Advances in Applied Mathematics 应用数学进展, 2023, 12(4), 1429-1437
Published Online April 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.124146

Hans汉斯

基于MLP的AKCN_MLWE算法侧信道分析

尹源源*，吴　震

成都信息工程大学网络空间安全学院，四川　成都

收稿日期：2023年3月7日；录用日期：2023年4月1日；发布日期：2023年4月11日

摘　要

在量子计算机背景下，Peter Shor提出的多项式时间算法使现有的公钥密码体制面临严重威胁，因此需要研究后量子密码算法。后量子密码算法可以抵抗量子计算机的威胁，但在实际应用中容易受到侧信道攻击。本文分析了AKCN-MLWE算法在STM32F1开发板上的实现，针对该算法解密过程中消息解码时的侧信道脆弱点，提出一种结合机器学习的侧信道分析方案。实验表明，使用PCA降维方式比SOSD提取兴趣点方式攻击效果更好。

关键词

侧信道分析，模板攻击，后量子密码，多层感知器，AKCN-MLWE

MLP-Based AKCN_MLWE Algorithm Side Channel Analysis

Yuanyuan Yin*, Zhen Wu

School of Cybersecurity, Chengdu University of Information Technology, Chengdu Sichuan

Received: Mar. 7th, 2023; accepted: Apr. 1st, 2023; published: Apr. 11th, 2023

Abstract

In the context of quantum computers, the polynomial time algorithm proposed by Peter Shor poses a serious threat to the existing public-key cryptography, so post-quantum cryptography algorithms need to be studied. Post-quantum cryptography algorithms can resist the threat of quantum computers, but are vulnerable to side-channel attacks in practical applications. This paper analyzes the implementation of AKCN-MLWE algorithm on STM32F1 development board, and proposes a side-channel analysis scheme combined with machine learning for the side-channel vulnerability point during message decoding during the decryption process of the algorithm. Experiments show that PCA dimensionality reduction is better than SOSD extraction of points of interest.

*通讯作者。

文章引用: 尹源源, 吴震. 基于 MLP 的 AKCN_MLWE 算法侧信道分析[J]. 应用数学进展, 2023, 12(4): 1429-1437.
DOI: 10.12677/aam.2023.124146

Keywords

Side Channel Analysis, Template Attacks, Post-Quantum Cryptography, Multilayer Perceptron, AKCN-MLWE

1. 引言

信息安全离不开密码保驾护航，现在广泛使用的公钥密码方案，如 RSA，ECC 等大多是基于大整数分解难题和离散对数难题等传统数论难题的，通过传统计算机解决这些难题非常困难，因此传统公钥密码体制相对安全。但 Peter Shor 于 1997 年提出了一种多项式时间算法[1]，这个算法的提出表明，一旦实用的量子计算机出现，这两个难题将得到解决，现有的公钥密码体制将受到威胁。由于这个算法的提出以及量子计算机的快速发展，密码界急需一种能够替代传统公钥密码体制的新密码，以抵抗日益严峻的量子计算机的威胁，因此催生了后量子密码算法的研究和相关标准的制定。

后量子密码算法是能够抵抗量子计算机攻击的新一代密码算法，因此也称抗量子密码算法(Quantum-Resistant Cryptography, QRC)，最早可以追溯到上世纪七十年代。美国 NIST 于 2016 年发起后量子密码算法征集，随后，中国密码学会(Chinese Association for Cryptologic Research, CACR)也发起了算法征集竞赛。目前提出的后量子密码算法大致可以分为四类[2]，分别是基于格的，如 NTRU，BLISS；基于哈希的，如 Merkle Signature；基于编码的，如 McEllice；基于多变量的，如 Rainbow。其中基于格的后量子密码由于运算时间短，占用内存少，具有较大的发展前景。

后量子密码算法可以抵抗量子计算机的威胁，但在实际应用中容易受到侧信道攻击。近年来，对后量子密码的侧信道分析方面已经有了一些工作。Kim 等[3]对 FrodoKEM 算法的恒定时间实现进行了侧信道分析，通过分析密钥生成阶段的高斯采样部分，恢复每个采样值，从而恢复完整密钥。Pessl 等[4]将加密过程作为攻击目标，以提高攻击效果，通过分析 Kyber 算法的 NTT 变换从而恢复密钥，同时还提出了三种提高攻击效率和成功率的方法。Huang 等[5]针对 NTRU Prime 算法的参考实现、优化实现等多种实现方式提出了相关能量分析、模板分析、简单能量分析等攻击方法，主要攻击的是密钥解封装过程中的多项式乘法。Ravi 等[6]针对 Round5、LAC 等格密码算法实施了选择密文攻击，攻击目标为纠错码，通过采集算法运行时的电磁泄露信息恢复密钥。Chen 等[7]以 Kyber 算法为例，提出了一种处理噪声或干扰问题的方案，这种方法相比于多数投票等传统方法可以减少约一半的问询次数。

