AGH University of Krakow

PQ-NEXT

MIST
WORKSHOP ON MOBILE AND IOT SECURITY TECHNOLOGIES

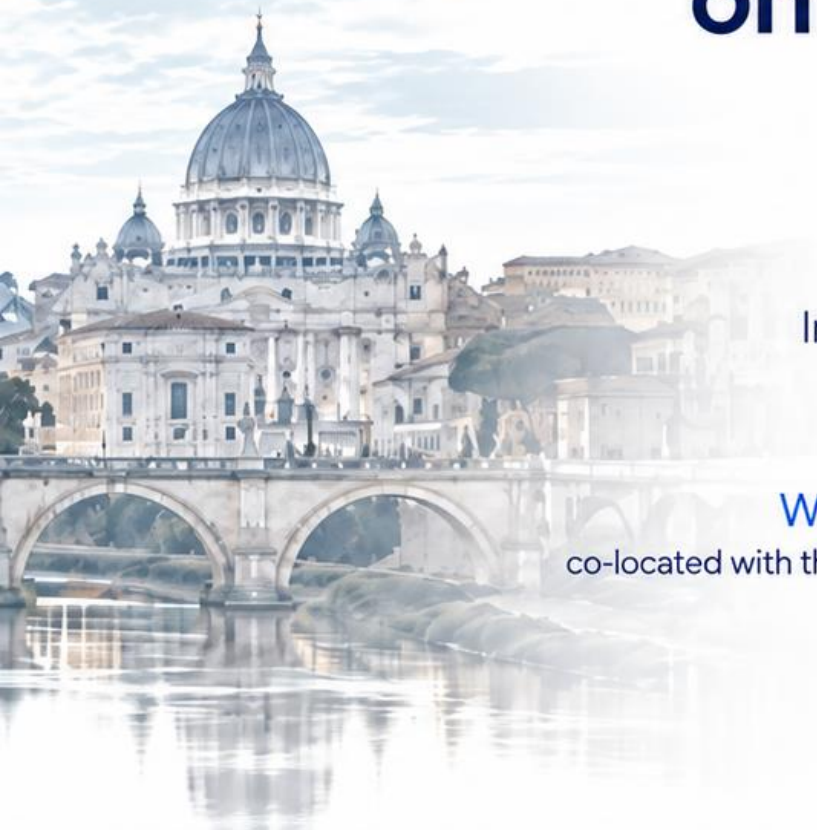
ESORICS

Lightweight Feature-Based Power Side-Channel Analysis of ML-KEM on Embedded Platforms

Daniel Karcz · Marcin Niemiec

AGH University of Kraków

Institute of Telecommunications and Cybersecurity





MIST 2026

Workshop on Mobile and IoT Security Technologies

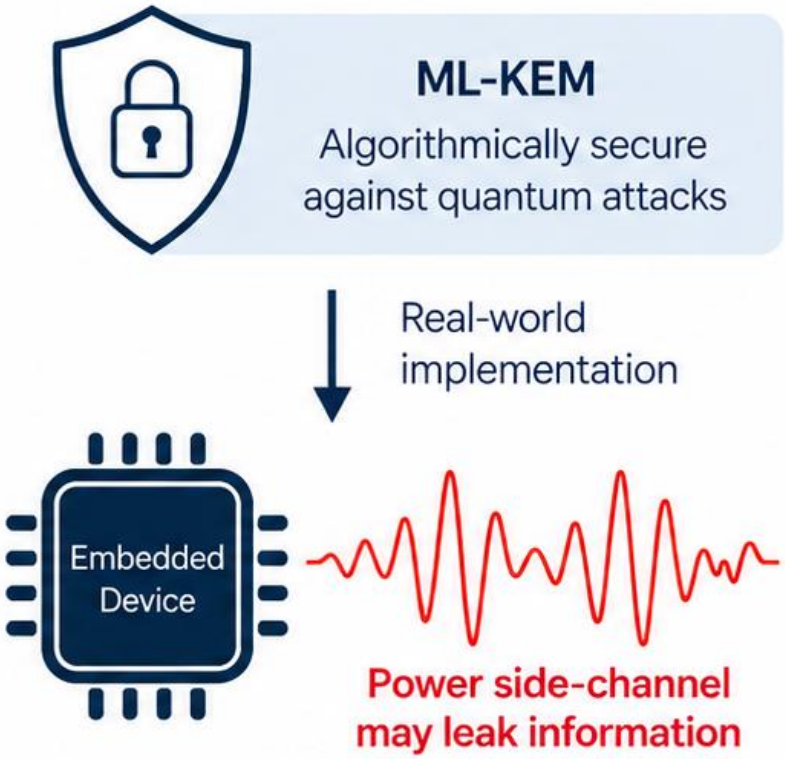
co-located with the 31st European Symposium on Research in Computer Security (ESORICS 2026)

