doi：10.3969/j.issn.1003-3106.2022.08.003
引用格式：张贺，王鹏，李思照.基于格的后量子密码系统研究[J].无线电工程，2022，52(8)：1310-1321.[ZHANG He，WANG Peng，LI Sizhao.Research on Lattice-based Post-quantum Cryptosystem[J].Radio Engineering，2022，52(8)：1310-1321]

基于格的后量子密码系统研究

张　贺，王　鹏，李思照*
（哈尔滨工程大学 计算机科学与技术学院，黑龙江 哈尔滨 150001）

摘　要： 后量子密码（Post-Quantum Cryptography，PQC）是当今密码学的发展方向，其中，基于格的PQC凭借着极强的安全性、平衡性和灵活性成为了PQC学中最为活跃的部分。由于基于格的密码自身的优势和在硬件中的高并行性，出现了大量对于格密码学的硬件实现的研究。美国国家标准与技术研究所进行了3轮PQC标准化的综合评估，其中，基于格的密码方案占比最大，有着广阔的研究前景。Saber算法、CRYSTALS-KYBER算法和CRYSTALS-Dilithium算法是第3轮决赛中的候选算法，研究者围绕这些算法中的多项式采样模块和多项式乘法器模块进行硬件设计与算法优化，以节省大量的硬件资源。此外，侧信道攻击在硬件中极易发生，尤其是轻量级设计，通过对加密和解密中关键操作的隐藏，来实现抗侧信道攻击。

关键词： 后量子密码；格基密码体制；多项式乘法；密钥封装机制；侧信道攻击
中图分类号： TN918　　**文献标志码：** A　　**开放科学（资源服务）标识码（OSID）：**
文 章 编 号： 1003-3106(2022)08-1310-12

Research on Lattice-based Post-quantum Cryptosystem

ZHANG He，WANG Peng，LI Sizhao*
（*College of Computer Science and Technology*，*Harbin Engineering University*，*Harbin* 150001，*China*）

Abstract: Post-quantum cryptography is the developing direction of today's cryptography. Among them，lattice-based post-quantum cryptography has become the most active part of post-quantum cryptography due to its strong security，balance and flexibility. Due to its own advantages as well as its high parallelism in hardware，the hardware implementation of lattice-based cryptography has been studied extensively. The National Institute of Standards and Technology of U.S. has conducted three rounds of comprehensive evaluation of post-quantum cryptography standardization，in which lattice-based cryptographic schemes account for the largest share and have a broad research prospect. Among them，Saber algorithm，CRYSTALS-KYBER algorithm and CRYSTALS-DILITHIUM algorithm are the candidates in the third round of finalists，and researchers have conducted hardware design and algorithm optimization for the polynomial sampling module and polynomial multiplier module in these algorithms to save a lot of hardware resources. Besides，side-channel attacks are very common in hardware，especially for lightweight designs，in which resistance to side-channel attacks is realized by hiding key operations in encryption and decryption.

Keywords: post-quantum cryptography; lattice-based cryptosystem; polynomial multiplication; key encapsulation mechanisms; side-channel attack

0　引言

后量子密码（Post-Quantum Cryptography，PQC）又称抗量子密码，是能够抵抗量子计算机对现有密码算法攻击的新一代密码算法。因为量子计算机的出现，现有的绝大多数公钥密码算法（RSA，Diffie-Hellman和椭圆曲线等）能被足够大且稳定的量子计算机攻破[1]，可以抵抗这种攻击的密码算法能够在量子计算和其之后时代存活下来，所以被称为“后”量子密码。

密码学的发展经过了4个阶段：古典密码学、近代密码学、现代密码学和新兴的量子密码学。其中，古典密码学的经典算法主要有凯撒加密和换位加密[2]等；近代密码学则是以DES为代表的对称密码

收稿日期：2022-03-11
基金项目：黑龙江省自然科学基金（优青项目）（JJ2019YX0922）；基础科研项目（JCKY2020208B045）
Foundation Item：National Natural Science Foundation of Heilongjiang Province of China（Outstanding Youth Project）（JJ2019YX0922）；Basic Science Research Project（JCKY2020208B045）

学,随后非对称加密算法标志着公钥加密成为新的主流[3]。公钥加密的原理一般是基于对应数学的问题在计算机上的难解性进行的,但是,量子计算机的出现使得这一难解的过程可以在有限的时间内得到解决。因此,量子密码学应运而生,它利用了单光子的量子性质[4],保证了数据传输的可证性安全。

标志着密码学进入量子密码学新时期的算法主要是 1994 年彼得·秀尔[5]提出的 Shor 算法和 1996 年Lov Grover[6]提出的 Grover 算法。Shor 算法可以以多项式时间求解周期函数的周期,而 RSA、椭圆曲线等密码体制基于的大整数分解问题和离散对数求解问题,都可以转化为周期函数求解周期的问题,因此 Shor 算法可以在有限时间内破解现有的 RSA 方案等公钥加密方案。Grover 算法又称为量子搜索算法,是一种用于非结构化搜索的算法。

PQC 按照其数学原理主要可以分为 4 种[7]:基于格的 PQC 体制、基于编码的 PQC 体制、基于哈希的 PQC 体制和基于多变量多项式的 PQC 体制。其中,基于格的 PQC 算法在安全性、公私钥尺寸和计算速度上有着更好的平衡,主要表现在: 第一,Ajtai 等人[8]在数学层面上证明了格中一般的困难问题与 NP 困难问题的难度等价,有着较强的安全性; 第二,基于格的 PQC 的应用场景丰富,在传统的公钥加密、数字签名和密钥交换等领域都有着优秀的表现; 第三,基于格的密码体制在硬件上的实现很灵活,可部署场景广泛,在资源受限和资源丰富的场合都可以应用[9]。因此,基于格的 PQC 体制被很多人认为是最有前景的 PQC 算法。

与常规的把复杂问题转换为简单问题再求解不同,密码算法的安全性完全要依赖于难解的数学问题,在有限的时间内无法求出解。与在软件上执行的 PQC 算法相比,硬件执行的效率是软件上的数十倍,甚至上百倍。在一些军方和医疗等复杂的应用场合[10],需要高速的执行来缩短加解密时间,因此关于 PQC 算法在硬件上的研究就显得十分必要。

1 基于格的 PQC 体制发展现状

量子计算机的出现对于目前的密码体制造成了巨大的冲击,各个国家都开始积极地推动 PQC 标准的制定。最具代表性的是美国国家标准与技术研究所(National Institute of Standards and Technology, NIST)。早在 2009 年,NIST 就启动了 PQC 计划来标准化一个或多个 PQC 方案。2015 年 4 月,NIST 举办了“后量子世界的网络安全”研讨会,对未来的标准化进行了讨论[11]。在 2016 年,NIST 正式宣布进行 NIST PQC 算法征集[12],邀请世界各国学者提交 PQC 方案,再经由评定选出适合标准化的算法。

1.1 评估标准

NIST 的标准化评定对于算法的评估主要分为 3 个方面,包括安全性、成本和性能以及算法和实现特征[13]。具体来说,安全性是评估 PQC 算法最为重要的因素,需要遵循传输层安全协议(Transport Layer Security, TLS)、安全外壳协议(Secure Shell, SSH)和网络密钥交换协议(Internet Key Exchange, IKE)等一系列互联网协议。NIST 对算法的安全强度定义了 5 种安全级别[14],如表 1 所示,以更好地比较提交算法的安全强度等级。成本是评估候选算法时的第二重要标准,包括计算效率和内存需求,具体包括公钥、密文和签名的大小,密钥生成,公钥和私钥的计算效率以及解密的成功率,所有提交算法的提交者需要提供 NIST 参考平台的性能评估。NIST 更希望接受灵活性大的候选算法,这有利于在各种平台上高效地运行,同时可以通过指令集的扩展来提高算法性能。