侧信道分析方法还可以与机器学习相结合，目前在传统密码体制的侧信道分析中已经有了许多工作，对不带防护措施的密码实现和带防护措施的密码实现都有许多成功的分析成果，比如 Eleonora [8]等提出了一种基于卷积神经网络的分析策略，同时结合机器学习中经典的数据增强技术，对使用时钟抖动防护措施的算法进行了非常有效的分析。本文将机器学习的方法应用到对后量子密码算法 AKCN-MLWE 的侧信道分析中。

2. 背景知识

2.1. AKCN-MLWE 算法

AKCN-MLWE 是中国密码学会举办的后量子密码算法竞赛中进入第二轮评选的算法之一，它是基于模 LWE 的后量子密码算法。算法提炼和引出已发表的基于 LWE 及其变体的密钥封装和公钥加密

的关键成分，称之为非对称密钥共识(Asymmetric Key Consensus, AKC)。抽象化 AKC 能够帮助证明参数之间的上界关系，这些上界可以帮助判断现存的格基密钥封装和公钥加密是否还有改进空间，这种通用并且非常实用的 AKC 方案称为 AKCN。AKCN 的一般构造的性能非常接近所证明的性能最优界，并可以在特定的参数实例下达到性能最优界，AKCN-MLWE 就是将基于 LWE 和 AKCN 的密钥封装机制的模块化通用化构造用 MLWE 进行实例化，从而得到的一种高效实现。另外，算法的密钥封装部分算法机制非常简洁，没有使用纠错码或格编码等其他额外的纠错机制，这使得算法在硬件上实现时更加方便。

AKCN-MLWE 算法的加密过程如下算法 1 所示，使用与密钥生成过程相同的种子 ρ，利用函数 Parse 和函数 Sam 生成矩阵 A。由于在密钥生成中，公钥的一部分是种子 ρ，且这一部分是被公开的，这样可以确保密钥生成和加密时使用的矩阵 A 相同。使用随机数 r 作为种子生成噪音向量 r，e_1，e_2，根据算法 1 中第 5，6 行计算密文 u，v。

Algorithm 1. CPA_AKCN_MLWE Encryption Algorithm *Enc* (*pk*, *m*, *r*)
算法 1. CPA_AKCN_MLWE 加密算法 *Enc* (*pk*, *m*, *r*)

输入：公钥 pk，明文 m，随机数 r
输出：密文 c
1. $t := Decompress(t, d_v)$
2. $\hat{A} \sim R_q^{k\times k} := Parse(Sam(\rho))$
3. $(r, e_1, e_2) \sim \beta_\eta^k \times \beta_\eta^k \times \beta_\eta := CBD_\eta(r)$
4. $\hat{r} := NTT(r)$
5. $u := NTT^{-1}(\hat{A}^{\mathrm{T}} \circ \hat{r}) + e_1$
6. $v := NTT^{-1}(NTT(t) \circ \hat{r}) + e_2 + k$
7. $c_1 := Compress_q(u, d_u)$
8. $c_2 := Con(v, m, params)$
9. RETURN $c = (c_1 \| c_2)$

解密过程如下算法 2 所示，从密文中拆分出 u 和 v，利用私钥计算得到中间量 x，再经过消息解码函数 poly_tomsg 解码恢复明文消息 m。

Algorithm 2. CPA_AKCN_MLWE Decryption Algorithm *Dec* (*sk*, *c*)
算法 2. CPA_AKCN_MLWE 解密算法 *Dec* (*sk*, *c*)

输入：私钥 sk，密文 c
输出：明文 m
1. $c \to u, v$
2. $x = v - sk \times u$
3. $m = poly\_tomsg(x)$

密钥解封装过程如下算法 3 所示。

解封装过程大致分为解密和重加密两部分，先调用公钥加密部分的解密算法解密得到解密消息 m'，然后调用公钥加密部分的加密算法重加密得到密文 c'，判断重加密的密文 c'与实际输入的密文是否相同，若相等则返回由密文 c 计算得到得共享密钥 K，若不相同，则返回一个伪随机值作为共享密钥。