Rome, Italy · September 18, 2026

Motivation

Why Power Side-Channels Matter for PQC

- **Post-quantum cryptography (PQC)** algorithms, such as ML-KEM, are designed to be secure against quantum adversaries.
- However, **algorithmic security does not guarantee** a secure implementation.
- **Physical side-channels,** in particular power consumption, can leak sensitive information due to data-dependent behavior in real-world implementations.
- **Embedded and IoT devices** have limited resources, so lightweight implementations are needed, which may amplify side-channel leakage.
- **Evaluating the susceptibility** of PQC implementations to power side-channels is crucial for their real-world deployment.

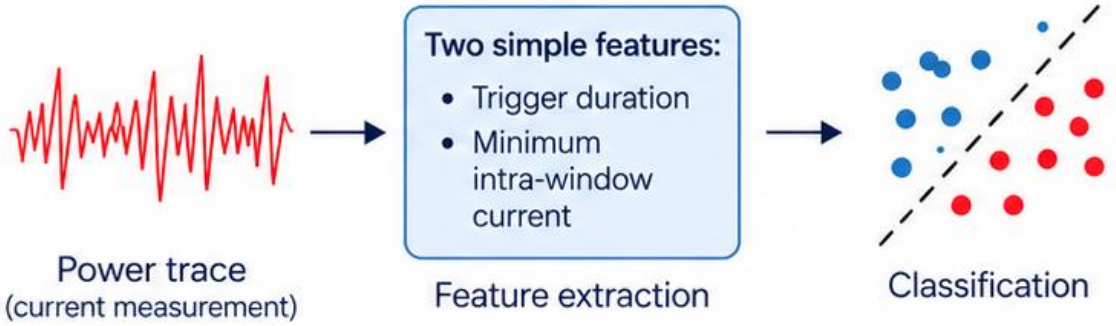


Research Question / Contribution

Can simple power features reveal the behavior of ML-KEM?

Research Question

Can **simple, manually selected features** extracted from power consumption measurements sufficiently distinguish the behavior of ML-KEM on embedded devices, **without using complex machine learning models?**



Our Contribution

Lightweight feature-based approach
We investigate whether simple statistical features can effectively capture differences in power traces of ML-KEM operations on embedded platforms.

Practical evaluation
We perform measurements on a real embedded device (ESP32-C6) running ML-KEM and evaluate multiple lightweight classifiers.

Security-relevant cases
We show that such features can distinguish, among others, different decapsulation behaviors (e.g., valid vs. corrupted ciphertext).

Goal
Provide an efficient and practical method for assessing side-channel susceptibility of PQC implementations on resource-constrained devices.

We show that simple power features are sufficient to effectively distinguish ML-KEM behavior on embedded platforms, without relying on complex machine learning models.

AGH University of Kraków
Institute of Telecommunications and Cybersecurity
Daniel Karcz · Marcin Niemiec

Experimental Setup

Hardware, target implementation and measurement configuration

Target Device



ESP32-C6

- RISC-V 32-bit (single-core)
- 160 MHz
- Wi-Fi 6 / Bluetooth 5
- Resource-constrained embedded device

Cryptographic Algorithm



ML-KEM-512

- NIST FIPS 203 (ML-KEM)
- Reference implementation (liboqs 0.14.0)
- Operations: key generation, encapsulation, decapsulation

