合内存零复制技术，建立应用于网络设备的直接交互通道，显著缩短了数据传输路径，并释放了CPU算力。值得注意的是，RDMA技术的高效运行依赖于底层网络的无损传输保障，当前主流实现方案可分为以下3种技术路线。

（1）无限带宽（InfiniBand，IB）网络：作为最早商用的方案，采用专用网络协议栈实现硬件级流控，具备优异的传输性能与稳定性，但因技术生态封闭性较强、部署成本较高，主要应用于特定高性能计算场景。

（2）融合以太网的远程直接存储器访问（RDMA over converged Ethernet，RoCE）：通过以太网协议扩展支持RDMA功能，在成本与性能间取得平衡，但需要配合基于优先级的流量控制（priority-based flow control，PFC）或显式拥塞通知（explicit congestion notification，ECN）等拥塞控制算法进行网络参数优化配置，以维持无损特性。

（3）互联网广域RDMA协议（Internet wide area RDMA protocol，iWARP）：虽然兼容传统IP网络基础设施，但协议转换带来的额外开销导致性能指标落后于前两者，市场应用规模逐渐萎缩[22]。

1.2　后量子密码算法中BW格的数学构造与性质

后量子密码算法中BW格的数学构造基于高斯整数环 $\mathbb{Z}[i]$ 的层次化递归扩展，其核心定义与数学性质如下。

在格理论研究范畴中，BW格是一类基于复整数空间构建的特殊格结构，其定义域限定于维度 $N=2^n(n\geqslant 1,n\in N)$。在数学表达体系中，BW格存在两种等价表示形式：基于维度参数的 BW_N 与基于指数参数的 $\mathbf{BW}^n$。本节在涉及基础定义与理论推导时，为保持符号体系的简洁性与一致性，统一采用 $\mathbf{BW}^n$ 作为标准表述，而在后续算法、实验与应用场景讨论中，考虑维度数值的直观可读性，将切换为 $\mathbf{BW}_N$ 的表示形式。

BW格的构造机理源于高斯整数环 $\mathbb{Z}[i]$ 的层次化递归扩展过程。该构造方法以 $\mathbb{Z}[i]$ 为基础代数结构，通过迭代应用特定的矩阵变换与嵌入规则，在复欧几里得空间 $\mathbb{C}^N$ 中逐层构建格点集合。其核心定义深度融合了代数数论与几何格论的理论特性，不仅满足格结构的基本公理体系，还具备独特的代数对称性与几何规整性，这些数学性质使其在密码学、编码理论等领域展现出重要的应用价值。

定义1　Kronecker积构造

BW格的生成矩阵可通过递推Kronecker积来构造，表示为：

$$\mathbf{BW}^n=\begin{bmatrix}1 & 1\\ 0 & \varphi\end{bmatrix}^{\otimes n},\varphi=1+i \tag{1}$$

其中，$\otimes$表示Kronecker积初始条件为$\mathbf{BW}^0=[1]$，每个递归步骤将维度扩展为原来的两倍。例如，$\mathbf{BW}^1=\begin{bmatrix}1 & 1\\ 0 & 1+i\end{bmatrix}$对应四维实空间中的$D_4$格。

定义2 递归矢量构造

BW格还可通过矢量对的递归组合来定义，表示为：

$$\mathbf{BW}^n=\left[\boldsymbol{u},\boldsymbol{u}+\varphi\boldsymbol{v}\right]\Big|\boldsymbol{u},\boldsymbol{v}\in\mathbf{BW}^{n-1} \tag{2}$$

该构造揭示了BW格的层次化结构，即每个高维格点由两个低维矢量经线性变换组合而成。

BW格的关键性质如下。

（1）最小平方距离：$d^2{}_{\min}\left(\mathbf{BW}^n\right)=2^{n-1}$，最小距离：$d_{\min}\left(\mathbf{BW}^n\right)=\sqrt{\frac{2^n}{2}}$。

（2）唯一解码半径：$\mathrm{dist}\left(\boldsymbol{t},\mathbf{BW}^n\right)<\frac{d_{\min}}{2}=\sqrt{\frac{2^n}{8}}$，其中$\boldsymbol{t}$表示$\mathbf{BW}^n$格点附近的任意点。