表 1 NIST 规定的算法安全级别

Tab.1 Algorithm security levels specified by NIST

安全级别	安全描述
Ⅰ	AES-128 穷举搜索
Ⅱ	SHA-256 碰撞搜索
Ⅲ	AES-192 穷举搜索
Ⅳ	SHA-384 碰撞搜索
Ⅴ	AES-256 穷举搜索

1.2 算法征集情况

截止现在,NIST 已经进行了 3 轮对广泛征集的 PQC 算法的评估,提交的算法分为公钥加密、密钥封装方案和数字签名方案。在 2017 年 12 月,NIST 对征集到的 69 份 PQC 算法方案进行了评估,其中包括 20 份数字签名方案和 49 份公钥加密或密钥封装方案[15]。经过第 1 轮评估,有 26 个候选方案进入了第 2 轮评估,包括 17 份公钥加密或密钥封装方案和 9 份数字签名方案[16],其中 12 份加密方案是基于格的 PQC 方案,占据了所有加密方案的一半。

在 2020 年 10 月,NIST 公布了第 3 轮征集的密码方案[17],具体方案如表 2 所示。其中,CRYSTALS-

KYBER,NTRU 和 SABER 是基于格的公钥加密或密钥封装方案,CRYSTALS-Dilithium 和 FALCON 是基于格的数字签名方案。基于格的加密体制成为了最有可能 PQC 标准化的方案。

表 2 NIST 第 3 轮提交算法

Tab.2 Submitted algorithms in NIST round 3

算法	公钥加密、密钥封装	数字签名
候选算法	ClassicMcEliece,Saber CRYSTALS-KYBER,NTRU	CRYSTALS-Dilithium, FALCON,Rainbow
备选算法	BIKE,FrodoKEM,HQC, NTRU Prime,SIKE	GeMSS,Picnic, SPHINCS+

NIST 预计在 2024 年之前完成对 PQC 最终入选算法的标准制定,公布 PQC 的标准化草案。届时,新的密码体系将取代现有的密码体系,来应对日益突出的量子威胁,为保密信息、数字签名和信息访问等应用保证其安全性和授权控制。

2 基于格的几种 PQC 算法

2.1 Saber 算法

Saber 算法是一种基于格的密钥封装机制(KEM)算法,在安全性级别的发展上,首先是一个基于 2 个模 p 和 q 的幂的选择明文攻击(Chosen Plaintext Attack,CPA)的安全公钥加密(PKE)方案,然后经过 Fujisaki-Okamoto 的改造为随机选择密文攻击(IND-CCA)安全的 KEM[18]。其安全性基于带舍入的学习(Learning-with-Rounding,LWR)的变体 MLWR,MLWR 问题是基于模块矩阵的 LWR 问题。LWR 是 LWE 问题的一种变体,其中误差项是通过舍入操作引入的,而不是从随机分布中获得的。

Saber 通过不同参数的设置决定了 Saber 的不同安全级别,但是三者的多项式的次数 N 都为 256,模 $q=2^{13}$,$p=2^{10}$,具体如表 3 所示。

表 3 Saber 算法的不同分支

Tab.3 Different branches of Saber algorithm

名称	安全级别	模块维度	采样范围	t
LightSaber	Ⅰ	2	[-5,5]	2^2
Saber	Ⅲ	3	[-4,4]	2^3
FireSaber	Ⅴ	4	[-3,3]	2^5

Saber 算法中出现的符号含义,β_μ 代表基于参数 μ 的二项分布,取值范围为 $[-\mu/2,\mu/2]$,概率为 $\frac{\mu!}{(\mu/2+x)!\ (\mu/2-x)}2^{-\mu}$。$x\leftarrow\beta_\mu$ 表示 x 从二项分布中随机抽样。如果用 X 替换 x,意味着一个多项式 X 从相关的二项分布中采样。$x\leftarrow U(S)$ 表示 x 是从 S 中均匀选取的。LWR 方案的样本由 $\left(a,b=\left\lfloor\frac{p}{q}\langle a,s\rangle\right\rfloor_p\right)\in\mathbb{Z}_q^n\times\mathbb{Z}_p$ 生成。

Saber 公钥加密方案包括 3 个阶段: 密钥生成、加密和解密。

算法 1: Saber 密钥生成

输入: 种子 $\mathcal{U}(0,1)^{256}$,常数向量 h

输出: 公钥 pk,私钥 sk

1.从 $\mathcal{U}(0,1)^{256}$ 均匀生成 $l\times l$ 阶多项式矩阵 A,$A=\mathrm{gen}(seed_A)\in\mathcal{R}_p^{l\times l}$

2.均匀生成 $r=\mathcal{U}(\{0,1\}^{256})$,从二项分布中随机生成 $s=\beta_\mu(\mathcal{R}_q^{l\times 1};r)$

3.缩放生成向量 b,$b=((A^{\mathrm{T}}s+h)\bmod q)$

4.对向量 b 舍入,$b\gg(\varepsilon_q-\varepsilon_p)\in\mathcal{R}_p^{l\times 1}$

5.得到公钥 pk,$pk=(seed_A,b)$

6.对 pk 进行哈希变换,$pkh=\mathcal{F}(pk)$

7.均匀生成向量 $z_1=\mathcal{U}(\{0,1\}^{256})$

8.得到私钥 sk,$sk=(s,z_1,pkh)$

在密钥生成阶段中,函数 gen 是基于 SHAKE-128 的伪随机数生成器,用于从种子中获取矩阵,生成了多项式的公共矩阵 A 和多项式的秘密向量 s。同时,对乘积 As 进行缩放和舍入得到向量 b,其中公钥由 A 和 b 组成,密钥为向量 s。

算法 2: Saber 加密操作

输入: 公钥 pk,种子 $\mathcal{U}(0,1)^{256}$,向量 r

输出: 密文 c,共享密钥 K

1.均匀选取生成消息 $m=\mathcal{U}(\{0,1\}^{256})$

2.如果 r 未被指定,则均匀生成 $r=\mathcal{U}(\{0,1\}^{256})$

3.从二项分布中随机生成 $s'=\beta_\mu(\mathcal{R}_q^{l\times 1};r)$

4.缩放生成向量 b',$b'=((As'+h)\bmod q)$

5.对向量 b' 舍入,$b'\gg(\varepsilon_q-\varepsilon_p)\in\mathcal{R}_p^{l\times 1}$

6.加密向量 $v_1'=b^{\mathrm{T}}(s'\bmod p)\in\mathcal{R}_p$

7.缩放生成 c_m,$c_m=(v_1'+h_1-2^{\varepsilon_p-1}m\bmod p)$

8.对向量 c_m 舍入,$c_m\gg(\varepsilon_p-\varepsilon_T)\in\mathcal{R}_T$

9.得到密文 $c=(c_m,b')$

10.通过哈希函数生成 $(\hat{K},c)=\mathcal{G}(F(pk),m)$

11.通过哈希函数生成 $K=\mathcal{F}(\hat{K},c)$

在加密阶段中,消息由 $v_1'=s'b^{\mathrm{T}}$(s'是专门为加密生成的向量)加密。生成的密文涉及到向量 b',b' 由 As' 产生。

算法 3: Saber 解密操作

输入: 公钥 pk,私钥 sk,密文 c,向量 z_1
输出: 共享密钥 K
1.解密向量 $v_1 = b'^{\mathrm{T}}(\mathrm{smod}p) \in \mathcal{R}_p$
2.解密生成原文消息 m',
缩放 $m' = ((v_1 - 2^{\varepsilon_p-\varepsilon_T}c_m + b_2) \bmod p)$
舍入 $m' \gg (\varepsilon_p - 1) \in \mathcal{R}_2$
3.调用加密操作生成 m' 的密文 c'
4.如果 $c = c'$,则返回共享密钥 $K = \mathcal{H}(\hat{K}, c)$
5.否则返回共享密钥 $K = \mathcal{H}(z_1, c)$