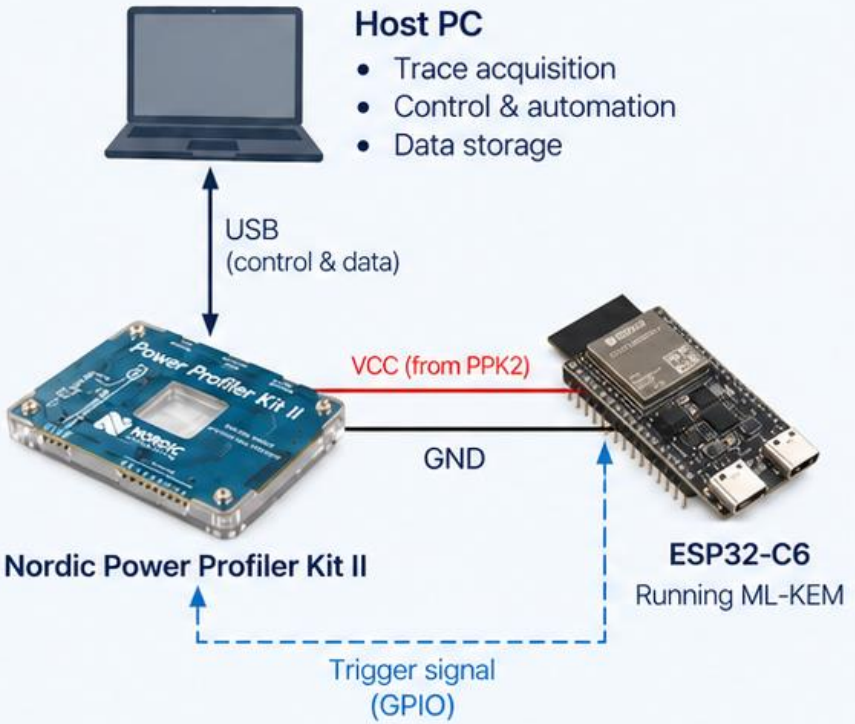
Power Measurement



Nordic Power Profiler Kit II

- High-precision current measurement
- Source mode
- Low-noise measurement front-end

Measurement Setup



Measurement Details

Measured Signal
Current consumption (shunt resistor inside Power Profiler Kit II)

Triggering
GPIO signal from ESP32-C6 set HIGH before the operation and LOW after the operation (e.g., decapsulation)

Captured Operations

- Key generation
- Encapsulation
- Decapsulation (valid ciphertext)
- Decapsulation (corrupted ciphertext)

Dataset
Multiple traces per operation, collected in a controlled environment (100 traces per operation for main classification: 300 total)

Controlled and reproducible measurements on a real embedded platform
enable analysis of power side-channels in ML-KEM using lightweight feature-based methods.

Measurement & Trace Acquisition

Data collection procedure and example power trace

Trace Acquisition Procedure

1 Execute ML-KEM operation

Key generation, encapsulation or decapsulation is executed on the ESP32-C6.

2 Trigger execution window

A GPIO signal is asserted immediately before the cryptographic operation and deasserted immediately after completion.

3 Acquire power trace

The Nordic Power Profiler Kit II records current consumption together with the digital trigger state. Raw measurements are stored as .ppk2 and selected trace segments are exported as .csv (Timestamp (ms), Current (µA), D0–D7).

Example Power Trace (ML-KEM-512)

