OpenSSL CONFERENCE

Testing PQC Timing Side-Channels

Alicja Kario

§ PRIVACY AS A FUNDAMENTAL HUMAN RIGHT

10/09/2026

QARC

JOINT POST-QUANTUM INITIATIVE

Funded by the European Union

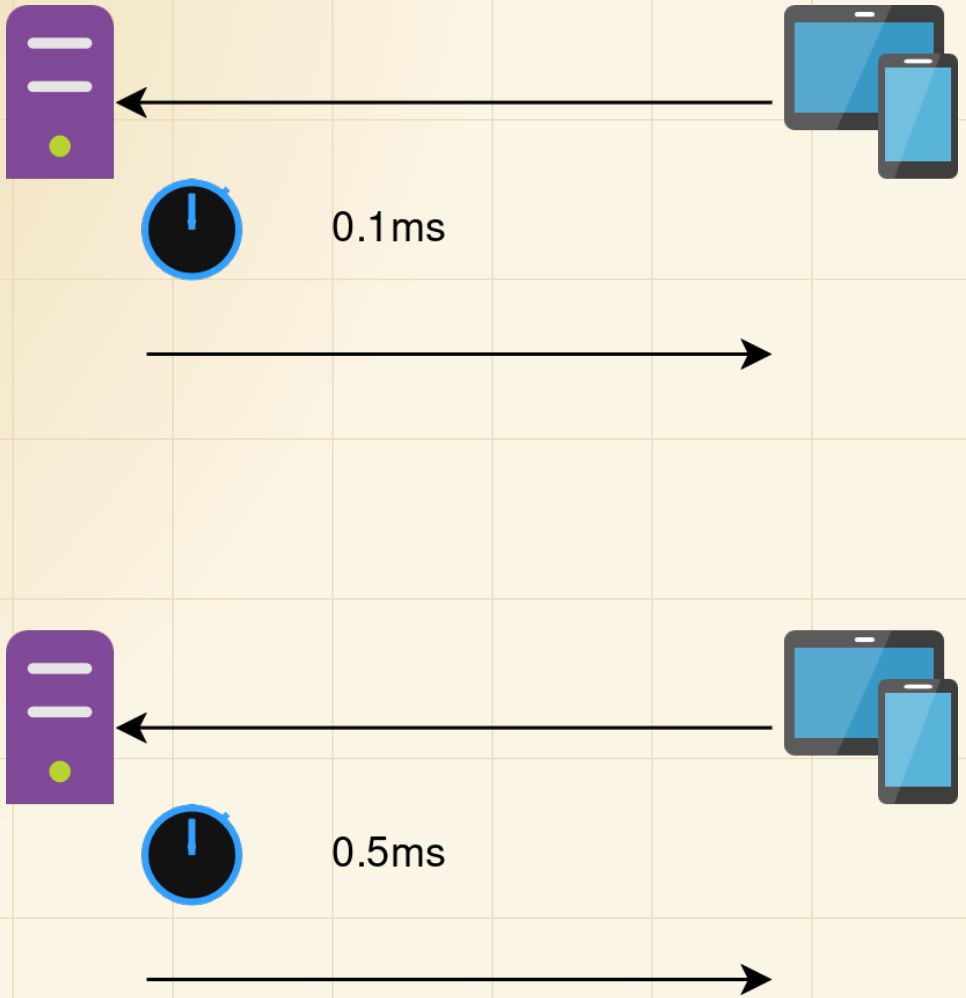
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