（3）体积：$\mathrm{vol}\left(\mathbf{BW}^n\right)=2^{(n-1)\cdot 2^{n-2}}$。

（4）编码增益：$\gamma_c(\mathbf{BW}^n)=\sqrt{N}$，随维度指数增长。

（5）代数结构：BW格是$\mathbb{Z}[i]$模，其旋转对称性支持高效解码。

1.3 BW格的解码算法

目前已知的BW格的解码算法主要有以下三大类。

（1）最大似然解码（maximum likelihood decoding，MLD）：MLD旨在找到与目标向量$\boldsymbol{t}\in\mathbb{C}^N$欧氏距离最近的格点，其理论最优性以指数级复杂度为代价。文献[23-24]提出基于网格表示的MLD算法，通过动态规划遍历所有可能路径，复杂度为指数级别，仅适用于$N\leqslant 16$的低维场景。例如，解码$\mathbf{BW}^4$（16维）需要调用2^8+2^4次$\mathbf{BW}^2$解码器。

（2）BDD：文献[25]提出首个多项式时间BDD算法，通过递归分解目标向量生成候选解，时间复杂度为$O\left(N\log^2 N\right)$。文献[24]进一步设计双路径递归解码，提升容错能力，但分支预测失败率在$N=128$时仍达12.7%，难以满足实时需求。

（3）列表解码（list decoding，LD）：文献[26]提出列表解码框架，通过多级剪枝来控制候选数量，但需要权衡解码半径与计算效率。

1.4 工程化实现的挑战与优化

BW格的实用化面临两大核心挑战：递归开销与内存效率。现有优化技术包括以下3种。

（1）递归消除：传统递归实现（如SCLOUD+开源代码）在低维场景下频繁的函数调用导致指令缓存抖动。实验表明，$N=128$时函数调用开销占比达40%。

（2）内存访问优化：动态内存分配（如malloc）引发堆碎片化，阻碍SIMD指令的连续内存访问。Intel Optane持久内存技术通过大容量存储缓解压力，但引入了额外时延。

（3）硬件加速：英伟达（NVIDIA）的CUDA-LWE方案利用图形处理单元（graphics processing unit，GPU）并行化提升吞吐量，但受限于显存带宽，难以适配微秒级时延场景。

2 算法设计

2.1 问题描述

人工智能与算力需求的指数级增长，推动智算网络成为下一代智能计算的核心底座，其核心使命在于通过“网络即计算”的深度融合，实现算力资源的泛在接入与智能调度。然而，当前技术体系仍面临多维度挑战：一方面，算力资源的异构性（如CPU、GPU、NPU等算力单元性能差异）与网络传输的动态性导致全局资源调度效率受限；另一方面，现有智算网络的通信安全模块

仍以RSA或ECC算法为核心，需要进一步适配后量子密码算法以满足长期安全需求，而智算网络对低时延、高吞吐的严苛要求，进一步压缩了后量子密码算法的工程化部署空间。例如，基于结构化格的后量子方案虽能实现较高效率，但其潜在代数结构漏洞可能引发长期安全隐患。这种“效率−安全”的深度博弈，亟须通过密码学理论与网络架构的协同创新实现破局，而基于非结构化格的高效解码技术，或将成为平衡抗量子安全性与智算网络实时性需求的关键支点。

SCLOUD+利用BW_{32}格优化LWE密钥封装机制，显著降低了公钥与密文尺寸。然而，BW格的实用化面临核心挑战：传统递归式BDD算法在高维场景（如$N=32$、64、128）下存在严重的性能瓶颈。研究表明，递归调用导致的指令缓存抖动、动态内存分配引发的堆碎片化以及分支预测失效等问题，使得解码效率较低。本节详述基于全展开递归消除的BW格解码优化方法，并整合SCLOUD+的标签化（labeling）与解标签化（delabeling）算法，阐明其在KEM中的具体应用。此外，本文方案通过对比SCLOUD+开源代码，验证全展开策略的优化效果。

2.2　SCLOUD+算法

SCLOUD+算法优于FrodoKEM，其核心依赖于递归式BDD算法实现BW_{32}格的解码。SCLOUD+递归式BDD算法流程如算法1所示，SCLOUD+标签化算法如算法2所示。

算法1　SCLOUD+递归式BDD算法

输入　待解码的目标向量$\boldsymbol{t}\in\mathbb{C}^{N/2}\left(N=2^n\right)$

输出　格点向量$\boldsymbol{y}\in\mathrm{BW}_N$，满足$\left\|\boldsymbol{t}-\boldsymbol{y}\right\|\leqslant\sqrt{N/8}$

（1）若$N=2$

对每个分量进行高斯整数四舍五入：$\boldsymbol{y}_i=\underset{z\in\mathbb{Z}[i]}{\operatorname{argmin}}\|t_i-z\|\,(i=0,1)$

