OPTIMIST

Open Tools, Interfaces and Metrics for Implementation Security Testing
https://optimist-ose.org/

Aydin Aysu (North Carolina State University)
Daniel Dinu (Intel)
Kris Gaj (George Mason University)
Fatemeh Ganji (Worcester Polytechnic Institute)
Mona Hashemi (National University of Singapore)
Renita J (Society for Electronic Transactions and Security, India)
Dev Mehta (Worcester Polytechnic Institute)
Markku-Juhani O. Saarinen (Tampere University, Finland)
Patrick Schaumont (Worcester Polytechnic Institute)
Caner Tol (Worcester Polytechnic Institute)

WORKING DOCUMENT
TESTING POST-QUANTUM CRYPTOGRAPHY IMPLEMENTATION SECURITY

This document consolidates community-driven insights and recommendations for implementation security testing of post-quantum cryptography, detailing essential terminology, open-source tools, hardware/software implementations, and strategies for creating reproducible side-channel datasets tailored to the unique challenges of PQC algorithms.

Version History

0.5	6/4/25	First version based on Optimist Working Group Meetings

Testing Post-Quantum Cryptography Implementation Security

Post-Quantum Cryptography (PQC) has become one of the most critical fields due to the anticipated emergence of quantum computers that could be capable of rendering widely deployed public-key cryptographic algorithms ineffective. RSA, DSA, and ECC are among such algorithms. As emphasized in the recent PQC roadmap published by MITRE Corporation [1], this presents an urgent security challenge, particularly in the context of adversaries who may already be collecting and storing encrypted data with the intent to decrypt it once sufficient quantum capabilities become available. To mitigate these long-term risks, it is imperative that organizations begin transitioning to quantum-resistant algorithms now.

The roadmap in [1] outlines a structured, four-phase framework to help organizations proactively prepare for quantum-era threats to classical cryptography: (1) preparation phase that establishes mission relevance for PQC, assigning a migration lead, and engaging stakeholders; (2) Baseline understanding which identifies crypto assets, prioritizing systems, and budgeting; (3) Planning & execution phase to come up with solutions and implementing them; and (4) Monitoring & Evaluation phase that accounts for validating implementations, tracking progress, and maintaining agility as quantum capabilities evolve.

In line with the PQC roadmap, the primary goal of this document is to consolidate the need for implementation security testing for post-quantum cryptography. The document aims to help guide the community by identifying the key areas for improvement, pinpointing the existing helpful open-source efforts, and defining security testing interfaces for post-quantum cryptography implementations.

The aim is to capture the *unique* features of post-quantum cryptography testing. General concerns about testing public-key cryptosystems (e.g., those that apply to RSA/ECC solutions) are out of scope. Previous OPTIMIST documents already define the glossary and capture interface descriptions for implementation security testing campaigns. These are avoided to reduce redundancy across multiple documents.

I. Open Source Reference Implementations

A. Software Implementations

- NIST reference submissions (Round1, Round2, Round3, Round4)
- ARM Cortex-M4 implementations (pqm4)
- Java implementations (BouncyCastle)
- Various targets collaboration with Linux Foundation (PQ Code Package)
- Open SSL and variants
  - OpenSSL v3.5,
  - OpenSSL for Google (Boring SSL)
  - Embedded SSL/TLS (WolfSSL)

- Liboqs (OpenPQ Code Package · GitHubquantumsafe.org)
- PQClean – Portable implementations in C99
  https://github.com/PQClean/PQClean
- libpqcrypto - library generated by the European PQCRYPTO project
  https://libpqcrypto.org

  Conclusion: Plenty of software libraries do exist.

B. HW implementations lists

- **Test Vectors:** NIST KAT files can be used. If separate test vectors are needed for component-level testing, they can be obtained from reference software implementations. An important issue is to test for negative cases (e.g., rejections in sampling and decapsulation errors)
  - ACVP JSON test vector files for FIPS 203,204,205 (and all other NIST algorithms, like AES, SHA3) from https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files
  - For an example of usage, see Python code at: https://github.com/mjosaarinen/py-acvp-pqc
  - Test-vector compliant Python code can also be useful for creating unit tests (e.g. SystemVerilog testbenches etc) for hardware implementations (e.g. testing NTT or rejection sampling components)

- **Pre-silicon reference models:** Standard mechanisms to produce traces.
  - **CRYSTALS-Kyber**
    - Chisel, SW/HW (generic):
      https://github.com/pq-crypto/vpqc
    - VHDL, HW only (Xilinx Artix-7):
      https://github.com/xingyf14/CRYSTALS-KYBER
  - **CRYSTALS-Dilithium**
    - VHDL, HW only (Xilinx Artix 7):
      https://github.com/Chair-for-Security-Engineering/dilithium-artix7
    - SystemVerilog, HW only (Xilinx Artix/Kintex 7):
      https://github.com/GMUCERG/Dilithium
    - SystemVerilog, HW only (generic):
      https://github.com/chipsalliance/adams-bridge/
  - **SLH-DLS (SPHINCS+)**
    - SystemVerilog, SW/HW (generic)
      https://github.com/slh-dsa/sloth
  - **XMSS**
    - SystemVerilog, SW/HW (Altera Cyclone V)
      https://caslab.csl.yale.edu/code/xmsshwswriscv/