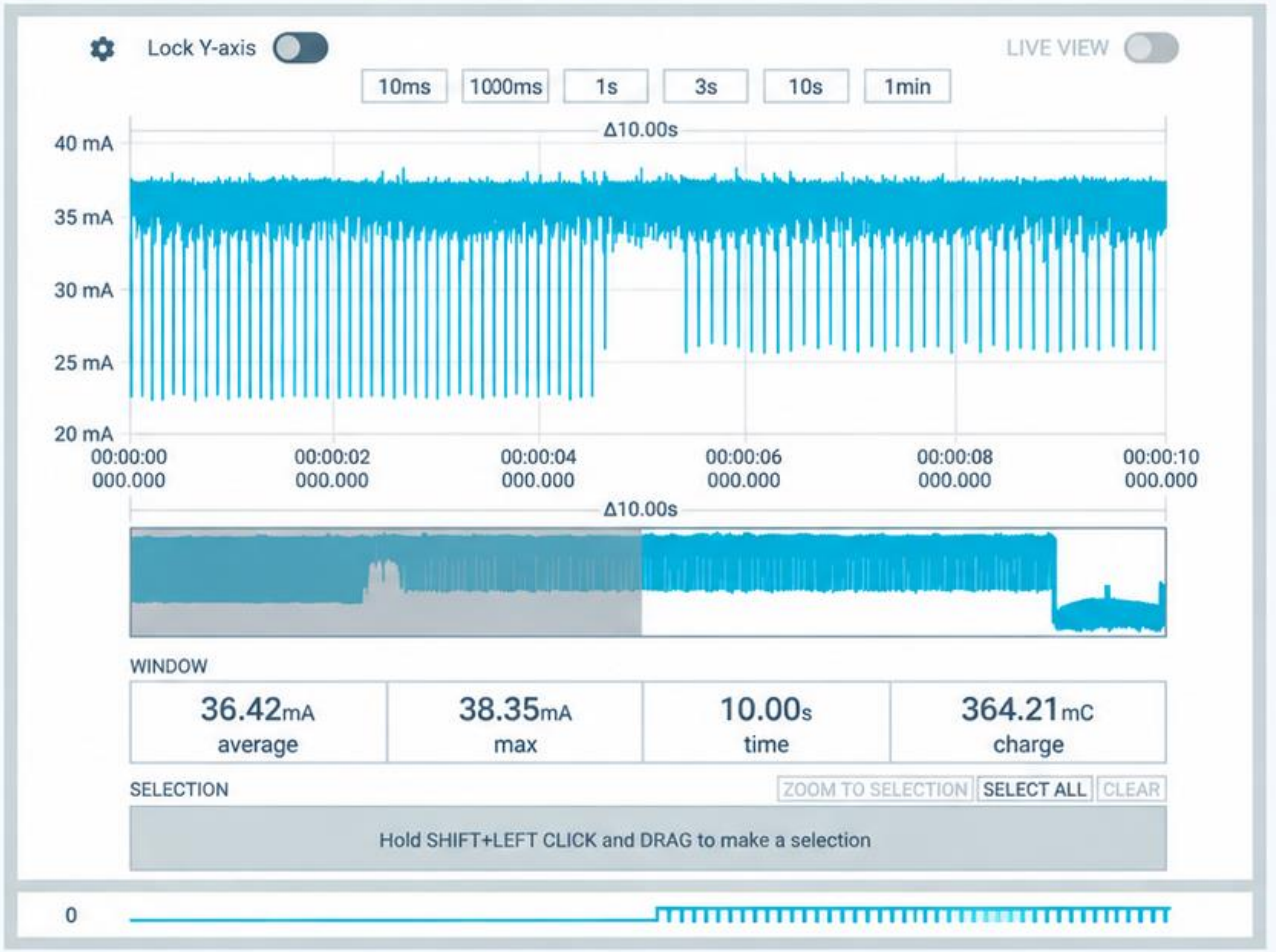


Fig. 4. Consecutive ML-KEM encapsulation (left) and decapsulation (right) trigger-aligned execution windows.

Power Trace Characteristics

Operation-level differences in ML-KEM-512

Comparative Trace Statistics

Table 2. Comparative summary of ML-KEM operation trace statistics.

Operation	Duration [ms]	Mean Current [mA]	Peak Current [mA]	Min Current [mA]
Key generation	103.31	36.39	37.60	22.85
Encapsulation	106.63	36.40	37.59	23.02
Decapsulation	129.36	36.47	37.71	32.76

Key Observations

1. **Execution duration**
   Decapsulation takes noticeably longer (129.36 ms) than key generation (103.31 ms) and encapsulation (106.63 ms).

2. **Minimum current**
   Decapsulation exhibits a significantly higher minimum current (32.76 mA) compared to key generation (22.85 mA) and encapsulation (23.02 mA).

3. **Mean and peak current**
   Mean and peak current values are similar across all operations (≈36.4 mA and ≈37.6 mA, respectively).

ML-KEM operations exhibit distinct and repeatable power characteristics,
particularly in execution duration and minimum current.

AGH University of Kraków
Institute of Telecommunications and Cybersecurity
Daniel Karcz · Marcin Niemiec

Feature Extraction

Two simple features for ML-KEM-512 classification

Why these features?

Simple and effective
Two scalar features are sufficient to distinguish ML-KEM operations.

No complex processing
No full waveform alignment or advanced signal processing ic required.

Lightweight
Low-dimensional representation, suitable for embedded devices.