在解密阶段中,通过来自 sb' 的 v_1 的近似来恢复消息,然后再加密与已有的密文消息对比,如果相同,则接受,否则拒绝。

2.2 CRYSTALS-KYBER 算法

CRYSTALS-KYBER 是一种基于晶格的密钥封装机制,是 NISTPQC 标准化第 3 轮的 4 个决赛项目之一。基于格的密码系统一般使用多项式环来执行复杂运算,特别是 2 个高次多项式的乘法[19]。$a \leftarrow \mathrm{R}_q$ 表示 a 是从 R_q 中均匀采样的,$a \xleftarrow{\$} \mathrm{R}_q$ 表示 a 是从 R_q 中二项采样的。

数论变换(Number Theoretic Transform,NTT)操作将传统的多项式乘法运算转化为系数式乘法运算,降低了多项式乘法运算的复杂度。INTT 操作是 NTT 操作的逆操作,将系数式乘法运算转化回多项式乘法运算。CWM 算法是将 2 个二次多项式在 $\mathbb{Z}_q[x]/(x^2 - \omega^i)$ 中相乘,其中 i 随系数的指标变化[20]。

Kyber 是由公钥加密方案转化而来的 KEM,是 NIST 后量子标准中提出的算法。Kyber 算法适用于多项式环 R_q,其中 $\phi(x)$,q,n 分别为 $x^n + 1$,3 329,256。Kyber 算法的密钥生成、加密和解密操作如下。

算法 4: Kyber 密钥生成

输入: 字节数组 $(0,1)^{256}$
输出: 公钥 pk,私钥 sk

1.从 $R_q^{k\times k}$ 中均匀采样生成多项式矩阵 $\overline{A}$
2.从 R_q^k 中中心二项分布采样生成向量 s,e
3.公钥 $pk = \overline{A}s + e$
4.均匀生成 32 位字节数组 z
5.私钥 $sk = (s, pk, \mathcal{F}(pk), z)$

密钥生成操作需要 $2k$ 次 NTT 操作和 k^2 次系数式乘法运算(CWM)操作。

算法 5: Kyber 加密操作

输入: 公钥 pk,消息 m
输出: 密文 c,共享密钥 K
1.消息 m 是均匀生成的 32 位字节数组
2. $m = \mathcal{F}(m)$
3. $(\overline{K}, r) = \mathcal{G}(m, \mathcal{F}(pk))$
4.从 $R_q^{k\times k}$ 中均匀采样生成多项式矩阵 $\overline{A}$
5.中心二项分布采样生成向量 r,$e_1 \xleftarrow{\$} R_q^k$
6.中心二项分布采样生成向量 $e_2 \xleftarrow{\$} R_q$
7.计算 $u = A^{\mathrm{T}}r + e_1$
8.生成密文 c,
$c = (u, v) = (A^{\mathrm{T}}r + e_1, pk^{\mathrm{T}}r + e_2 + m)$
9.共享密钥 $K = (\overline{K}, \mathcal{F}(c))$

加密操作需要 k 次 NTT,$k^2 + k$ 次 CWM 和 $k + 1$ 次 INTT 操作。

算法 6: Kyber 解密操作

输入: 私钥 sk,密文 c,向量 z
输出: 共享密钥 K
1.解密生成原文消息 m',$m' = v - sk^{\mathrm{T}}u$
2.加密操作的常数向量 h
3.哈希变换得到 $(\overline{K}', r') = \mathcal{G}(m', h)$
4.利用加密操作生成 m' 的密文 c'
5.如果 $c = c'$,则返回共享密钥 $K = (\overline{K}', \mathcal{F}(c))$
6.否则返回共享密钥 $K = (z, \mathcal{F}(c))$

解密:对于 $sk = \overline{s}$ 和 $c = (u, v)$,利用私钥 sk 和密文 c 得到消息 m。解密操作需要 k 次 NTT,k 次 CWM 和 1 次 INTT 操作。

Kyber 算法中 k 取值为 $\{2,3,4\}$,通过改变参数 k 可以调整其安全级别。

2.3 CRYSTALS-Dilithium 算法

CRYSTALS-Dilithium 是 NIST 第 3 轮后量子数字签名候选方案中的一种,是一种基于晶格的数字签名方案,属于代数格加密套件(CRYSTALS)及密钥封装机制中的一种。Dilithium 算法的核心运算是多项式矩阵和向量的运算,安全性基于模块化带错误学习(M-LWE)和最短整数解(SIS)问题,但是不同的是所有多项式的采样都是均匀采样,多项式的生成比较简约[21]。Dilithium 算法中 $n = 256$,$q = 2^{23} - 2^{13} + 1$。Dilithium 数字签名方案的安全级别可通过表 4 中的参数进行调整。

表 4　Dilithium 算法不同安全级别下的参数设置

Tab.4　Parameter settings of Dilithium algorithm at different security levels

安全级别	模数 q	t 中删除位数 d	c 中 ±1 个数 τ	y 系数最大值 γ_1	低阶舍入最大值 γ_2	维数 (k,ℓ)	密钥范围 η	$\tau\cdot\eta\beta$	h 中 1 的个数 ω
Ⅱ	8 380 417	13	39	2^{17}	95 232	(4,4)	2	78	80
Ⅲ	8 380 417	13	49	2^{19}	261 888	(6,5)	4	196	55
Ⅴ	8 380 417	13	60	2^{19}	261 888	(8,7)	2	160	75

Dilithium 数字签名方案的 3 个核心算法是密钥生成、签名生成和签名验证,具体算法如下。

算法 7: Dilithium 密钥生成

输入: 种子 $\zeta\in\{0,1\}^{256}$

输出: 公钥 pk 和私钥 sk

1.从多项式环 R_q 均匀生成 $k\times\ell$ 阶多项式矩阵 $A\in\mathcal{R}_p^{k\times l}$

2.从多项式环 R_q 均匀生成多项式向量 s_1 和 s_2

3.计算 $t=As_1+s_2$

4.将 t 拆解为 $t=t_0+bt_1$

5.生成公钥 $pk=(A,t_1)$

6.生成私钥 $sk=(A,t_0,s_1,s_2)$

在 Dilithium 数字签名方案的密钥生成中,为了保持公钥大小较小,矩阵 A 被种子 ρ 替换,种子 ρ 生成确定性密钥,这是基于晶格的密码学中广泛使用的技术。此外,将 t 中每个系数的较低 d 位放在密钥中,来进一步降低公钥大小。