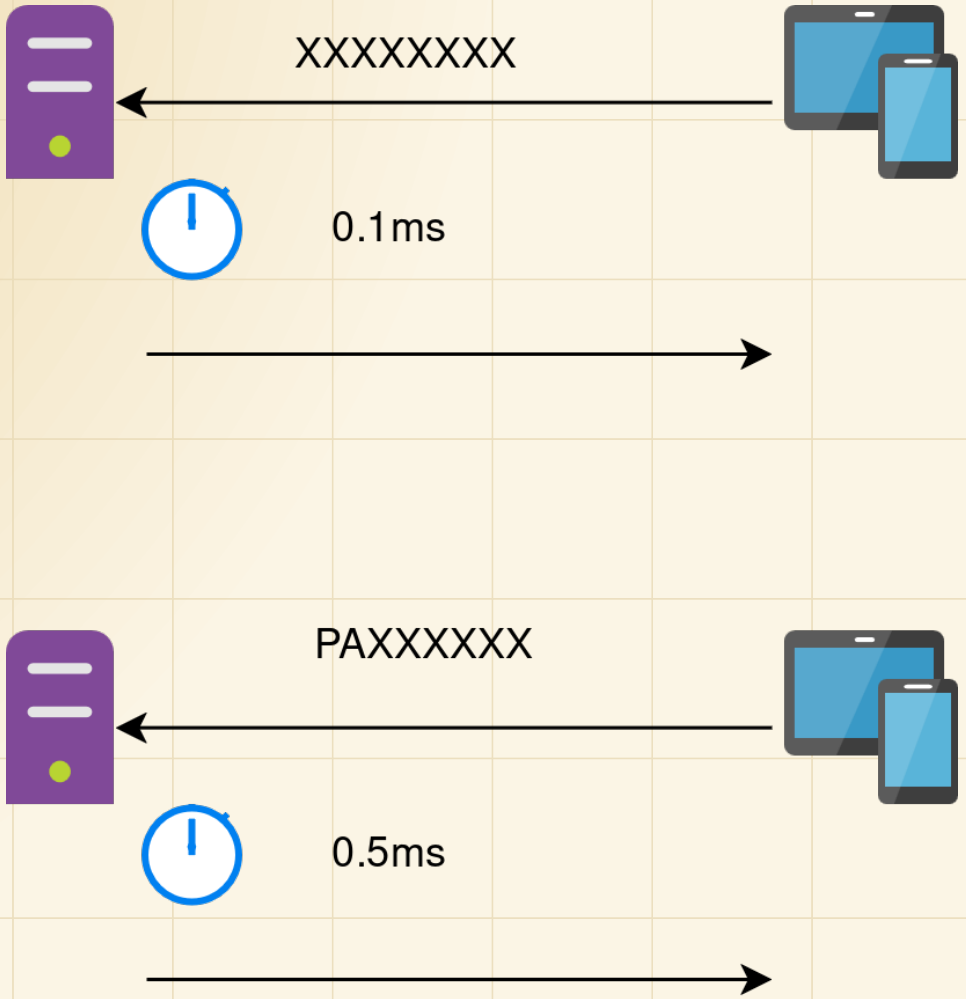
The project funded under Grant Agreement No. 101225691 is supported by the European Cybersecurity Competence Centre.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the granting authority can be held responsible for them.

Why to consider side-channels

- Algorithms considered unbreakable now (AES-256, ML-KEM, ...) are only unbreakable *theoretically*
- Low barrier to entry
- Compromise has high impact (possible long-term key extraction)


0.1ms
0.5ms


XXXXXXXX
0.1ms
PAXXXXXX
0.5ms

Design of experiments

While performing an experiment in principle is simple, there may be some non obvious interactions.

10/09/2026

Design of experiments

While performing an experiment in principle is simple, there may be some non obvious interactions.

10/09/2026

- Take a test tube
- Perform some reaction in it
- Pour the contents out
- Perform another reaction in it

Sources of noise

Cache effects

Depending what data is in CPU cache (L1, L2, ...) the executed code will take different amounts of time: cache misses mean RAM round-trip.