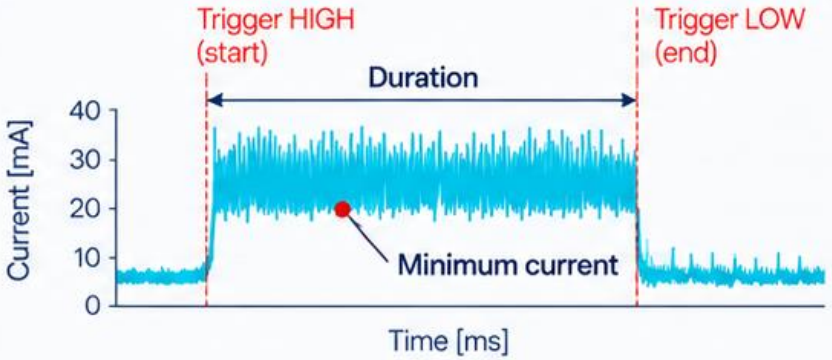
Proven in our scenario
Enables classification of key generation, encapsulation, decapsulation, and valid vs. corrupted ciphertext.

Extracted Features

1. **Trigger Duration [ms]**
Time between the rising and falling edges of the GPIO trigger signal, corresponding to the execution time of the cryptographic operation.

2. **Minimum Intra-Window Current [mA]**
Minimum measured current value within the trigger-defined execution window.

Illustration of feature extraction



Feature Vector and Usage

Feature vector (2-dimensional)

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

x_1: Trigger duration [ms]
x_2: Minimum current [mA]

Used in classifiers
- Logistic Regression
- Random Forest

Output
- Operation type (key generation, encapsulation, decapsulation)
- Valid vs. corrupted ciphertext

Only two scalar features (trigger duration and minimum intra-window current) are used
for operation-level classification of ML-KEM-512 power traces.

Main Results

High classification accuracy using two simple features

Feature-Space Separability of ML-KEM Operations

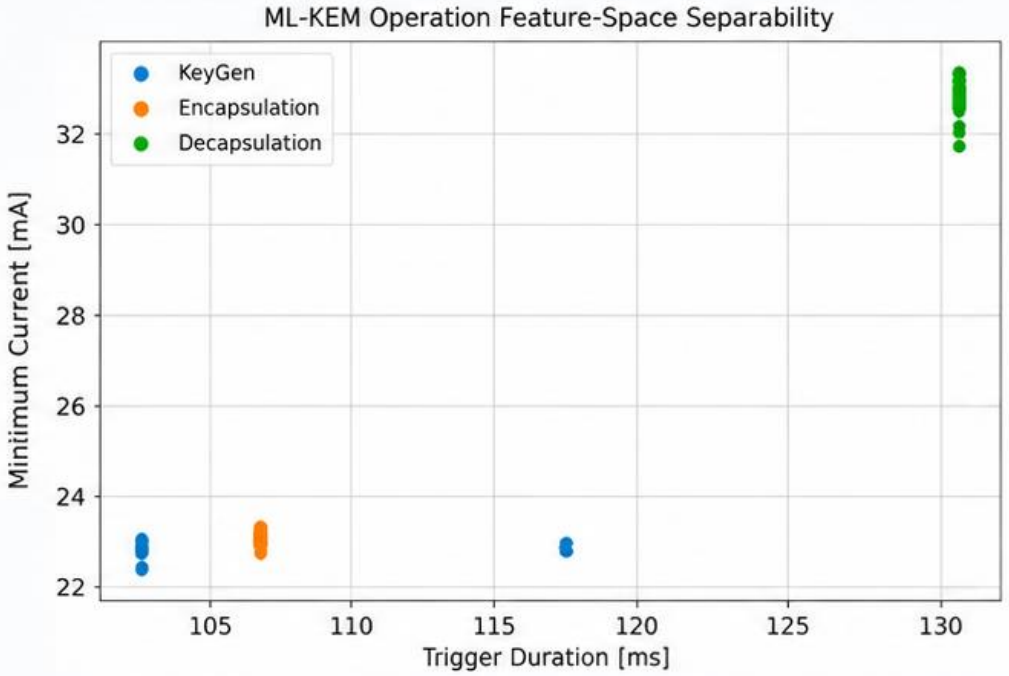


Fig. 5. Two-dimensional feature-space clustering of ML-KEM operation classes using trigger duration and minimum intra-window current. Decapsulation forms a fully isolated cluster, while key generation exhibits a bimodal distribution, indicating multiple stable execution modes.