算法 8: Dilithium 签名生成

输入: 私钥 sk,消息 $M\in\{0,1\}^*$,常数向量 h

输出: 三元组 $\sigma=(z,h,c)$

1.初始化 (z,h) 为无效

2.当 (z,h) 无效时执行循环

3.生成隐蔽向量 $y\leftarrow S_{\gamma_1}^{\ell}$

4.计算 $w=Ay$,使 $w=w_1\cdot2\gamma_2+w_0$

5.$c=$ 哈希 $(M\|w_1)\in B_\tau$,使多项式 c 的 τ 个系数为 ±1,其他系数为 0

6.生成潜在签名 $z=y+cs_1$

7. $w-cs_2=w_2\cdot2\gamma_2+w_3$

8.如果 $\|z\|_\infty\geqslant\gamma_1-\beta$ 或者 $\|w_2\|_\infty\geqslant\gamma_2-\beta$ 则返回 (z,h) 无效

9.否则计算由 t 中的未知部分产生的进位向量 h,h 由 $-ct_0$ 和 $w-cs_2+ct_0$ 的高位产生

10.如果 $\|ct_0\|_\infty\geqslant\gamma_2$ 或者 $h\geqslant\omega$ 则

11.返回 (z,h) 无效

12.结束循环

在签名生成中,签名算法选择一个掩蔽向量 y,其中的系数来自 $[-\gamma_1,\gamma_1)$,来计算后续 $w=Ay$。检查多项式的最大范数是否在可接受的预定义范围和向量的最大范数,来检查签名是否泄漏有关该秘密值的信息。额外增加的一个拒绝条件是提示 h 验证算法期间 t 的哪些系数需要进位。根据安全级别的不同,平均需要尝试 3 ~ 5 次才能生成有效的签名。

算法 9: Dilithium 签名验证

输入: 公钥 pk,消息 $M\in\{0,1\}^*$,三元组 $\sigma=(z,h,c)$

输出: 有效或无效

1.$Az-ct_1$ 结果的四舍五入的高位为 w_1',$w_1'=(h,Az-ct_1\cdot2^d,2\gamma_2)$

2.如果 $\|z\|_\infty<\gamma_1-\beta$ 且 $c=\mathcal{H}(M\|w_1')$ 且 $h\leqslant\omega$,则返回有效

3.否则返回无效

在签名验证过程中,由于 t 中每个系数的低位不包含在公钥中,因此验证器利用提示 h 来执行该操作。随后,根据消息和 w_1' 重新计算质询 c,并将其与签名中提供的质询 c 进行比较。此外,检查 z 是否具有有效范数(即每个系数是否具有签名生成期间检查的最大值)。

3　硬件实现

目前,对于基于格的密码体制的高效硬件实现是当期研究的热点,大量的研究者们不但提出了 PQC 方案,也提供了硬件上实现的结果。格基密码体制易于在硬件中高效实现,主要优势在于格基密码体制中的运算主要是多项式计算,可以利用 NTT 算法实现高效快速乘法,在硬件中极大地提高了格基密码方案的吞吐量[22]。同时,硬件实现相比于软件有着高效并行处理的优势,加速处理多项式中的复杂计算,而且性能与资源上的权衡十分出色,应用场景广泛。

硬件上的开发方法主要可以分为软硬件协同设计(RTL)和高级综合(High Level Synthesis,HLS)2 种方法,在硬件上的实现平台主要有 Xilinx FPGA 上的 Virtex-7,UltraScale+和 Artix-7 等平台,完成对格基密码体制的硬件实现。

3.1　Saber 算法的硬件实现

Saber 算法是 NIST 第 3 轮评估中的候选算法之

一,研究者们对于 Saber 算法在硬件上的实现的研究很多。Roy 等人[23]设计了一个指令集协处理器架构,可以提供指令级的灵活性和模块化,易于添加或修改新的指令。在多项式的乘法优化上,对绝对值进行乘法运算,然后累加器通过系数符号位决定是加法还是减法,这样将乘法转化为移位和相加操作,优化了架构,减少周期、逻辑和寄存器数量,具体的多项式乘法器结构如图 1 所示。同时,乘法器是大规模并行的,不会受到内存访问瓶颈的影响。多项式的结构易于扩展,可以满足不同性能的需求,支持 Saber 算法的几种变体。

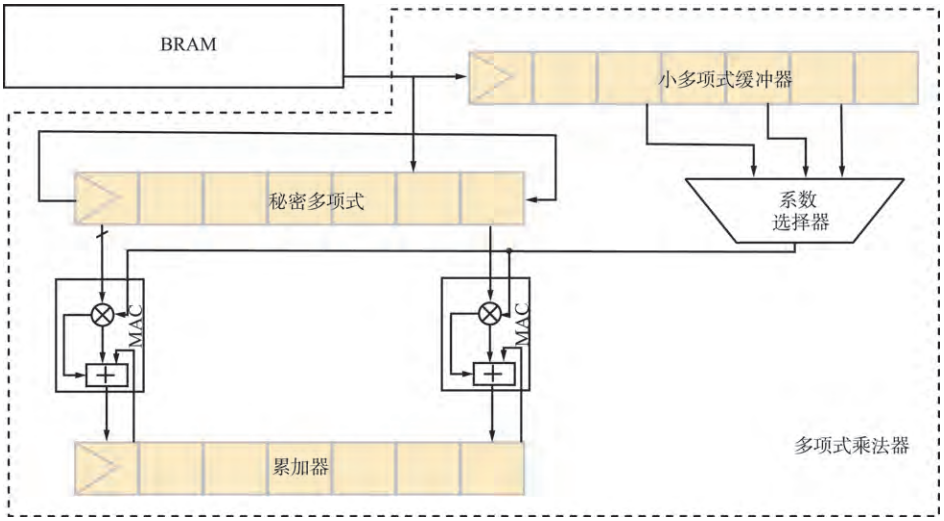


图 1　多项式乘法器结构

Fig.1　Polynomial multiplier structure

文献[24]基于多项式乘法产生的乘法输出是并行的,但需要的输出是串行的问题,提出了一种新的循环面向行处理(CROP)策略,把乘法器的输出放到下一个寄存器中存储,进行 N 步后,把所有寄存器中的输出进行循环累积,进而很容易地实现串行操作。具体来说,把 N 步操作拆为了 $r \times t$ 步,即每个 t 子累积需要 r 步循环,从而更高速地实现多项式乘法的输出操作。使用了 2 个移位寄存器(SR)和 N 个处理元件(PE),实现了 CROP 策略在硬件上的实现。

Zhu 等人[25]对之前的分层 Karatsuba 框架进行了优化,之前的这一框架实现了多项式乘法的分步操作,分为了预加、乘法和后加 3 步,但是没有考虑输入端加法器和寄存器的复用。通过调度层实现了寄存器和加法器的重用,资源上节约了 90%。同时,权衡了调度层和内核层,使得预加和后加操作变得更加紧凑,均衡了计算速度和延迟面积。Saber 算法中的多项式可以通过流水线方式生成,且与乘法硬件并行生成,减少了延迟开销和多项式的内存。同时,引入了截断乘子来实现细粒度处理。