Branch predictor

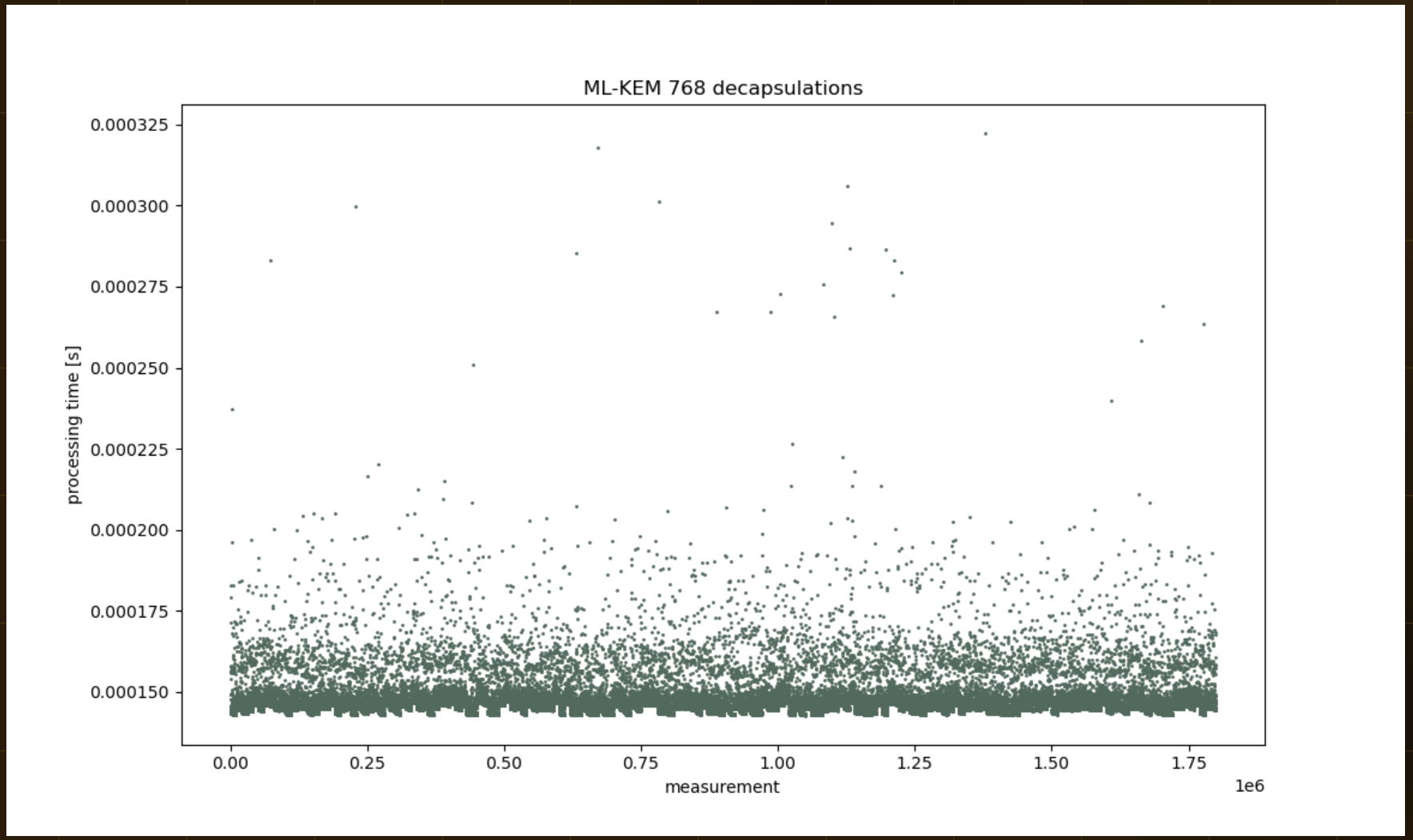
To execute code faster, the CPU “guesses”, based on previous executions, which way a particular check will go and executes code ahead of time.