    - VHDL, HW only (Xilinx Artix 7)
      https://github.com/Chair-for-Security-Engineering/XMSS-VHDL
  - **XMSS & LMS**
    - VHDL, HW only (Xilinx Artix 7)
      https://github.com/Chair-for-Security-Engineering/XMSS-LMS-HW-Agile
  - **Classic McEliece**
    - SystemVerilog, HW only (Altera Stratix V)
      https://caslab.csl.yale.edu/code/niederreiter/
    - SystemVerilog, HW only (Xilinx Artix 7)
      https://caslab.csl.yale.edu/code/pqc-classic-mceliece/
  - **BIKE:**
    - VHDL, HW only (Xilinx Artix 7)
      https://github.com/Chair-for-Security-Engineering/RacingBIKE
  - **HQC:**
    - VHDL, HW only (Xilinx Artix 7)
      https://github.com/caslab-code/pqc-hqc-hardware
  - **FALCON:**
    - Signature only, System Verilog, HW (Xilinx Zynq-7000 ZU7EV)
      https://github.com/YiOuyang1/FalconSign

Conclusion: Hardware implementations are limited to a few possible choices for each algorithm (if any), especially compared to available software libraries. The community should incentivize open-source hardware implementations.

C. Educational tools

- Video lectures:
  - Alfred Menezes's introductory course on The Mathematics of Lattice-Based Cryptography - YouTube
- Textbooks:
  - *Cryptography and Network Security Principles and Practice*, 8th edition, William Stallings, Pearson, 2020 (Chapter: 14)
  - *Cryptography Theory and Practice*, 4th edition, Douglas R. Stinson, Maura B. Paterson, CRC Press, 2019 (Chapter 9)
  - *Post-Quantum Cryptography*, Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen, Springer, 2009
  - Understanding Cryptography – From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms, Christof Paar , Jan Pelzl , Tim Güneysu, Springer, 2024
  - Embedded Cryptography 2, Emmanuel Prouff, Guenael Renault, Mattieu Rivain, Colin O'Flynn, Wiley (Chapter 11)

        - No textbooks exist on the implementation security testing of PQC. This is a critical need.

    - NIST links:
        - NIST FIPS standards list
        - NISTIR 8413: Status Report on the Third Round of the NIST Post-Quantum Cryptography PQC Standardization Process, 09/29/2022
        - Selected Algorithms - Post-Quantum Cryptography | CSRC 2022
        - Recommendation for Stateful Hash-Based Signature Schemes: SP 800-208 | CSRC

    - Tutorials:
        - Tim Güneysu - Part I: Introduction to Post Quantum Cryptography Tutorial@CHES 2017 - Taipei
        - Post-Quantum Cryptography Trimester - Workshop 1
        - Summer School on Post-Quantum Cryptography 2017 https://2017.pqcrypto.org/school/schedule.html
        - Isogeny-Based Cryptography in Hardware by Reza Azarderaksh, CHES 2019, Atlanta, USA, 2019 https://ches.iacr.org/2019/src/tutorials/ches2019tutorial_azarderakhsh.pdf
        - Post-quantum cryptography by Michael Hamburg, hardware.io 2019: https://hardwear.io/archives/usa-2019/
        - Optimizing Crypto on Embedded Microcontrollers by Peter Schwabe & Matthias Kannwischer, hardware.io 2021: https://hardwear.io/usa-2021/training/optimizing-crypto-on-embedded-microcontrollers.php
        - Implementing Kyber and Dilithium on Microcontrollers by Matthias J. Kannwischer, CHES 2023: https://ches.iacr.org/2023/affiliated.php
        - Post-Quantum Cryptography: Implementation Attacks and Countermeasures by Daniel Dinu, Prasanna Ravi and Markku-Juhani Saarinen, HOST 2024: http://www.hostsymposium.org/host2024/program-html.php
        - Post-Quantum Cryptography: Implementation Attacks and Countermeasures by Daniel Dinu, Silvio Dragone, Prasanna Ravi and Markku-Juhani Saarinen, DAC 2024: https://61dac.conference-program.com/presentation/?id=TUT109&sess=sess271
        - CPA Attack on Hardware Implementation of ML-DSA in Post-Quantum Root of Trust by Merve Karabulut and Reza Azarderakhsh, HOST 2025: http://www.hostsymposium.org/program-html.php

    - Central index for research papers: PQC Zoo covers some early papers. Nothing exists for recent papers. This is a critical need.
    - Overview Papers on Implementations and Implementation security:

        - Ravi et al 2024. Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results
        - Chowdhury et al 2021. Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions. Journal of Cryptographic Engineering, pp.1-37.
        - Ravi et al, 2021. Lattice-based key-sharing schemes: A survey. ACM Computing Surveys (CSUR), 54(1), pp.1-39.
        - Nejatollahi, Hamid, et al. "Post-quantum lattice-based cryptography implementations: A survey." ACM Computing Surveys (CSUR) 51(6), pp. 1-41.
        - Konstantopoulou et al. 2025. Review and Analysis of FPGA and ASIC Implementations of NIST Lightweight Cryptography Finalists. ACM Computing Surveys, 57(10), pp.1-35.
    - Major Conferences publishing papers on the implementation security of PQC: TCHES, HOST, DAC, DATE, PQCrypto, FPL, CCM, NIST Workshops:
        - Sixth PQC Standardization Conference, September 24-26, 2025
        - Fifth PQC Standardization Conference, April 10-12, 2024
        - Fourth PQC Standardization Conference, Nov. 29 –Dec. 1, 2022
        - Third PQC Standardization Conference, June 7-9, 2021
        - Second PQC Standardization Conference, Aug. 22-25, 2019
        - First PQC Standardization Conference, Apr. 11-13, 2018
        - Workshop on Cybersecurity in a Post-Quantum World, Apr. 2-3, 2015

Conclusion: There are significant resources out there. But they are scattered and buried. A structured, central index with useful resources can help accelerate the learning curve. A textbook dedicated to the implementation security testing of PQC with hands-on components is needed.

II. Open Source Datasets

We identify a critical gap in open-source datasets for the implementation security testing of post-quantum cryptography. The following are our guidelines for generating this dataset for side-channel analysis. Fault injection datasets would also be useful, but need more research maturity before discussing specifics.

Dataset Scope and Strategy

- Full algorithm execution traces may result in large, impractical datasets.
- A more effective approach may involve generating side-channel traces for unit-level testing.

Proposed Units for Trace Collection

- Targeted units include:
  - Number Theoretic Transform (NTT)
  - Samplers (e.g., binomial, ternary, Gaussian, rejection sampling)
  - Hash functions (particularly those used in PRFs for PQC)
  - Comparison functions (e.g., for rejection sampling)
  - Arithmetic-to-Boolean and Boolean-to-Arithmetic conversions used in masking
- Each unit can be further subdivided based on algorithmic variations and implementation strategies.

Implementation Variants

- Each unit should ideally be implemented in four forms:
  1. Baseline (no countermeasures)
  2. First-order masking
  3. Shuffling
  4. Higher-order masking

Hardware Measurement Emphasis

- While software-based measurements are acceptable, there is a strong community preference for hardware-based side-channel measurements, particularly from FPGA implementations

Metadata Requirements

- Trace datasets should be accompanied by detailed metadata, including:
  - FPGA implementation specifics
  - Description of applied defense techniques
  - Methodology for input generation
- The metadata description could further follow details in our earlier document.

Examples of existing datasets: Kyber https://eprint.iacr.org/2025/811

III. Open Source Analysis Techniques

We identify several challenges in extending common testing methods for post-quantum cryptography. TVLA as such has been used extensively in both academic work as well as in the industry evaluation of side-channels. However, the TVLA application for post-quantum sub-routines is non-trivial and may generate incorrect results. Open source analysis techniques and establishing best practices are critical as PQC deployments are emerging.

Several challenges stem from the inherent algorithmic randomness and complex data dependencies in PQC schemes like Kyber. Traditional fixed-vs-random testing paradigms struggle to isolate sensitive intermediates due to probabilistic operations such as sampling and rejection, which dilute leakage signals. Additionally, PQC algorithms often require significantly more traces for meaningful statistical analysis (which makes them more susceptible to false positives), and the variability in implementation platforms (e.g., softcore RISC-V on FPGAs) introduces noise and architecture-specific artifacts. These issues limit the effectiveness of conventional TVLA and highlight the need for tailored test strategies, better trace preprocessing, and complementary leakage detection methods.

At a higher level, there may be confusion about which variables in the algorithms are sensitive (vs. public) and must be protected. This is especially important for lattice-based encryptions with rejection subroutines. Having general guidelines on the critical variables and the number of measurements for testing could benefit future infrastructures.

The sources below identify some challenges and efforts in TVLA tuning for PQC:

1. WiP: Applicability of ISO Standard Side-Channel Leakage Tests to NIST Post-Quantum Cryptography https://www.computer.org/csdl/proceedings-article/host/2022/09839849/1FHqIHDJuN2
2. Evaluating the effectiveness of Test Vector Leakage Assessment when performed on Kyber running on a softcore RISC-V processor on an FPGA https://essay.utwente.nl/98334/1/Becker_BA_EEMCS.pdf
3. Continuous integration side channel testing for ML-KEM , PQShield https://www.creachlabs.fr/sites/default/files/public/media/document/2024-12/ecw2024-pqc-zijlstratimo-20241120.pdf

References

[1] The MITRE Corporation, Post-Quantum Cryptography (PQC) Migration Roadmap, 2025 [Online] https://pqcc.org/wp-content/uploads/2025/05/PQC-Migration-Roadmap-PQCC-2.pdf