Classification Results

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1-score [%]
Logistic Regression	98.67	98.79	98.67	98.66
Random Forest	99.67	99.70	99.67	99.67

Table 3. Using stratified 10-fold cross-validation, Logistic Regression achieves 98.67 percent accuracy, while Random Forest reaches 99.67 percent.

Dataset and Evaluation

300 traces
100 KeyGen
100 Encapsulation
100 Decapsulation

Two features
- Trigger duration [ms]
- Minimum intra-window current [mA]

Evaluation
Stratified 10-fold cross-validation

Key Takeaways

High accuracy with simple features
98.67% (LR) and 99.67% (RF) using only two scalar features.

Clear class separability
Decapsulation forms a distinct cluster, while key generation shows a bimodal distribution.

Statistically significant differences
Welch's t-test and Mann–Whitney U test: $p < 0.001$ for all class pairs.

The three ML-KEM operations (KeyGen, Encapsulation, Decapsulation) are clearly separable in the feature space,
allowing for high classification accuracy with lightweight models (up to 99.67% with Random Forest).

Valid vs. Corrupted Ciphertext PoC

Distinguishing valid and invalid decapsulation using simple power features

Feature-Space Separation

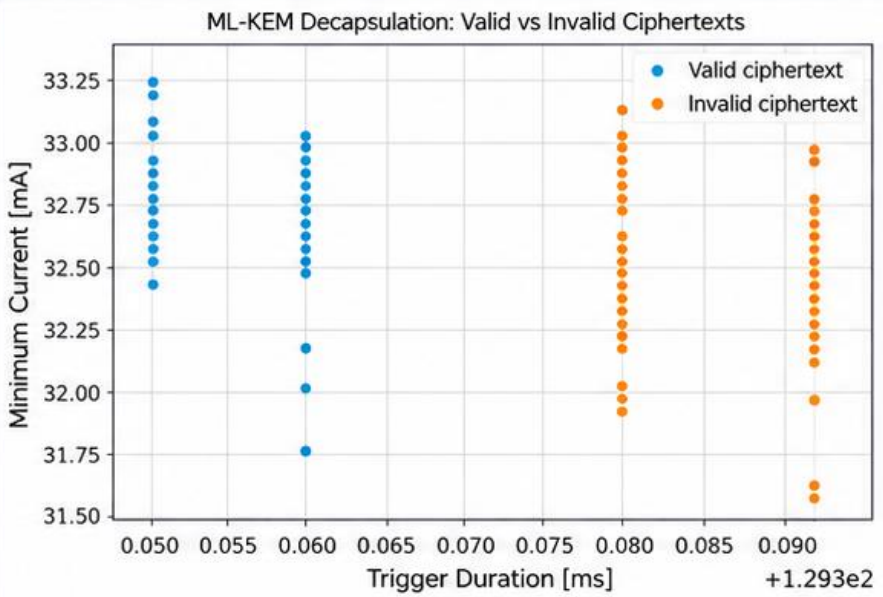


Fig. 6. Feature-space comparison of valid and invalid ML-KEM decapsulation traces using trigger duration and minimum intra-window current.

Example Invalid Decapsulation Trace (ML-KEM) – Different Measurement Windows

Window before return to idle



Window during execution



Both screenshots show the same invalid decapsulation trace with a shifted measurement window.
The different statistics result from selecting a different time region, not from a valid vs. invalid comparison.

Clear class separation
Valid and invalid ciphertexts form distinct clusters in the feature space.

Security-relevant case
Enables detection of decapsulation failures using simple power analysis (bit-flipped ciphertext).

No key recovery
This is a proof-of-concept for detection, not a key recovery attack.

Simple power features enable reliable distinction between valid and corrupted ML-KEM ciphertexts during decapsulation.

AGH University of Kraków
Institute of Telecommunications and Cybersecurity
Daniel Karcz · Marcin Niemiec

Security Implications & Conclusions

Key takeaways from our analysis of ML-KEM side-channel leakage

1. Measurable leakage



- Different ML-KEM operations can be distinguished using only **two simple power features:** trigger duration and minimum intra-window current.
- Lightweight machine learning models achieve **high classification accuracy (up to 99.67%).**

2. Security-relevant behavior



- Valid and corrupted ciphertext decapsulation produce distinguishable power characteristics.
- A single bit flip in the ciphertext leads to measurable differences in the power traces.
- This is a **detection proof-of-concept,** not a key recovery attack.