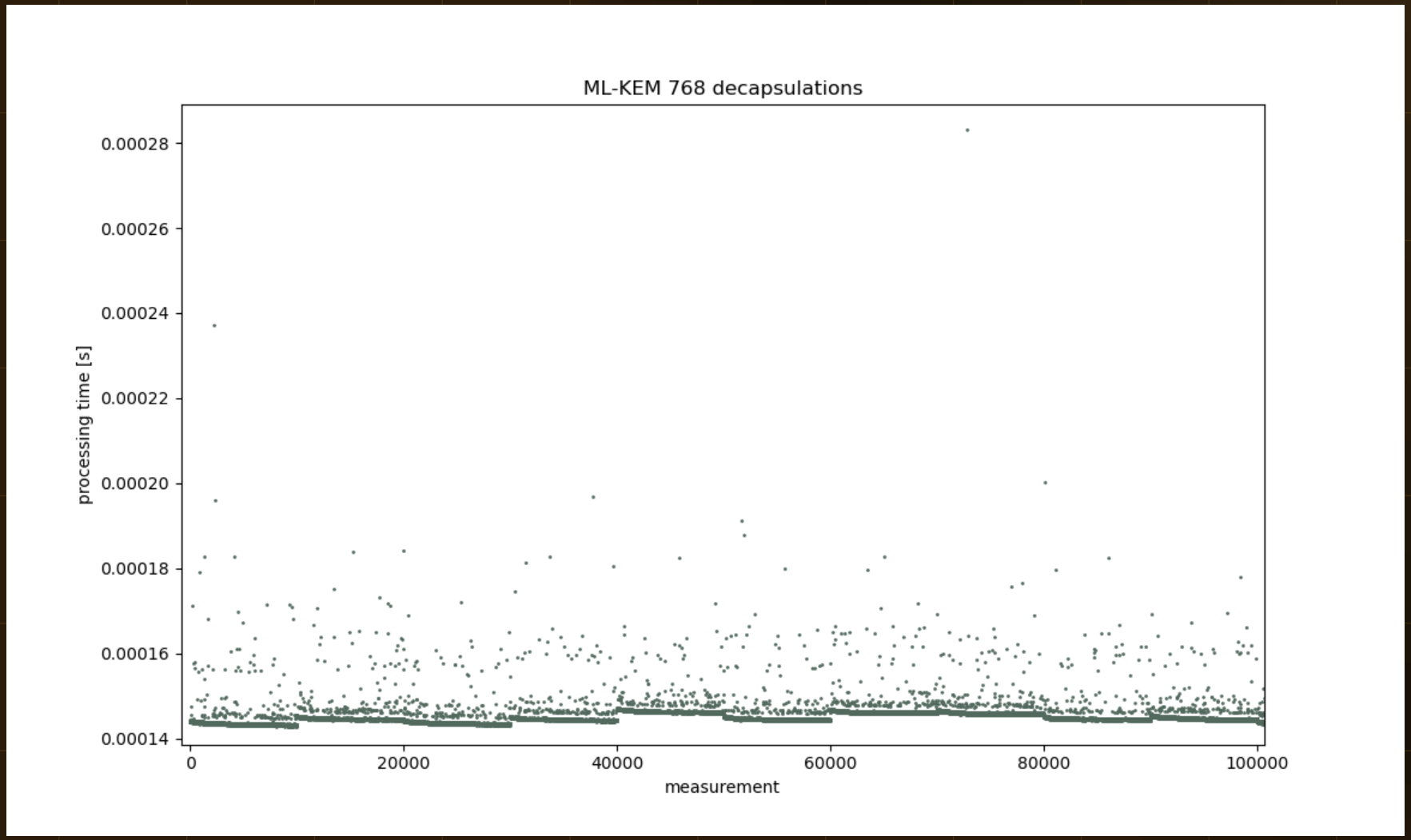
Temperature

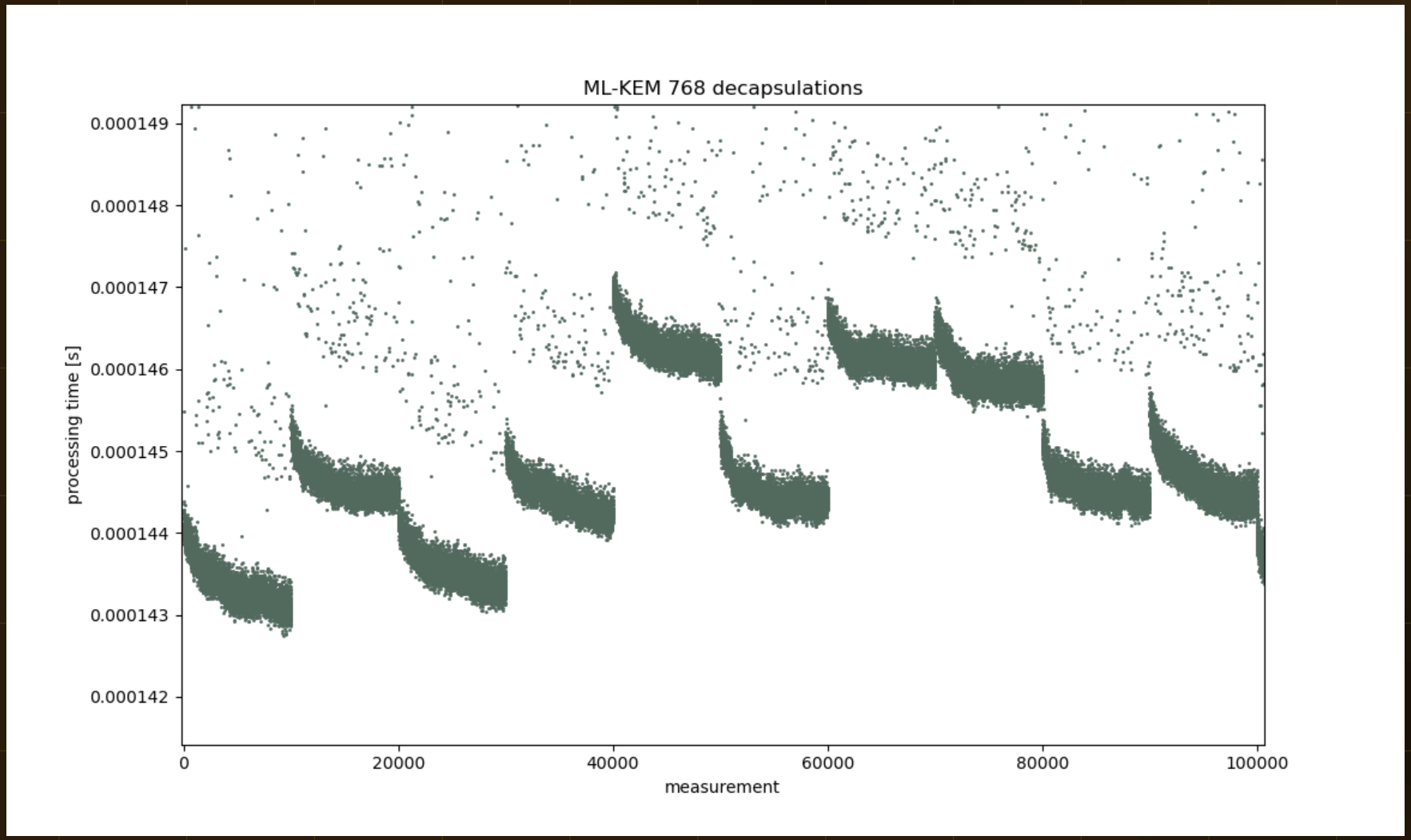
All silicon-based electronics use more power when they are hotter. With frequency scaling that is very pronounced.