多项式乘法时,一个多项式的系数较小(-4~+4),另一个多项式的系数为 10 位或 13 位。因此,Basso 等人[26]采取集中式乘法器结构,使用预计算的方法,把乘法结果预先计算好,这样 MAC 内部的乘法变成了简单的选择操作,显著减少了 MAC 单元的面积。数字信号处理模块(DSP)负责处理这种系数乘法,一个 DSP 包含 2 个公共多项式系数和2 个秘密系数,一个 DSP 就可以计算 4 个系数乘法,通过 128 组 DSP 可以实现 128 个周期内计算 256 系数多项式的乘法。通过使用多路复用器(MUX)将多项式乘法的并行输出变为串行输出存储到外存中。

He 等人[27]通过可伸缩矩阵源处理(SMOP),把多项式乘法中的系数相乘转换为矩阵形式。建立了一套完整的体系结构,将算术运算与信号控制流均集成到硬件上,得到紧凑的协处理器结构。整个系统包括数据流和控制流,其中重点的多项式乘法部分,将 SMOP 的算法做到硬件上,通过循环移位寄存器(CSR)、乘法和加法器(MAA)和累加器(AC)来实现多项式乘法和并行输入串行输出(PISO)。He 等人还展示了 2 个或 4 个 CSR 分组下,整个处理器的不同流程。这些实验的具体对比如表 5 所示[23-27],Saber 算法为标准的 Saber 算法。

表 5　Saber 算法硬件实现的对比

Tab.5　Comparison of Saber algorithm hardware implementation

文献	平台	时间(密钥生成、加密、解密)/μs	周期	时钟频率/MHz	LUT/10^3	FF/10^3	DSP	BRAM
[27]	UltraScale+	36.4/44.1/53.6	256	150	24.9	10.7	0	2
[28]	Stratix V	—/—/—	—	204.54	—	—	—	—
[29]	UltraScale+	10.7/14.6/17.0	859	100	34.9	9.9	85	6
[30]	UltraScale+	—/—/—	131	250	15.6	14.1	128	—
[31]	UltraScale+	48.9/63.2/78.5	—	250	10.1	7.7	0	3

Xie 等人的设计与 Basso 等人的设计相比,在性能无差距下,设计的面积延迟积(ADP)降低了 10%。

从表 5 中可以看到,Zhu 等人针对分层 Karatsuba 框架的优化,在性能上的表现最好,但是其需要的资源也最多,尤其是 DSP 和 BRAM,因此,比较适合追求性能不注重成本的应用场合。He 等人的可伸缩矩阵源处理需要的硬件资源最少,因此性能表现较差,适合轻量化的场景。Roy 等人对多项式乘法器进行优化,将乘法转化为移位和相加操作,在综合的性能和资源上的表现最优。

3.2　CRYSTALS-KYBER 算法的硬件实现

KYBER 体系结构分为 3 个主要核心: Keccak(哈希和采样)、NTT 和 Control(控制器和所有其他所需函数)。Huang 等人[28]发现在算法的整体阶段中,尤其是加密和解密阶段,存在很多可以重复使用的单元,且加密和解密过程一般并不冲突,可以复用这些单元来节省硬件成本。因此,他们将所有哈希函数集成为一个哈希模块,同时引入 BRAM 来存储中间数据,尽可能地利用 FPGA,在一个时钟周期内实现 Keccak 置换过程。在使用 Montgomery Reduction 时,其中的 3 次乘法和 1 次减法,通过在硬件中使用流水线的方法,算法的时钟周期变为原来的 1/4。

Bisheh-Niasar 等人[29]利用 KRED 和 KRED-2X 算法来实现多项式的模化约减,且只需要移位和加法操作,但是只能输出 16 位数据,为了输出 32 位数据,提出了 K2-RED 算法,减少了移位和加法操作,且 12 位的输出也意味着占用的内存更少。为了避免多项式乘法中的位反转成本,提出了 2×2 的可重构蝶形结构,支持 CT 和 GS 操作,来合并 2 层 NTT,且每层执行 2 个蝶形操作,减少硬件资源。

Yarman 等人[30]在模约减单元采用了滑动递归的方法,组合相同的位来消除冗余操作,同时简化操作,实现了一个时钟周期的延迟。设计的蝶形单元可以用于 NTT 和 INTT 操作,每个蝶形单元包括4 个双端口的 BRAM,其中 2 个 BRAM 存储第 1 个输入多项式和输出多项式,另外 2 个存储第 2 个输入多项式,同时还拥有一个 BROM,用于存储预计算和加载的旋转因子幂,通过多路复用器进行多项式的传输操作。还设计了轻量级、平衡和高性能的硬件架构,分别使用 1 个、4 个和 16 个蝶形单元,来应对不同的使用场景。

Xing 等人[31]实现了 CRYSTALS-KYBER 算法的纯手动设计硬件,不需要借助 ARM Cortex 等硬连线处理器和 RISC-V 等可重构逻辑。系统中设计使用了 2 个蝶形单元,分别用于处理 KYBER 中的一个 256 项 NTT,NTT 分解为 2 个独立的偶数索引和奇数索引,具体的 NTT 操作(上方)和 INTT 操作(下方)如图 2 所示。

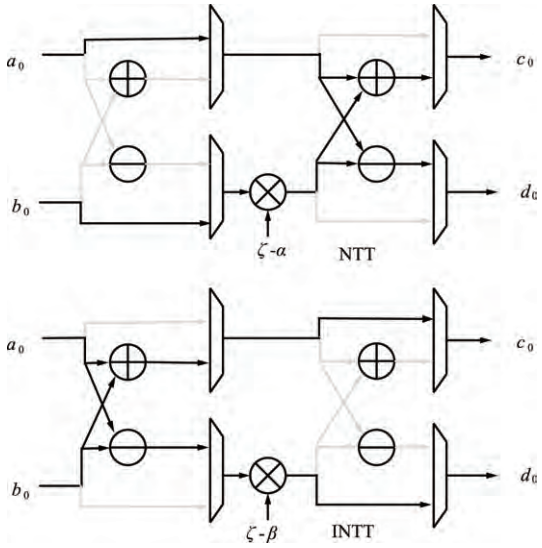


图 2　NTT 和 INTT 中的蝶形单元

Fig.2　Butterfly units in NTT and INTT

在逐点乘法中,采用了 Karatsuba 算法,降低了乘法操作的次数。同时,使用 Barrett 的模约减操作降低了 DSP 等资源的消耗。将减少系统资源消耗的关键放在了内存占用上,主要包括数据在 RAM

上的即时存储、解决读写速率不匹配而灵活多变的FIFO存储器以及重新加密阶段充分利用FIFO存储器等方法。这些实验的具体对比如表6所示[28-31]，KYBER算法为Kyber-512。

表6 KYBER算法硬件实现的对比

Tab.6 Comparison of hardware implementation of Kyber algorithm

实验方法	实验平台	周期数/10^3	时间/μs	频率/MHz	LUT	FF	slice	DSP	BRAM
集成复用	Artix-7	—/49.0/68.8	—/316/444	155	88 901	152 875	—	202	354
K2-RED算法	Artix-7	1.9/2.4/3.8	12.2/12.9/17.8	200	10 502	6 859	3 547	8	15
滑动递归	Artix-7	—/—/—	—/—/—	172	9 508	—	—	16	35
纯手动设计	Artix-7	3.8/5.1/6.7	23.4/30.5/41.3	161	7 412	4 644	2 126	3	2