返回$\boldsymbol{y}=\left(\boldsymbol{y}_0,\boldsymbol{y}_1\right)$

（2）否则

分割$\boldsymbol{t}$为两个子向量$\boldsymbol{t}_0$、$\boldsymbol{t}_1$

递归解码：$\boldsymbol{y}_0=\mathrm{BDD}\left(\boldsymbol{t}_0\right),\boldsymbol{y}_1=\mathrm{BDD}\left(\boldsymbol{t}_1\right)$

计算误差向量：$\boldsymbol{z}_0^{\mathrm{in}}=\varphi^{-1}\left(\boldsymbol{t}_1-\boldsymbol{y}_0\right),\boldsymbol{z}_1^{\mathrm{in}}=\varphi^{-1}\left(\boldsymbol{t}_0-\boldsymbol{y}_1\right)$

递归解码误差向量：$\boldsymbol{z}_0=\mathrm{BDD}\left(\boldsymbol{z}_0^{\mathrm{in}}\right),\boldsymbol{z}_1=\mathrm{BDD}\left(\boldsymbol{z}_1^{\mathrm{in}}\right)$

生成候选解：$\boldsymbol{x}_0=\left[\boldsymbol{y}_0,\boldsymbol{y}_0+\varphi\boldsymbol{z}_0\right],\boldsymbol{x}_1=\left[\boldsymbol{y}_1+\varphi\boldsymbol{z}_1,\boldsymbol{y}_1\right]$

返回满足最小范数的候选点：$\boldsymbol{y}=\underset{\boldsymbol{x}\in\{\boldsymbol{x}_0,\boldsymbol{x}_1\}}{\operatorname{argmin}}\|\boldsymbol{t}-\boldsymbol{x}\|$

算法2　SCLOUD+标签化算法（消息→格点）

输入　原始消息向量$\boldsymbol{m}\in\{0,1\}^{64}$

输出　格点向量$\boldsymbol{x}\in\mathrm{BW}_{32}$

（1）将消息分为16个子块$\boldsymbol{u}_j\in\{0,1\}^{6-W_{\mathrm{H}}(j)}$，其中$W_{\mathrm{H}}(j)$表示第$j$个子块的汉明重量（Hamming weight）。

（2）将每个子块通过函数f_l映射为高斯整数$v_j=a_j+b_ji$，满足$0\leqslant a_j<2^{\lceil l/2\rceil},0\leqslant b_j<2^{\lfloor l/2\rfloor}$，其中$l$表示该子块的比特长度，$\lceil l/2\rceil$表示将$l/2$向上取整，$\lfloor l/2\rfloor$表示将$l/2$向下取整。

（3）计算返回格点向量$\boldsymbol{x}=\boldsymbol{W}_{32}v \bmod 2^3$，其中$\boldsymbol{W}_{32}$为$\mathrm{BW}_{32}$的生成矩阵Kronecker展开（固定值）。

解标签化算法为上述标签化算法的逆变换，本文不再赘述。

2.3　本文方案

基于全展开递归消除的BW格解码优化方法的流程如算法3所示。

算法3　基于全展开递归消除的BW格解码优化方法

输入　待解码的目标向量$\boldsymbol{t}\in\mathbb{C}^{N/2}$（$N=2^n$，$n\in\{5,6,7\}$对应维度32、64、128）

输出　格点向量$\boldsymbol{y}\in\mathrm{BW}_N$，满足$\left\|\boldsymbol{t}-\boldsymbol{y}\right\|\leqslant\sqrt{N/8}$

（1）预定义分层处理函数，对于每个$m=2^l(1\leqslant l\leqslant\mathrm{lb}\,N)$，定义函数$\mathcal{F}_m:\mathbb{C}^{m/2}\to\mathrm{BW}_m$，直接

实现以下操作，其中$\mathcal{F}_2$为直接对目标向量的每个分量进行四舍五入并返回。

输入分割：$\boldsymbol{t}_0=\left(t_0,t_1,\cdots,t_{m/4-1}\right),\boldsymbol{t}_1=(t_{m/4},t_{m/4+1},\cdots,\boldsymbol{t}_{m/2-1})$。

下层计算：通过预先展开的代码逻辑$\mathcal{F}_{m/2}$，直接利用计算结果进行后续操作，避免递归调用。具体而言，在编译阶段将$\mathcal{F}_{m/2}$的计算过程展开为扁平化代码，计算得到$\boldsymbol{y}_0=\mathcal{F}_{m/2}(\boldsymbol{t}_0),\boldsymbol{y}_1=\mathcal{F}_{m/2}(\boldsymbol{t}_1)$。