Algorithm 3. CCA_AKCN_MLWE_KEM Decapsulation *Dec* (*c*, *sk*)
算法 3. CCA_AKCN_MLWE_KEM 解封装 *Dec* (*c*, *sk*)

输入：密文 c
输入：私钥 $sk=\left(sk'\|pk\|H\left(pk\right)\|z\right)$
输出：共享密钥 K
1. $m':=AKCN\_MLWE\_CPAPKE.Dec\left(sk',c\right)$
2. $\left(\bar{K}',r'\right):=G\left(m'\|H\left(pk\right)\right)$
3. $c':=AKCN\_MLWE\_CPAPKE.Enc\left(pk,m',r'\right)$
4. If $c=c'$ then
5. RETURN $K:=H\left(\bar{K}'\|H\left(c\right)\right)$
6. else
7. RETURN $K:=H\left(z\|H\left(c\right)\right)$
8. end if
9. RETURN K

2.2. 侧信道分析

密码算法在实际应用中一般借助集成电路和半导体技术，通过硬件或软件在物理设备上实现，常见的密码设备有智能卡、开发板等。密码设备在执行密码算法时会泄露多种信息，常见的有能量消耗、电磁辐射、声音、时间、热量等，这些即是侧信道信息。攻击者可以检测并记录这些侧信道信息，获取有助于密码分析的关键数据。侧信道分析的概念最早是 Kocher [9]在 Crypto96 上提出的：利用密码设备实际工作时所释放的侧信道信息，恢复敏感安全参数或者密钥信息的过程被称为侧信道分析。

侧信道分析方法多种多样，主要包括能量分析、模板攻击、故障注入、计时分析、声音分析等。其中能量分析实施起来技术相对简单，且代价较小，主要方法包括简单能量分析(Simple Power Analysis, SPA)、差分能量分析(Differential Power Analysis, DPA)、相关能量分析(Correlation Power Analysis, CPA)和高阶差分能量分析(Higher-Order DPA)。一次完整的能量分析过程包含两个阶段：采集数据和数据分析。采集数据也就是获取能迹，能迹中包含关键信息，其精度由测量仪器和测试方法决定，但也由于这些的原因，能迹中还包含有大量噪声。能量分析则是把能迹数据、能量分析方法和密码算法的具体实现三部分结合起来，找出中间值，进而恢复秘密信息。

模板攻击[10]提取每个样本中所有可能的信息，因此从信息理论意义上讲可能是最有效的侧信道分析方法。实施模板攻击有一个非常重要的要求，即攻击者可以获取到与目标设备完全相同的设备，并且能够对这台设备进行编程，下发任意指令。这个要求是非常容易实现的，因为这些密码设备通常都是可以大规模生产的标准设备，攻击者可以很容易的获取到相同的设备和使用指南。这样，攻击者就可以通过这个设备对被攻击的秘密信息的所有可能值创建模板，再与从被攻击设备上获取的能迹进行模板匹配，匹配度最高的模板是被攻击的秘密信息真实值的可能性就最大。

2.3. 多层感知器

多层感知器是最经典的人工神经网络，由生物神经元模型抽象而来，可以将一组输入向量映射为一组输出向量，通常包括三层，分别为输入层、隐藏层和输出层。输入层为一层，接收待处理的数据信息，输出层也为一层，通常执行分类或预测功能。隐藏层的层数可以根据不同的需求来确定，是多层感知器真正的计算单元，这部分的设计直接影响实验分析的结果。

多层感知器的三层之间是全连接的，隐各藏层之间也是全连接的，结合权重、偏置和激活函数三个要素对输入层接收的数据进行训练。权重表示两个神经元之间连接的强度，也就是可能性的大小；偏置控制神经元的激活状态，使网络的拟合能力增强；激活函数向网络中引入非线性因素，将输出限制在某个特定的范围，对训练算法性能的影响十分显著，常用的激活函数包括 ReLU，sigmoid，tanh，softmax 等。

3. 侧信道脆弱点分析

3.1. AKCN-MLWE 算法脆弱点

AKCN-MLWE 算法的侧信道脆弱点之一是解密过程的最后一步消息解码，解密过程如第一节中算法 2 所示。解密过程由密文 c 计算出多项式 x，然后通过消息解码函数 poly_tomsg 恢复明文消息 m。