从表6中可以看出，Xing等人的纯手动设计硬件实现，由于不需要ARM Cortex等硬连线处理器和RISC-V等可重构逻辑，因此在综合的性能与资源上的表现最为优秀。Bisheh-Niasar等人提出的K2-RED算法从算法层次减少了移位和加法操作，根本上降低了硬件上的资源使用。

3.3 CRYSTALS-Dilithium算法的硬件实现

Beckwith等人[32]实现了对Dilithium的不同安全级别下不同的相关参数的高性能硬件实现，算法中很多操作存在大量的数据依赖性，使得通过高度并行化来提升效率变得很难。提高性能的方向主要是优化多项式运算单元和优化操作调度。在多项式运算单元上，使用了2×2的蝶形单元、地址解析单元和FIFO存储器，一次处理2层来降低NTT中的内存访问成本，同时也不会占用过多的资源。Dilithium算法中的多项式采样需要大量的伪随机数据，使用了3个Keccak核，为向量和矩阵添加了多条拒绝通道，并行处理伪随机数据，提升了吞吐量。在优化操作调度上，将签名生成单元中的拒绝循环拆分为2级管道，加速单个操作的性能来加速整体的性能。

Land等人[33]利用DSP模块中的动态配置、预加法和单指令多数据(SIMD)的功能实现了NTT，MACC等所有的具体功能模块，减少了内存占用，省去了多项式重新采样，模块间又执行流水线处理，加快了算法实现。在NTT模块中，之前的NTT结构一般只使用DSP进行低延迟乘法，充分利用了DSP中的移位寄存器、多路复用器等单元，通过预计算蝶形单元中的旋转因子，加快了INTT操作，给NTT等所有相关的算术操作都实现了低延迟。这里同样使用BRAM来存储多项式，不同的是系数存放的间接地址，使读写冲突的数量降到了最低。在模约简模块中，利用模运算的数学关系对46位值的s进行了简化。整个架构的算术操作由NTT，MACC和矩阵向量乘法执行，检查模块会对多项式的范数进行检查，采样模块和Keccak模块负责多项式的随机生成，共同受控制单元的调度。

Ricci等人[34]用VHDL语言设计和实现了Dilithium算法中的所有底层功能，包括SHAKE-128，SHAKE-256，NTT和ExpandA$_q$等函数，而且针对硬件环境对这些函数做了进一步的优化，然后将这些函数集成到了Dilithium算法的密钥生成、签名和验证3个阶段，来提升算法的处理速度。

Zhao等人[35]提出了一种分段流水线处理方法，将算法中的操作分成多个段，硬件以流水线方式一次处理一个段。这种方法大大减少了中间结果的存储需求，并隐藏了许多操作的执行时间。优化模块，包括高速流水线NTT模块、基于BRAM的采样模块、紧凑分解模块和3个定制的模块化简模块。整体的系统结构如图3所示。

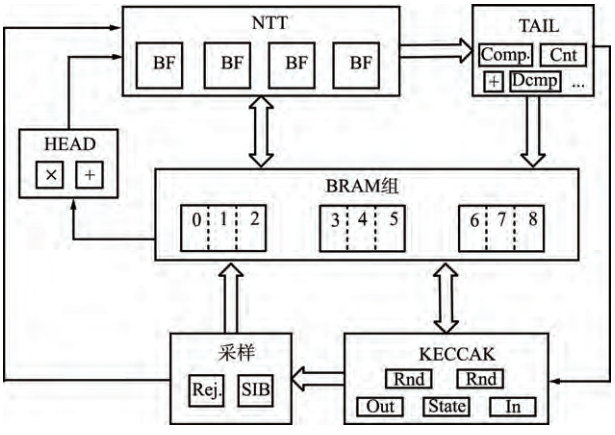


图3 系统结构

Fig.3 System structure diagram

本文将整个系统分为了NTT、头部、尾部等多个模块组件，因此采用了分段流水线的处理方法，将数字签名算法中的操作分为多个段，以流水线的方式一次处理一个段，每段通过对应模块分别执行多项式生成和采样以及NTT操作，大大减少

了中间结果的存储需求,并隐藏了许多操作的执行时间。根据模数 q 和 BRAM 的位宽关系,将 3 个阵列为一组存储多项式的 4 个系数,提高 BRAM 存储空间的利用率。同时,多项式矩阵是实时计算,而非预先计算,从而避免在 BRAM 中的无效存储。通过利用 BRAM 的空闲区域,来替换移位寄存器,折叠变换蝶形单元,提高了资源利用率,优化了 NTT 模块。

这些实验的具体对比如表 7 所示[32-35],Dilithium 算法为安全级别 2 下的算法。

表 7 Dilithium 算法硬件实现的对比

Tab.7 Comparison of hardware implementation of Dilithium algorithm

实现方法	实验平台	周期数/10^3	时间/μs	频率/MHz	LUT	FF	DSP	BRAM
高性能硬件实现	UltraScale+	4.9/29.9/6.9	19/117/26	256	53 907	28 435	16	29
可重构硬件	Artix-7	18.8/76.6/19.7	115/470/121	163	27 433	10 681	45	15
优化底层函数	UltraScale+	18.2/21.0/15.0	52/64/97	350	184 382	146 494	1 463	178
分段流水线处理	Artix-7	4.2/28.1/4.4	43/290/46	96.9	29 998	10 366	11	10

Ricci 等人使用 VHDL 语言重新设计了所有底层功能,极大地降低了 Dilithium 算法的密钥生成、签名和验证时间,但是占用的资源也极多。Zhao 等人提出的分段流水线处理,一方面极大地降低了资源消耗,同时时间上表现的差距不大,综合性能表现上最好。

4 侧信道攻击分析

格基密码体制通过格中难解数学问题保证了其算法的安全性,但是硬件的具体实现往往会面临侧信道攻击,进而影响算法的安全性和性能[36],例如芯片的瞬时功耗,以提取由实现处理的密钥[37]。侧信道攻击一般通过检测目标设备泄露的物理信息进行分析,如执行算法每一步时需要的时间或者功率轨迹、消耗的能量、发射的电磁波和产生的错误输出等,从而提取出算法的密钥。典型的侧信道攻击分为冷启动攻击[38]、故障攻击[39]、定时攻击[40]和功率分析[41]等。因此,需要对这些格基密码算法对侧信道攻击进行分析,从而找到解决对策。

物理安全性有限或没有物理安全性的轻量级应用程序更容易受到此类攻击,因为对手可以轻松收集旁道信息。不同平台的泄漏模式不同。例如,源于处理器架构的泄漏可能会影响软件,而依赖于基本组合和时序电路构建块的小故障会影响硬件实现。针对基于晶格密码体制的第 1 种侧通道攻击是对 NTRUEncrypt 实现的定时攻击[42],该攻击利用的是解密过程中哈希函数的执行时间取决于密文这一过程。

在 Saber 算法去封装的过程中,一般会长期使用私钥,这会导致容易受到侧信道分析,解决这一问题的方法是对密钥操作进行隐藏处理。有效的掩模设计分为 2 个模的幂和舍入学习的有限噪声采样,面临的主要挑战是将逐位运算与算术掩蔽相结合,要求算法在掩蔽表示之间进行安全转换。所描述的设计包括一个用于算术共享上的屏蔽逻辑移位的新原语,以及对现有的 Saber 屏蔽二项式采样器进行调整。