候选生成：基于$\boldsymbol{y}_0$、$\boldsymbol{y}_1$构造误差向量$\boldsymbol{z}_0^{\text{in}}=\varphi^{-1}(\boldsymbol{t}_1-\boldsymbol{y}_0),\boldsymbol{z}_1^{\text{in}}=\varphi^{-1}(\boldsymbol{t}_0-\boldsymbol{y}_1)$，同样通过预先展开的$\mathcal{F}_{m/2}$代码逻辑计算误差向量解码结果$\boldsymbol{z}_0=\mathcal{F}_{m/2}\left(\boldsymbol{z}_0^{\text{in}}\right)$，$\boldsymbol{z}_1=\mathcal{F}_{m/2}\left(\boldsymbol{z}_1^{\text{in}}\right)$。生成候选解$\boldsymbol{x}_0=\left[\boldsymbol{y}_0,\boldsymbol{y}_0+\varphi\boldsymbol{z}_0\right],\boldsymbol{x}_1=[\boldsymbol{y}_1+\varphi\boldsymbol{z}_1,\boldsymbol{y}_1]$，返回候选点$\boldsymbol{y}=\underset{\boldsymbol{x}\in\{\boldsymbol{x}_0,\boldsymbol{x}_1\}}{\operatorname{argmin}}\|\boldsymbol{t}-\boldsymbol{x}\|$。

（2）静态内存分配，为每层预分配连续内存块$\mathcal{M}_m\in\mathbb{C}^{m/2}$，确保64字节对齐，适配512位高级矢量扩展（advanced vector extensions 512，AVX-512）指令集，输入向量$\boldsymbol{t}$按层分割后直接写入$\mathcal{M}_m$，避免动态内存操作。

（3）分层解码执行（以$N=32$为例）

第1层（$m=2$）：调用$\mathcal{F}_2$对所有2维子向量独立解码。

第2层（$m=4$）：基于$\mathcal{F}_2$结果，调用$\mathcal{F}_4$生成4维候选格点。

…

迭代至顶层（$m=32$）：基于$\mathcal{F}_{16}$结果，最终调用$\mathcal{F}_{32}$输出解码得到的格点向量$\boldsymbol{y}$。

定理1 设目标向量$\boldsymbol{t}\in\mathbb{C}^{N/2}$满足$\operatorname{dist}(\boldsymbol{t},\mathrm{BW}_N)<\sqrt{N/8}$，则算法3输出的$\boldsymbol{y}$是唯一满足$\|\boldsymbol{t}-\boldsymbol{y}\|<\sqrt{N/8}$的格点。

定理1的证明过程如下。

（1）基底情形：当$N=2$时，BW_2对应高斯整数格$\mathbb{Z}[i]$（复空间中的一维，实空间中的二维）。此时目标向量$\boldsymbol{t}$满足$\operatorname{dist}(\boldsymbol{t},\mathrm{BW}_2)<\frac{1}{2}$，则其唯一最近格点可通过高斯整数四舍五入直接确定，同时确保了解码的正确性。

（2）归纳假设：假设对任意维度$M<N$的BW_M格，算法3在解码半径$\sqrt{M/8}$内能唯一正确解码。

（3）迭代步骤：算法3将目标向量$\boldsymbol{t}\in\mathbb{C}^{N/2}$分割为两个子向量$\boldsymbol{t}_0,\boldsymbol{t}_1\in\mathbb{C}^{N/4}$，并迭代解码得到$\boldsymbol{y}_0,\boldsymbol{y}_1\in\mathrm{BW}_{N/2}$，随后构造误差向量$\boldsymbol{z}_0^{\text{in}}$、$\boldsymbol{z}_1^{\text{in}}$，并迭代解码得到$\boldsymbol{z}_0$、$\boldsymbol{z}_1\in\mathrm{BW}_{N/2}$，最终生成候选解$\boldsymbol{x}_0$、$\boldsymbol{x}_1$，并选择欧氏距离最小的候选解作为输出。