消息解码函数 poly_tomsg 将多项式 x 的每个系数 $x[k](k\in[0,255])$转换为对应的消息位 $msg[i][j]$ $(i\in[0,31], j\in[0,7])$，从而每次计算一个 msg 消息位，poly_tomsg 函数如下算法 4 所示。

Algorithm 4. poly_tomsg
算法 4. poly_tomsg

```
void poly_tomsg(unsigned char msg[32], const poly *x)
{
  uint16_t t[256];
  int i,j;
  for(i=0;i<AKCN_SYMBYTES;i++)
  {
    msg[i] = 0;
    for(j=0;j<8;j++)
    {
      t[i*8+j]=(((freeze(x->coeffs[8*i+j])<<1)+AKCN_Q/2)/AKCN_Q) & 1;
      msg[i] |= t[i*8+j] << j;
    }
  }
}
```

该函数的输入为多项式 x，包含 256 个系数，输出消息 msg 为 32 字节。在函数中，消息 msg 的所有字节先被初始化为 0，将 x 的每个系数 $x[k](k\in[0,255])$依次转换为 t，然后更新字节 $msg[i]$的第 j 位。这样，每个字节 $msg[i](i\in[0,31])$通过内层循环变量 j，每次更新一位，这样持续的一位差异可以通过电磁侧信道检测到。

当今社会常用的密码设备通常都是标准设备，攻击者可以通过合法途径获取到与被攻击设备相同的设备，这也意味着攻击者可以向相同的设备任意下发指令，没有次数限制。给定一个密文，由于密码算法是公开的，攻击者的主要目的是恢复其中隐藏的消息 m，通过恢复的消息 m 和对应的密文，可以恢复共享密钥。在本次实验中，恢复得到明文消息 m 后，根据第一节中的算法 3 即可直接计算出共享密钥。

3.2. 泄露分析

针对上述消息解码过程中的侧信道脆弱点，采集电磁泄露信息。实验将提交给中国密码学会的 AKCN-MLWE 算法的参考实现在 STM32F1 开发板上实现，该开发板采用 ARM 公司设计的 Cortex M3 内核。结合 Inspector 侧信道攻击平台、XYZ 工作台和 Lecory610Zi 示波器采集算法运行时的能量信息。通过放置在开发板芯片顶部的电磁探头测量能量泄露，使用示波器以 1.25GSam/秒的采样率收集，采集到的能量曲线如下图 1 所示。

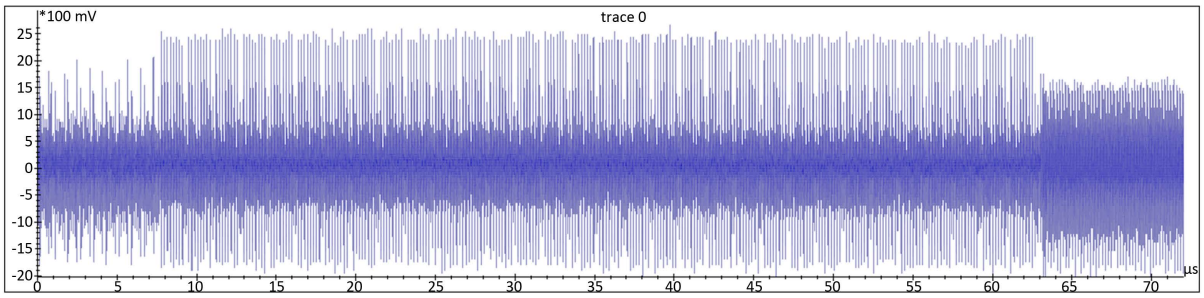

Figure 1. AKCN-MLWE algorithm power traces
图 1. AKCN-MLWE 算法能量曲线

采集能量曲线时，为了尽量降低噪声，采集到更明显的电磁泄露信息，需要先扫描 STM32F1 开发板的芯片，找到电磁信息泄露最明显的位置，将电磁探头移动到该位置再进行能迹的采集，否则采集到的电磁信号较弱，会影响后续实验的开展。

为确保实验成功，先对获取到的能量曲线进行泄露分析。SOSD 是一种常用的泄露分析方法，通过计算分组能迹的均值能耗的距离衡量分组能耗的差异，实验采用 SOSD 方法对能迹进行泄露分析，针对消息解码过程的第一个字节的分析结果如下图 2 所示。