Abdulgadir 等人[43]采用寄存器—传输层(RTL)的方法来构建硬件,对使用私钥操作的每一单元都进行了隐藏处理。在多项式运算单元上,将多项式生成为 2 个多项式,使得乘积为另一个多项式与这 2 个多项式乘积的和。在 SHA3 单元,使用不相关的状态来提供非线性运算所需的随机性。中心二项分布(CBD)取样器单元上,使用了 Goubin 布尔型转向算术型的方法,转换采样中的几位。同时,还设计了高效的屏蔽逻辑移位单元方法实现了逻辑移位单元的隐藏。实验表明,抗 CSA 攻击下,系统使用的 LUT 和延迟分别是不抗 CSA 攻击的 2.9 倍和1.4 倍。

对于 CRYSTALS-KYBER 算法的抗侧信道攻击防护,Jati 等人[44]在整体的硬件架构上使用了广泛的资源共享,包括算术运算、控制信号及 FSM,同时实现了更小的 SHA-3 内核,LUT 和 FF 的数量减少了 70%和 50%,在具有高性能的同时有着最小的开销,同时提供了良好的抗故障性。实现抗侧信道攻击的策略有 3 种: 随机延迟,由基于 TRNG 的环形振荡器的真随机数生成器生成; 地址随机化,在系统启动/重置时使用随机数据和 fisher-yates-shuffle 技术[45]创建的地址随机化,在每条指令结束后随机更新排列; 指令随机化,引入一条名为 INSTRND 的指令使接下来的 N 条指令以随机顺序执行。同时,设计了一种基于替代置换网络(SPN)结构的良好微分偏差的 16×8 哈希来保护指令指针和内存。

Karabulut 等人[46] 发现 CRYSTALS-Dilitium 算法中的 ω-小多项式采样过程会泄漏有关系数的 -1,0 或 +1 赋值的信息,进一步证明,这一采样过程可以在单个功率测量中找到,并且可以恢复会话密钥,针对这一问题的攻击恢复系数符号的成功率超过 99.99%,将被拒绝的挑战多项式的熵降低到 39~60 位。

Chen 等人[47] 针对 CRYSTALS-Dilithium 算法提出了一个保守的方案和一个快速的两阶段攻击策略。对手可以将这 2 种方案结合起来,通过高效的混合 CPA(相关功率分析) 攻击,可以分析出 p 的最低有效位,进而恢复相关候选序列,恢复完整的秘密系数。实验表明,在合理的功率跟踪量下,该算法的置信度可达 99.99%。得益于这一策略,对手可以以 7.77 倍的加速度恢复密钥。这项工作指出,基于非保护 NTT 的多项式乘法是脆弱的。

针对不同类型的攻击和不同应用应当采取不同的对策,对于轻量级的硬件实现,这些设备往往布置在受约束的环境和移动设备上,攻击者的攻击一般是在设备的正常操作期间进行攻击,因此,主要需要进行电磁攻击的额外防护。而在其他任何嵌入式的设备当中,尤其是针对多个用户共享的平台,需要防止缓存泄露。大多数的 PQC 算法仍在开发和实验验证阶段,实验框架一直处于大量修改和扩展阶段,提出的一些针对侧信道攻击的防御对策往往只使用于特定算法,具有局限性。

5 结束语

通过对 Saber 算法、CRYSTALS-KYBER 算法和 CRYSTALS-Dilithium 算法的详细算法介绍和相关的硬件实现介绍,展示了格基量子密码学相关的背景、基础知识和发展现状。着重介绍了格基密码的硬件实现部分,对于算法中的多项式采样模块和多项式乘法模块的具体实现进行了总结分析,尤其是 NTT 算法在多项式乘法器模块中的应用,通过蝶形单元大大降低了 NTT 操作时间,借由实验数据对比展现了不同改进型密码系统的架构和优劣。最后,对硬件实现中可能面临的侧信道攻击进行了总结分析,介绍了抗侧信道攻击的硬件设计思路。

在后量子时代,基于格的 PQC 体制以其强大的安全性、平衡性和灵活性成为了 PQC 学中最为活跃的部分,其与硬件的高效结合充分展示了其灵活性和广阔的实用场景,推动着密码学与实际应用的不断结合。

参考文献

[1] 佚名.九章——踏破量子优越性 [J].工业设计,2021(9):19.

[2] 张成丽.几类格基密码方案的研究 [D].西安: 西安电子科技大学,2019.

[3] 徐铮.同源图及其在密码学上的应用 [D].合肥: 中国科学技术大学,2021.

[4] ZHONG H S,WANG H,DENG Y H,et al.Quantum Computational Advantage Using Photon [J].Science,2020,370(6523):1460-1463.

[5] SHOR P.Algorithms for Quantum Computation: Discrete Logarithms and Factoring [C] // Proceedings 35th Annual Symposium on Foundations of Computer Science.Santa Fe: IEEE,1994: 124-134.

[6] GROVER L K.A Fast Quantum Mechanical Algorithm for Database Search [C] // Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing.Pennsylvania: SIGACT,1996: 212-219.

[7] 杨嘉宇.抗量子密码的研究与应用 [D].西安: 西安电子科技大学,2021.

[8] AJTAI M.Generating Hard Instances of Lattice Problems [C] // Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing.Pennsylvania: SIGACT,1996: 99-108.

[9] NEJATOLLAHI H,DUTT N,CAMMAROTA R.Special Session: Trends,Challenges and Needs for Lattice-based Cryptography Implementations [C] // 2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS).Seoul: IEEE,2017: 1-3.

[10] GARCIA-MORCHON O,RIETMAN R,SHARMA S,et al.DTLS-HIMMO: Achieving DTLS Certificate Security with Symmetric Key Overhead [C] // European Symposium on Research in Computer Security.Vienna: Springer,2015: 224-242.

[11] ALAGIC G,ALPERIN-SHERIFF J M,APON D,et al.Status Report on the First Round of the NIST Post-quantum Cryptography Standardization Process [R].Washington,DC: US Department of Commerce,National Institute of Standards and Technology,2019.

[12] MOODY D.Post-quantum Cryptography Standardization: Announcement and Outline of NIST's Call for Submissions [C] // International Conference on Post-Quantum Cryptography-PQCrypto.Fukuoka: NIST,2016: 1-10.

[13] CHEN L,JORDAN S P,LIU Y K,et al.Report on Post-quantum Cryptography [R].Gaithersburg: US Department of Commerce,National Institute of Standards and Technol-

ogy,2016.

[14] PAUL S,SCHEIBLE P.Towards Post-quantum Security for Cyber-physical Systems: Integrating PQC into Industrial M2M Communication [C] //25th European Symposium on Research in Computer Security.Guildford: Springer,2020: 295-316.

[15] NIST.Announcing Request for Nominations for Public-Key Post-quantum Cryptographic Algorithms [S/OL].Gaithersburg: National Institute of Standards and Technology, 2016: 92787-92788 [2022-02-15].https://www.federalregister.gov/documents/2016/12/20/2016-30615/announcing-request-for-nominations-for-public-key-post-quantum-cryptographic-algorithms.

[16] TURAN M S,MCKAY K A,ÇALIK Ç,et al.Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process [R]. Gaithersburg: National Institute of Standards and Technology,2019.

[17] KUMAR M, PATTNAIK P. Post Quantum Cryptography (PQC)-An Overview: (Invited Paper) [C] //2020 IEEE High Performance Extreme Computing Conference (HPEC).Waltham: IEEE,2020: 1-9.