（4）唯一性证明（反证法）：假设存在两个不同的格点$\boldsymbol{y}$、$\boldsymbol{y}'\in\mathrm{BW}_N$，均满足：$\|\boldsymbol{t}-\boldsymbol{y}\|<\sqrt{\frac{N}{8}}$、$\|\boldsymbol{t}-\boldsymbol{y}'\|<\sqrt{\frac{N}{8}}$。根据三角不等式$\|\boldsymbol{y}-\boldsymbol{y}'\|\leqslant\|\boldsymbol{t}-\boldsymbol{y}\|+\|\boldsymbol{t}-\boldsymbol{y}'\|<\sqrt{\frac{N}{8}}+\sqrt{\frac{N}{8}}=\sqrt{\frac{N}{2}}$，简化得到$\|\boldsymbol{y}-\boldsymbol{y}'\|<\sqrt{\frac{N}{2}}$。然而，由于$\boldsymbol{y}$、$\boldsymbol{y}'$是$\mathrm{BW}_N$格的两个不同的格点，它们之间的距离必须至少为$\mathrm{BW}_N$格的最小距离$\sqrt{\frac{N}{2}}$，因此$\boldsymbol{y}$与$\boldsymbol{y}'$的距离必须满足$\|\boldsymbol{y}-\boldsymbol{y}'\|\geqslant\sqrt{\frac{N}{2}}$。这与上述不等式矛盾，故假设不成立，唯一性得证。

（5）算法正确性：由于$\boldsymbol{t}_0$、$\boldsymbol{t}_1$到子格的噪声被限制在$\sqrt{\frac{N}{8}}$，迭代步骤满足归纳假设的解码条件，确保了$\boldsymbol{y}_0$、$\boldsymbol{y}_1$的正确。构造的误差向量$\boldsymbol{z}_0^{\text{in}}$、$\boldsymbol{z}_1^{\text{in}}$的噪声通过$\varphi^{-1}$缩放后，仍满足子格的唯一解码半径$\sqrt{\frac{N/2}{8}}=\sqrt{\frac{N}{16}}$，从而确保$\boldsymbol{z}_0$、$\boldsymbol{z}_1$正确。通过选择最小范数的候选解$\boldsymbol{x}_0$、$\boldsymbol{x}_1$，并结合唯一性条件，算法必然输出正确格点。

2.4 实验评估

本实验旨在通过将SCLOUD+开源代码与本

文提出的全展开优化方案进行对比，验证本文方案的正确性，以及在解码效率和解封装性能方面的提升效果。实验环境配置：操作系统采用Ubuntu22.04，处理器为Intel i9-14900K（主频3.20 GHz），为避免超线程技术对时钟周期计数的干扰，实验过程中已禁用CPU超线程功能，并通过基本输入输出系统（basic input/output system，BIOS）设置将处理器固定在3.20 GHz主频运行，以确保时钟周期计数的准确性与一致性，内存容量为192 GB，代码实现语言为C语言。为确保实验结果的可靠性与一致性，对比测试采用SCLOUD+设计者提供的开源代码进行验证。此外，每组实验均重复运行10次并取平均值，以降低随机因素导致的实验误差，保障实验数据的科学性与有效性。

在一致性测试环节，实验采用两种测试方式。其一，向BDD算法输入相同数据，对SCLOUD+开源代码与本文方案的输出结果进行比对；其二，完整调用KEM算法流程，使用相同的随机数种子，对比SCLOUD+开源代码生成的对称密钥与本文方案实现替换后生成的对称密钥。经上述两种测试验证，本文方案在功能层面与SCLOUD+设计者官方方案保持高度一致。

性能测试同样包含两个部分：首先是BDD算法性能测试，针对BW格在高维情形下，分别选取$N=32$、64、128（即BW_{32}、BW_{64}、BW_{128}），对SCLOUD+开源代码与本文方案的BDD算法进行性能测试评估。其次是在3种安全等级下开展调用BDD算法的KEM解封装性能测试。KEM算法主要由密钥生成、密钥封装和密钥解封装3个子算法构成，鉴于仅有解封装算法会调用BDD算法，故本次仅对解封装算法的性能进行测试。SCLOUD+算法设定了128 bit、192 bit、256 bit这3个安全强度等级。在密码学领域，安全强度通常以“比特（bit）”作为量化指标，其本质是指破解该密码算法所需的计算复杂度等同于暴力破解具有相应比特长度的对称加密密钥的复杂度。例如，128 bit安全等级意味着攻击者需要执行约2^{128}次操作才有可能破解该方案。