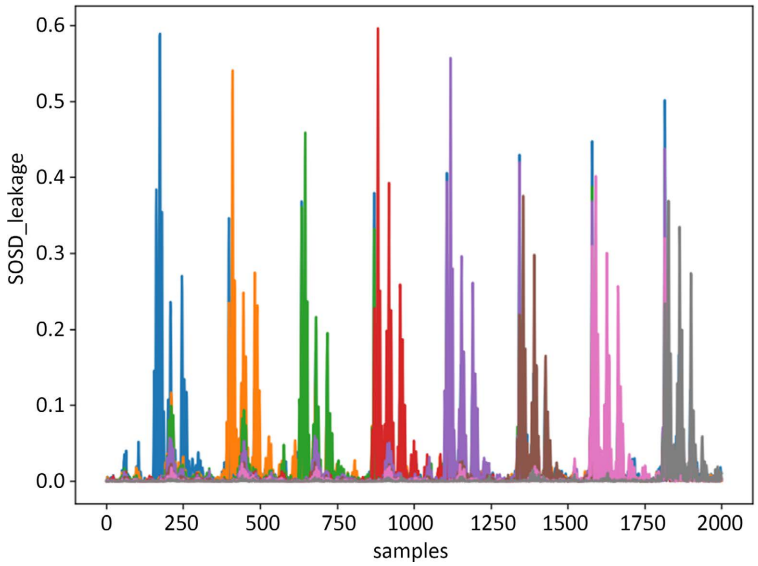

Figure 2. AKCN-MLWE algorithm electromagnetic leak detection
图 2. AKCN-MLWE 算法电磁泄漏检测

从图 2 中可以看到 8 个明显的尖峰，对应明文消息 m 的第一个字节，每个尖峰表示一个位的更新。对其余的 31 个字节做同样的泄露分析也取得了相同的结果，证实了前面对于消息解码过程存在侧信道脆弱点的分析。

4. 模型设计与实验

4.1. 实验模型及数据

本文使用的多层感知器模型基于 Keras 搭建，如下图 3 所示，包含输入层，隐藏层和输出层。输入为包含多个样本点的能量曲线，能迹数据经过 SOSD 或 PCA 处理后作为模型输入。隐藏层是数据处理的

最核心部分，每一层隐藏层均包含相同个数的神经元，均为 32 个，均使用激活函数 Leaky_Relu，向模型中引入非线性因素。Leaky_Relu 函数可以避免 Relu 函数在 $x < 0$ 时出现的 Dead Relu 现象。输出层包含 8 个神经元，对应输入字节的八个比特位，使用 sigmoid 激活函数将输出值限定在 0 到 1 的范围。

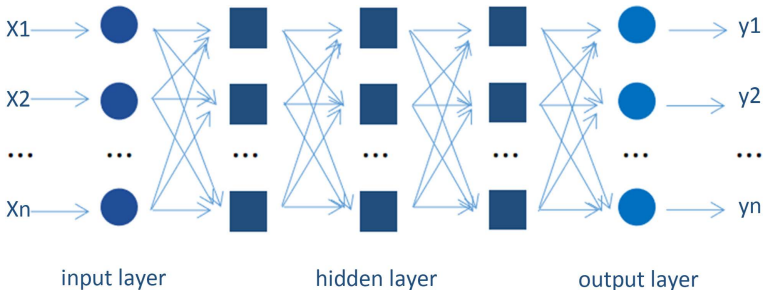

Figure 3. Multilayer perceptron basic model
图 3. 多层感知器基本模型

使用与第二节中相同的设备和设置采集实验数据，共采集 80,000 条，其中每一条能迹曲线有 90,000 个样本点，包含消息解码过程的电磁信息。由于采集到的能迹数据带有较大的噪声，需要提高信噪比后才能进行分析和实验。提高信噪比的处理方式为低通滤波和静态对齐。使用 Inpsector 平台对能迹曲线进行低通滤波处理，再对滤波后的曲线进行静态对齐处理。由于能迹左右波动范围较大，需要先进行大致对齐，然后缩小对齐参数再次对齐，以达到更好的静态对齐效果，对齐后的部分能迹曲线如下图 4 所示。