3. Implementation matters



- Embedded PQC implementations (including ML-KEM) should be **evaluated for side-channel leakage before deployment.**
- Consider countermeasures such as constant-time implementations, power masking, and noise injection.
- Perform evaluations under realistic conditions (hardware, power setup, workloads).

4. Open and extensible



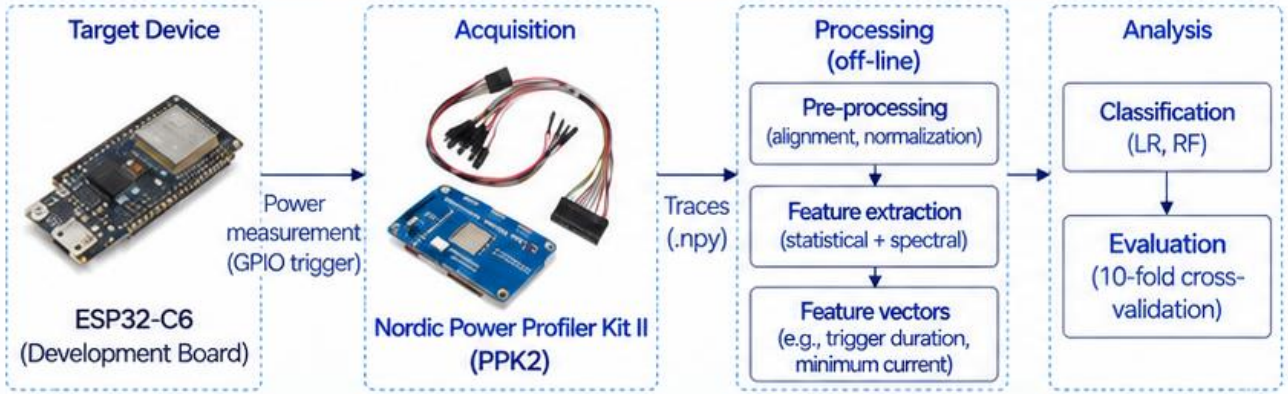
- We provide an **open-source evaluation framework and dataset.**
- While this work focuses on ML-KEM, the framework also supports **ML-DSA and SLH-DSA.**
- We hope it enables further research on the physical security of post-quantum cryptography.

Post-quantum cryptography must be secure not only against quantum attacks, but also against classical side-channel attacks.

Framework / Reproducibility

Open-source evaluation framework for side-channel analysis of PQC implementations

Framework Overview



Supported PQC Algorithms

ML-KEM
(Kyber, NIST FIPS 203)
- Key generation
- Encapsulation
- Decapsulation

Evaluated in the paper (ML-KEM-512)

ML-DSA
(Dilithium, NIST FIPS 204)
- Key generation
- Signing
- Verification

Supported in the framework

SLH-DSA
(SPHINCS+, NIST FIPS 205)
- Key generation
- Signing
- Verification

Supported in the framework

The framework provides a unified pipeline for side-channel analysis of multiple PQC algorithms.
This paper presents experimental results for ML-KEM-512, while support for ML-DSA and SLH-DSA is also included.

Reproducibility

Open-source code
Full measurement and analysis framework available on GitHub.

Dataset
Power traces and pre-processed features released for reproducibility of results.

Documentation
Instructions, environment setup and experiment configuration included in the repository.

Repositories

github.com/dk379x/pqc-embedded-evaluation
- Evaluation framework (Python)
- Support for ML-KEM, ML-DSA and SLH-DSA
- Documentation and setup instructions

github.com/dk379x/pqc-power-analysis
- Measurement scripts and analysis tools
- Datasets and pre-processed features
- Reproducible evaluation pipeline

Open-source framework and datasets enable reproducible research and further investigation of side-channel leakage in PQC implementations.

AGH University of Kraków
Institute of Telecommunications and Cybersecurity
Daniel Karcz · Marcin Niemiec

AGH
AGH University
of Krakow

PQ-NEXT

MIST
WORKSHOP ON MOBILE AND IOT SECURITY TECHNOLOGIES

ESORICS

Thank you!

Questions?

Daniel Karcz · Marcin Niemiec

AGH University of Kraków
Institute of Telecommunications and Cybersecurity

danielkarcz@agh.edu.pl