[18] D'ANVERS J P, KARMAKAR A, SINHA R S, et al. Saber: Module-LWR Based Key Exchange, CPA-secure Encryption and CCA-secure KEM [C] // International Conference on Cryptology in Africa.Marrakesh: Springer, 2018: 282-305.

[19] ALAGIC G,ALPERIN-SHERIFF J,APON D,et al.Status Report on the Second Round of the NIST Post-quantum Cryptography Standardization Process [R/OL].Gaithersburg: NIST,2020: 1-39 [2022-02-15].https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

[20] AVANZI R,BOS J W,DUCAS L,et al.CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation [J].Computer Science,2017(1): 1-26.

[21] BASU K,SONI D,NABEEL M,et al.Nist Post-quantum Cryptography-A Hardware Evaluation Study [J].Computer Science,Mathematics,2019(1): 1-10.

[22] BELLIZIA D,MRABET N E,FOURNARIS A P,et al. Post-quantum Cryptography: Challenges and Opportunities for Robust and Secure HW Design [C] //2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT).Athens: IEEE, 2021: 1-6.

[23] ROY S S, BASSO A. High-speed Instruction-set Coprocessor for Lattice-based Key Encapsulation Mechanism: Saber in Hardware [J].IACR Transactions on Cryptographic Hardware and Embedded Systems,2020(4): 443-466.

[24] XIE J,HE P,LEE C Y.CROP: FPGA Implementation of High-performance Polynomial Multiplication in Saber KEM Based on Novel Cyclic-row Oriented Processing Strategy [C] //2021 IEEE 39th International Conference on Computer Design (ICCD). Storrs: IEEE, 2021: 130-137.

[25] ZHU Y,ZHU M,YANG B,et al.LWRpro: An Energy-efficient Configurable Crypto-processor for Module-LWR [J]. IEEE Transactions on Circuits and Systems I: Regular Papers,2021,68(3): 1146-1159.

[26] BASSO A,ROY S S.Optimized Polynomial Multiplier Architectures for Post-quantum KEM Saber [C] //2021 58th ACM/IEEE Design Automation Conference (DAC).San Francisco: IEEE,2021: 1285-1290.

[27] HE P,LEE C Y,XIE J.Compact Coprocessor for KEM Saber: Novel Scalable Matrix Originated Processing [J]. Computer Science,2021(1): 1-16.

[28] HUANG Y,HUANG M,LEI Z,et al.A Pure Hardware Implementation of CRYSTALS-KYBER PQC Algorithm through Resource Reuse [J].IEICE Electronics Express, 2020,17(17): 1-13.

[29] BISHEH-NIASAR M, AZARDERAKHSH R, MOZAFFARI-KERMANI M.High-speed NTT-based Polynomial Multiplication Accelerator for Post-quantum Cryptography [C] //2021 IEEE 28th Symposium on Computer Arithmetic (ARITH).Lyngby: IEEE,2021: 1-10.

[30] YARMAN F,MERT A C,ÖZTÜRK E,et al.A Hardware Accelerator for Polynomial Multiplication Operation of CRYSTALS-KYBER PQC Scheme [C] //2021 Design, Automation & Test in Europe Conference & Exhibition (DATE).Grenoble: IEEE,2021: 1020-1025.

[31] XING Y, LI S. A Compact Hardware Implementation of CCA-secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA [J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021 (2): 328-356.

[32] BECKWITH L,NGUYEN D T,GAJ K.High-performance Hardware Implementation of CRYSTALS-Dilithium [C] // 2021 International Conference on Field-Programmable Technology (ICFPT).Auckland: IEEE,2021: 1-10.

[33] LAND G,SASDRICH P,GÜNEYSU T.A Hard Crystal-implementing Dilithium on Reconfigurable Hardware [C] // International Conference on Smart Card Research and Advanced Applications.Lübeck: LNSC,2021: 210-230.

[34] RICCI S,MALINA L,JEDLICKA P,et al.Implementing

Crystals-Dilithium Signature Scheme on FPGAs [C] //The 16th International Conference on Availability, Reliability and Security. Vienna: ARES, 2021: 1-11.

[35] ZHAO C, ZHANG N, WANG H, et al. A Compact and High-performance Hardware Architecture for CRYSTALS-Dilithium [J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2022(1): 270-295.

[36] SCHAMBERGER T, RENNER J, SIGL G, et al. A Power Side-channel Attack on the CCA2-secure HQC KEM [C] //International Conference on Smart Card Research and Advanced Applications. Online: Springer, 2020: 119-134.

[37] KOCHER P, JAFFE J, JUN B. Differential Power Analysis [C] //19th Annual International Cryptology Conference. Santa Barbara: Springer, 1999: 388-397.

[38] SIMMONS P. Security through Amnesia: A Software-based Solution to the Cold Boot Attack on Disk Encryption [C] //Proceedings of the 27th Annual Computer Security Applications Conference. Orlando: ACSA, 2011: 73-82.

[39] ZHANG F, ZHANG Y, JIANG H, et al. Persistent Fault Attack in Practice [J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(2): 172-195.

[40] KOCHER P C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems [C] // Annual International Cryptology Conference. Santa Barbara: Springer, 1996: 104-113.

[41] LERMAN L, BONTEMPI G, MARKOWITCH O. Power Analysis Attack: An Approach Based on Machine Learning [J]. International Journal of Applied Cryptography, 2014, 3(2): 97-115.

[42] SILVERMAN J H, WHYTE W. Timing Attacks on NTRUEncrypt via Variation in the Number of Hash Calls [C] // Cryptographers' Track at the RSA Conference. San Francisco: Springer, 2007: 208-224.

[43] ABDULGADIR A, MOHAJERANI K, DANG V, et al. A Lightweight Implementation of Saber Resistant Against Side-channel Attacks [C] // International Conference on Cryptology in India. Jaipur: Springer, 2021: 224-245.

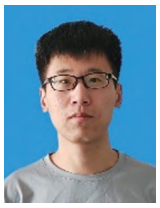
[44] JATI A, GUPTA N, CHATTOPADHYAY A, et al. A Configurable Crystals-Kyber Hardware Implementation with Side-channel Protection [J]. Cryptology ePrint Archive, 2021, 1: 1-23.

[45] HAZRA T K, GHOSH R, KUMAR S, et al. File Encryption Using Fisher-yates Shuffle [C] //2015 International Conference and Workshop on Computing and Communication (IEMCON). Vancouver: IEEE, 2015: 1-7.

[46] KARABULUT E, ALKIM E, AYSU A. Single-trace Side-channel Attacks on ω-small Polynomial Sampling: With Applications to NTRU, NTRU Prime, and CRYSTALS-DILITHIUM [C] //2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). Tysons Corner: IEEE, 2021: 35-45.

[47] CHEN Z, KARABULUT E, AYSU A, et al. An Efficient Non-profiled Side-channel Attack on the CRYSTALS-Dilithium Post-quantum Signature [C] //2021 IEEE 39th International Conference on Computer Design (ICCD). Storrs: IEEE, 2021: 583-590.

作者简介



张　贺　男，(1999—)，就读于哈尔滨工程大学电子信息专业，硕士研究生。主要研究方向：后量子密码学、FPGA 设计验证。

王　鹏　男，(1998—)，硕士研究生。主要研究方向：后量子密码学、FPGA 设计验证。

(* 通信作者) 李思照　男，(1982—)，博士，副教授，硕士生导师。主要研究方向：高性能并行计算、FPGA 设计验证。