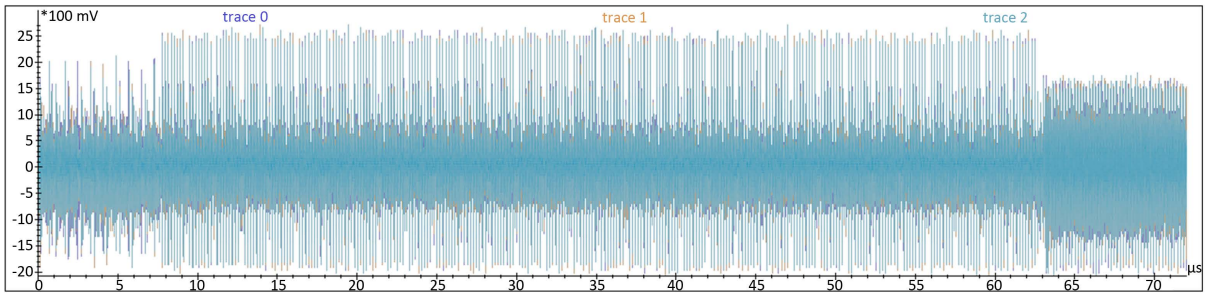

Figure 4. Static alignment of partial trace curves
图 4. 静态对齐后的部分能迹曲线

将对齐后的能迹数据转换为 H5 格式，便于后续开展实验。AKCN-MLWE 解密算法的消息解码过程通过 32 次循环依次恢复明文的 32 个字节，每个字节的恢复过程相同，因此实验先对其中一个字节进行训练，其他的 31 个字节可采用相同的方式恢复。

在静态对齐后的能迹数据中，每个字节大约包含 2000 个样本点。若直接使用全部样本，一方面会降低效率且内存占用过大，另一方面采集的能迹中包含消息解码过程的全部能耗信息，但关键信息只在某些位置出现，其余位置的能耗无法提供有效信息。因此需要先提取兴趣点，也即是特征点，选出能迹曲线中差异较大的位置作为最终使用的实验数据。常用的提取兴趣点的方法有 SOSD (sum of squared distance)，SOST (sum of squared t-test)，PCA (Principal Component Analysis)等。SOSD 在第二节中已经介绍过，此处不再赘述。SOST 使用统计学应用十分广泛的 Welch's t test，计算两个大小和方差均可能不同的集合的均值相等的概率。提取兴趣点实际上是一个降维的过程，PCA 就是一种降维的方法，通过将非正交的协方差矩阵转换为正交的特征矩阵，再选择特征值最大的 k 维，PCA 降维在许多领域都有广泛的应用，在侧信道分析中也有不错的效果。实验分别采用了 SOST 和 PCA 两种方法处理数据。

4.2. 实验分析

使用 SOSD 处理数据时，兴趣点的个数对实验能否成功有非常明显的影响。兴趣点个数太少会导致关键信息丢失，进而导致攻击成功率降低，兴趣点个数太多会导致计算协方差矩阵时计算维度过大，攻击效率低，内存占用大。经过对比，如下图 5 所示，当兴趣点个数增加时，攻击效果也随之变好，但在实际中还要考虑攻击效率，因此，当提取兴趣点的阈值设置为 0.1，兴趣点个数为 170 时，在攻击效率和成功率两方面综合考虑为最优。

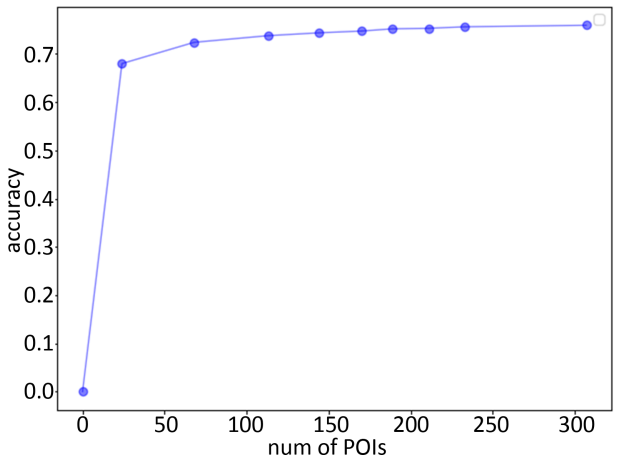

Figure 5. Attack effect of SOSD
图 5. SOSD 方式攻击效果

使用 PCA 降维处理数据时，主成分数量对攻击能否成功非常关键。如下图 6 所示，主成分数量较少时，攻击效果不太理想，随着主成分数量增加，攻击成功率先提高，后略微下降，当主成分数量为 512 时攻击效果最好，且 PCA 降维的方式攻击效果优于 SOSD 方式，攻击成功率高 8%。