BDD算法性能测试结果见表1。表1展示了在相同含噪声输入向量条件下，针对BW格不同维度（$N=32$、64、128）开展测试所获得的实验结果。测试结果以CPU时钟周期（cycle）为度量单位，数值越小，表明性能越优。每次调用的含噪声输入向量中的噪声均独立服从高斯分布，通过重新生成随机噪声序列确保每次实验的噪声相互独立，避免出现噪声叠加现象，以此消除累积噪声对解码成功率的潜在影响。

表1　BDD算法性能测试结果

维度N	SCLOUD+开源代码/cycle	本文方案/cycles
32	7 933	1 665
64	31 461	7 243
128	147 798	30 107

由表1可知，本文提出的全展开优化方案在解码效率上显著优于SCLOUD+开源代码。以$N=32$为例，SCLOUD+开源代码消耗的CPU时钟周期数约为本文方案的4.76倍，当$N=64$时，该倍数约为4.34，当$N=128$时，倍数更高，约为4.91。该结果充分证明，本文提出的全展开优化方案在不同维度的BW格场景下，均能有效降低解码所需的CPU时钟周期，且随着维度的增加，性能优势愈发显著。这意味着，该方案不仅在低维场景下具备高效性，更在高维复杂计算中展现出卓越的可扩展性。相较于SCLOUD+开源代码，本文方案在解码效率方面具有显著优势，为实际应用中的性能优化提供了有力支撑。

SCLOUD+解封装算法性能测试结果见表2。表2呈现了在采用相同随机数种子的前提下，针对SCLOUD+的3种不同安全强度等级（128 bit、192 bit、256 bit）进行测试的结果。值得注意的是，在SCLOUD+的这3种安全强度测试场景中，均采用BW_{32}作为格编码工具。

表2 SCLOUD+解封装算法性能测试结果

安全强度/bit	SCLOUD+开源代码/cycle	本文方案/cycle
128	429 336	418 324
192	817 453	805 487
256	1 400 993	1 378 132

由表2可知，本文提出的全展开优化方案在各安全强度下均展现出优于SCLOUD+开源代码的性能表现。在128 bit安全强度下，SCLOUD+开源代码的KEM解封装过程消耗429 336 cycle，而本文方案仅需要418 324 cycle，节省了11 012 cycle，性能提升幅度达2.56%；当安全强度提升至192 bit时，SCLOUD+开源代码耗时增至817 453 cycle，本文方案则为805 487 cycle，减少11 966 cycle，性能提升1.46%；在256 bit最高安全强度下，SCLOUD+开源代码的计算开销达到1 400 993 cycle，本文方案则为1 378 132 cycle，降低22 861 cycle，性能提1.63%。尽管随着安全强度等级的提升，两种方案的计算开销均显著增加，但本文方案始终能以更低的CPU时钟周期完成KEM解封装操作。

从解封装流程来看，其包含多个关键步骤，首先用私钥解密密文，接着用哈希函数处理初步消息与公钥哈希得到验证数据，再用新的随机数重新加密消息进行验证，随后对比验证密文和原始密文来确定共享密钥生成方式，最后输出共享密钥。在这一流程中，BDD算法在整体流程中占比较小。其中，矩阵运算（含LWE模运算、多项式乘法等）占比约55%，随机数生成（涉及高斯采样与噪声生成）占比约20%，哈希计算等其他后续操作占比约15%。经理论测算，即便将BDD算法速度提升接近5倍，对整体解封装流程的性能提升幅度仅为1.6%～2.7%。此外，SCLOUD+开源代码采用固定维度BW格（N=32）实现不同安全等级的需求，128 bit、192 bit安全等级需要调用2次BW_{32}格的BDD算法，256 bit安全等级则需要调用4次BW_{32}格的BDD算法。尽管本文方案对单次BW_{32}格的BDD算法优化效果显著（从7 933 cycle降至1 665 cycle），但受多次调用机制影响，其总耗时在整体解封装流程中的占比仍相对较低。

理论研究表明，采用更高维度的BW格编码可进一步提升纠错增益，优化SCLOUD+开源代码KEM的传输效率。然而，传统递归式BDD算法在高维场景下性能衰减严重，以BW_{128}格为例，单次解码时延高达147 798 cycle，极大限制了高维格码的实际应用。例如，在256位安全等级场景下，若SCLOUD+采用BW_{128}格，则基于递归方案需要调用1次BW_{128}格的BDD，需要耗时147 798 cycle，而本文优化方案仅需要30 107 cycle，理论加速达117 691 cycle。结合解封装流程中其他模块的协同优化，整体密钥解封装效率可提升近10%。

在BW格编码的实际应用语境下，传统算法中使用的BW格维度普遍较低，相较于传统方案，BW32已属于较高维度的应用场景，而BW64和BW128更可被认定为高维格。此外，本研究涉及的KEM方案主要用于协商对称密钥，而对称密钥长度通常为32 bit，单个BW128维度已足以完成编码任务，能够满足实际应用需求。因此，在当前研究阶段聚焦于N=32、64、128这3个维度，足以体现全展开优化方案的有效性与优势，且能为后续研究提供可靠的基础。实际上，本文提出的全展开优化方案同样适用于BW256、BW512甚至更高维度，但从目前的应用需求来看，暂未发现对此类高维度进行探索的必要性。更高维度的探索可作为未来的研究方向，待后续实际需求明确、技术可行性得到充分评估后再行开展。

在硬件实现层面，当涉及现场可编程门阵列（field-programmable gate array，FPGA）和专用集成电路（application-specific integrated circuit，ASIC）等专用硬件时，相较于传统递归式BDD算法，迭代实现方式展现出显著优势。FPGA的并行处理架构与ASIC的定制化电路特性，使得迭代算法能够充分利用硬件资源。递归算法在硬

件实现时，需要频繁调用函数栈，会产生额外的存储开销和调用时延，并且递归深度受限，难以满足高维计算需求；迭代算法可以通过流水线设计和循环展开技术，将计算任务拆解为多个可并行执行的子任务，减少数据依赖和控制逻辑复杂度，有效提升硬件资源利用率。同时，迭代实现更易于硬件电路的优化设计，能够通过合理分配寄存器和存储器资源，降低关键路径时延，实现更高的时钟频率和吞吐量。这意味着，在将本文方案部署到FPGA或ASIC等硬件平台时，基于迭代实现的BDD算法将进一步放大性能优势，不仅为高维BW格编码在SCLOUD+中的应用提供更坚实的硬件支撑，也为实现抗量子安全通信系统的低功耗、高实时性目标奠定了基础。这一成果不仅验证了高维BW格编码在SCLOUD+中的应用可行性，更实现了抗量子安全性与智算网络低时延需求的平衡，为格密码技术的实际应用提供了重要的理论与实践支撑。

3　结束语

综上所述，本文通过深入研究智算网络的安全需求与后量子密码算法的特性，聚焦于非结构化LWE框架下的SCLOUD+方案，针对其依赖的BW格解码效率瓶颈，提出维度特化全展开递归消除技术，在编译期常量优化、分层硬编码及SIMD友好内存布局等方面实现创新。实验结果表明，该技术使BW格在128维场景下的解码时钟周期大幅降低，显著提升了SCLOUD+后量子密钥封装机制的效率，为智算网络构建了兼顾高效性与抗量子安全性的轻量级密钥封装机制范式，有力支撑了分布式联邦学习等低时延场景的发展。

然而，研究仍存在可进一步探索的空间。在算法优化方面，尽管本文已取得显著的性能提升，但随着智算网络规模的不断扩大和对实时性要求的持续提高，仍需要寻找更优的优化策略，进一步挖掘算法潜力。在实际应用中，跨厂商兼容性、不同网络环境下的适应性等问题仍需要深入研究，以推动后量子密码算法在智算网络中的大规模部署。未来研究可以朝着算法与硬件深度融合、探索新型数学结构在密码算法中的应用等方向展开，持续完善智算网络的安全体系，为信息时代的安全基石筑牢根基。

参考文献：

[1] Internet Engineering Task Force (IETF). A remote direct memory access protocol specification: RFC 5040[S]. 2007.

[2] BAI W, ABDEEN S S, AGRAWAL A, et al. Empowering azure storage with RDMA[C]//Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2023). Boston: USENIX Association, 2023: 49-67.

[3] Internet Engineering Task Force (IETF). Direct data placement protocol (DDP)/remote direct memory access protocol (RDMAP) security: RFC 5042[S]. 2007.

[4] RESCORLA E. The transport layer security (TLS) protocol version 1.3: RFC 8446[S]. 2018.

[5] KENT S, SEO K. Security architecture for the internet protocol: RFC 4301[S]. 2005.

[6] RESCORLA E, MODADUGU N. Datagram transport layer security (DTLS): RFC 4347[S]. 2006.

[7] SHOR P W. Algorithms for quantum computation: discrete logarithms and factoring[C]//Proceedings of the 35th Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press, 2002: 124-134.

[8] RIVEST R L, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120-126.