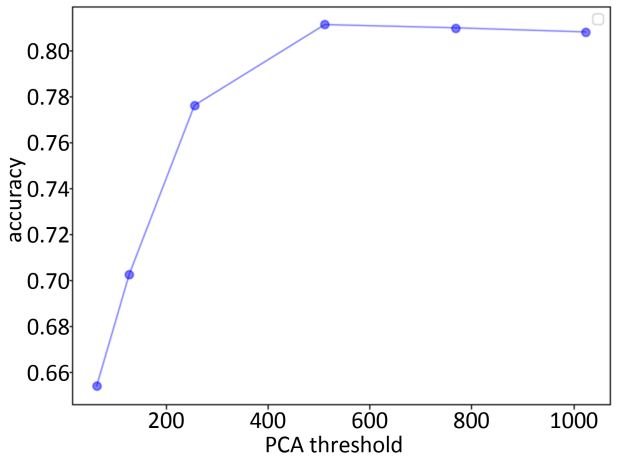

Figure 6. Attack effect of PCA
图 6. PCA 方式攻击效果

5. 结束语

本文分析研究了后量子密码算法 AKCN-MLWE 的侧信道脆弱点，针对该脆弱点提出了一种结合机器学习的侧信道分析方案。在 STM32F1 开发板上实现了 AKCN-MLWE 算法，针对这个具体实现实施了

侧信道攻击，并对比了两种不同数据处理方式的攻击效果。理论上讲，该侧信道分析方案对含有相似脆弱点的密码算法均可适用。

基金项目

四川省科技计划资助(项目号：2021ZYD0011)。

参考文献

[1] Shor, P.W. (1999) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, **41**, 303-332. https://doi.org/10.1137/S0036144598347011

[2] Kumar, R. (2019) A Survey on Post-Quantum Cryptography for Constrained Devices. *International Journal of Applied Engineering Research*, **14**, 2608-2615.

[3] Kim, S. and Hong, S. (2018) Single Trace Analysis on Constant Time CDT Sampler and Its Countermeasure. *Applied Sciences*, **8**, 1809. https://doi.org/10.3390/app8101809

[4] Pessl, P. and Primas, R. (2019) More Practical Single-Trace Attacks on the Number Theoretic Transform. In: Schwabe P. and Thériault, N., Eds., *Progress in Cryptology-LATINCRYPT* 2019, *LATINCRYPT* 2019, *Lecture Notes in Computer Science*, Springer, Cham. https://doi.org/10.1007/978-3-030-30530-7_7

[5] Huang, W.-L., Chen, J.-P. and Yang, B.-Y. (2020) Power Analysis on NTRU Prime. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, **2020**, 123-151. https://doi.org/10.46586/tches.v2020.i1.123-151

[6] Ravi, P., Roy, S.S., Chattopadhyay, A. and Bhasin, S. (2020) Generic Side-Channel Attacks on CCA-Secure Lattice-Based PKE and KEMS. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, **2020**, 307-335. https://doi.org/10.46586/tches.v2020.i3.307-335

[7] Shen, M., Cheng, C., Zhang, X., Guo, Q. and Jiang, T. (2023) Find the Bad Apples: An Efficient Method for Perfect Key Recovery under Imperfect SCA Oracles—A Case Study of Kyber. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, **2023**, 89-112. https://doi.org/10.46586/tches.v2023.i1.89-112

[8] Cagli, E., Dumas, C. and Prouff, E. (2017) Convolutional Neural Networks with Data Augmentation against Jitter-Based Countermeasures. In: Fischer, W. and Homma, N., Eds., *Cryptographic Hardware and Embedded Systems-CHES* 2017, *CHES* 2017, *Lecture Notes in Computer Science*, Springer, Cham. https://doi.org/10.1007/978-3-319-66787-4_3

[9] Kocher, P.C. (1996) Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N., Ed., *Advances in Cryptology-CRYPTO*'96, *CRYPTO* 1996, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-68697-5_9

[10] Chari, S., Rao, J.R. and Rohatgi, P. (2003) Template Attacks. In: Kaliski, B.S., Koç, Ç.K., and Paar, C., Eds., *Cryptographic Hardware and Embedded Systems-CHES* 2002, *CHES* 2002, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36400-5_3