[9] KOBLITZ N. Elliptic curve cryptosystems[J]. Mathematics of Computation, 1987, 48(177): 203-209.

[10] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 1175-1191.

[11] 中国移动通信研究院 . 新型智算中心以太网物理层安全（PHYSec）架构白皮书[R]. 2024.
China Mobile Research Institute. Ethernet physical layer security (PHYSec) architecture white paper for new intelligent computing centers[R]. 2024.

[12] National Institute of Standards and Technology (NIST).

NISTIR 8105: report on post-quantum cryptography[R]. 2016.

[13] YESINA M V, OSTRIANSKA Y V, GORBENKO I D. Status report on the third round of the NIST post-quantum cryptography standardization process[J]. Radiotekhnika, 2022(210): 75-86.

[14] REGEV O. On lattices, learning with errors, random linear codes, and cryptography[J]. Journal of the ACM, 2009, 56(6): 1-40.

[15] National Institute of Standards and Technology (NIST). Module-lattice-based key-encapsulation mechanism standard: FIPS 203 (Final)[S]. 2024.

[16] CRAMER R, DUCAS L, PEIKERT C, et al. Recovering short generators of principal ideals in cyclotomic rings[M]//Advances in Cryptology-EUROCRYPT 2016. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016: 559-585.

[17] National Institute of Standards and Technology (NIST). FrodoKEM[R]. 2020.

[18] RAN M, SNYDERS J. Efficient decoding of the Gosset, Coxeter-Todd and the Barnes-Wall lattices[C]//Proceedings of the 1998 IEEE International Symposium on Information Theory. Piscataway: IEEE Press, 2002: 92.

[19] WANG A Y, ZHENG Z X, ZHAO C H, et al. SCLOUD: an efficient LWE-based KEM without ring/module structure[M]//Security Standardisation Research. Cham: Springer Nature Switzerland, 2025: 147-174.

[20] BARNES E S, WALL G E. Some extreme forms defined in terms of Abelian groups[J]. Journal of the Australian Mathematical Society, 1959, 1(1): 47-63.

[21] CRAMER R, SHOUP V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack[J]. SIAM Journal on Computing, 2003, 33(1): 167-226.

[22] 王学聪, 冀思伟, 李聪. 面向大模型预训练的智算网络技术研究[J]. 电信科学, 2024, 40(6): 160-172.
WANG X C, JI S W, LI C. Research on intelligent computing network technology for large-scale pre-trained models[J]. Telecommunications Science, 2024, 40(6): 160-172.

[23] FORNEY G D. Coset codes. II. Binary lattices and related codes[J]. IEEE Transactions on Information Theory, 1988, 34(5): 1152-1187.

[24] CORLAY V, BOUTROS J J, CIBLAT P, et al. On the decoding complexity of Barnes-Wall lattices[J]. arXiv preprint, 2020: 2001.05907.

[25] MICCIANCIO D, NICOLOSI A. Efficient bounded distance decoders for Barnes-Wall lattices[C]//Proceedings of the 2008 IEEE International Symposium on Information Theory. Piscataway: IEEE Press, 2008: 2484-2488.

[26] GRIGORESCU E, PEIKERT C. List decoding Barnes-Wall lattices[C]//Proceedings of the 2012 IEEE 27th Conference on Computational Complexity. Piscataway: IEEE Press, 2012: 316-325.

[作者简介]

潘洁（1978– ），女，中国移动通信集团设计院有限公司高级工程师，主要研究方向为算力网络安全和网络信息安全。

郗卓宁（1980– ），男，中国移动通信集团有限公司高级工程师，主要研究方向为 IT 信创技术、云计算及人工智能技术等。

赵占军（1972– ），男，中国移动通信集团设计院有限公司网络所所长，主要研究方向为移动通信核心网、云计算及人工智能技术、算网安全等。

卜忠贵（1976– ），男，中国移动通信集团设计院有限公司正高级工程师，主要研究方向为移动通信核心网、云计算及人工智能技术、算网安全等。

侯慧芳（1986– ），女，现就职于中国移动通信集团设计院有限公司，主要研究方向为基于算力网络的新一代网络安全关键技术。

叶兰（1979– ），女，中国移动通信集团有限公司工程师，主要研究方向为计算机科学技术网络及数据安全。

陈曦（1989– ），男，现就职于中国移动通信集团设计院有限公司，主要研究方向为网络与数据安全。

薛曌（1992– ），女，现就职于中国移动通信集团设计院有限公司工程师，主要研究方向为网络安全监测、算力网络安全、商用密码。