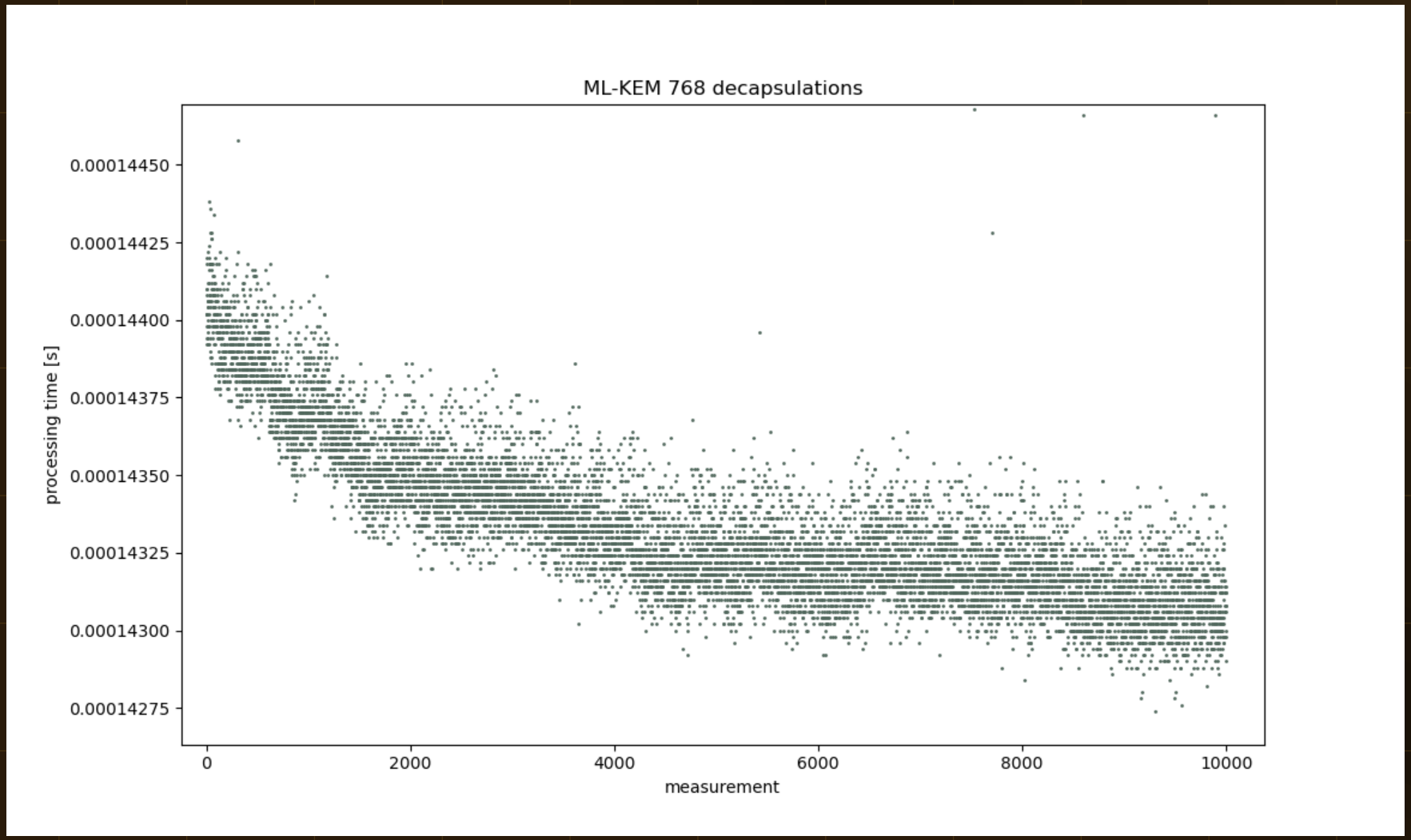
OS state

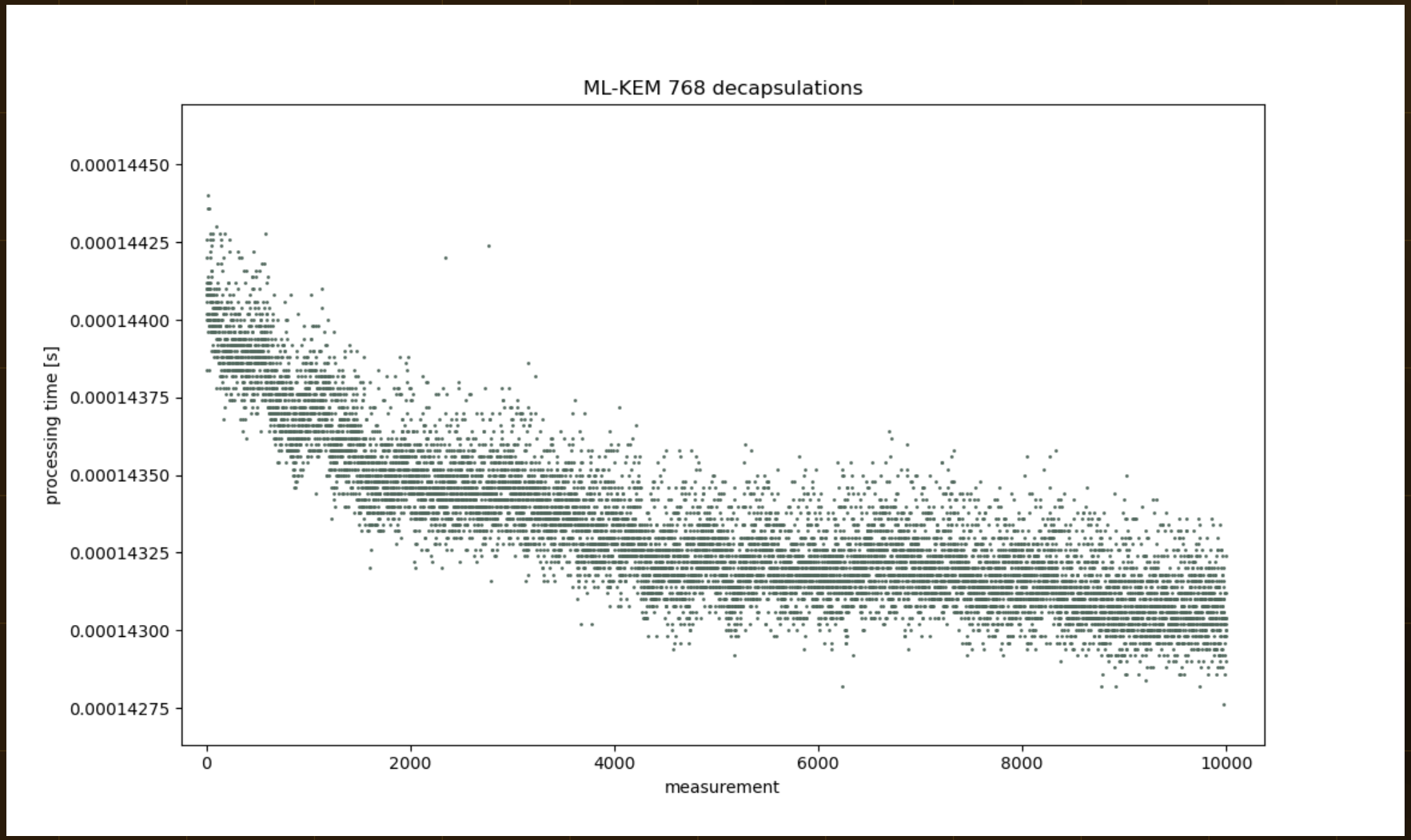
Selected ASLR seed (memory location), other tasks that get scheduled on the same die, IRQ handlers, etc.

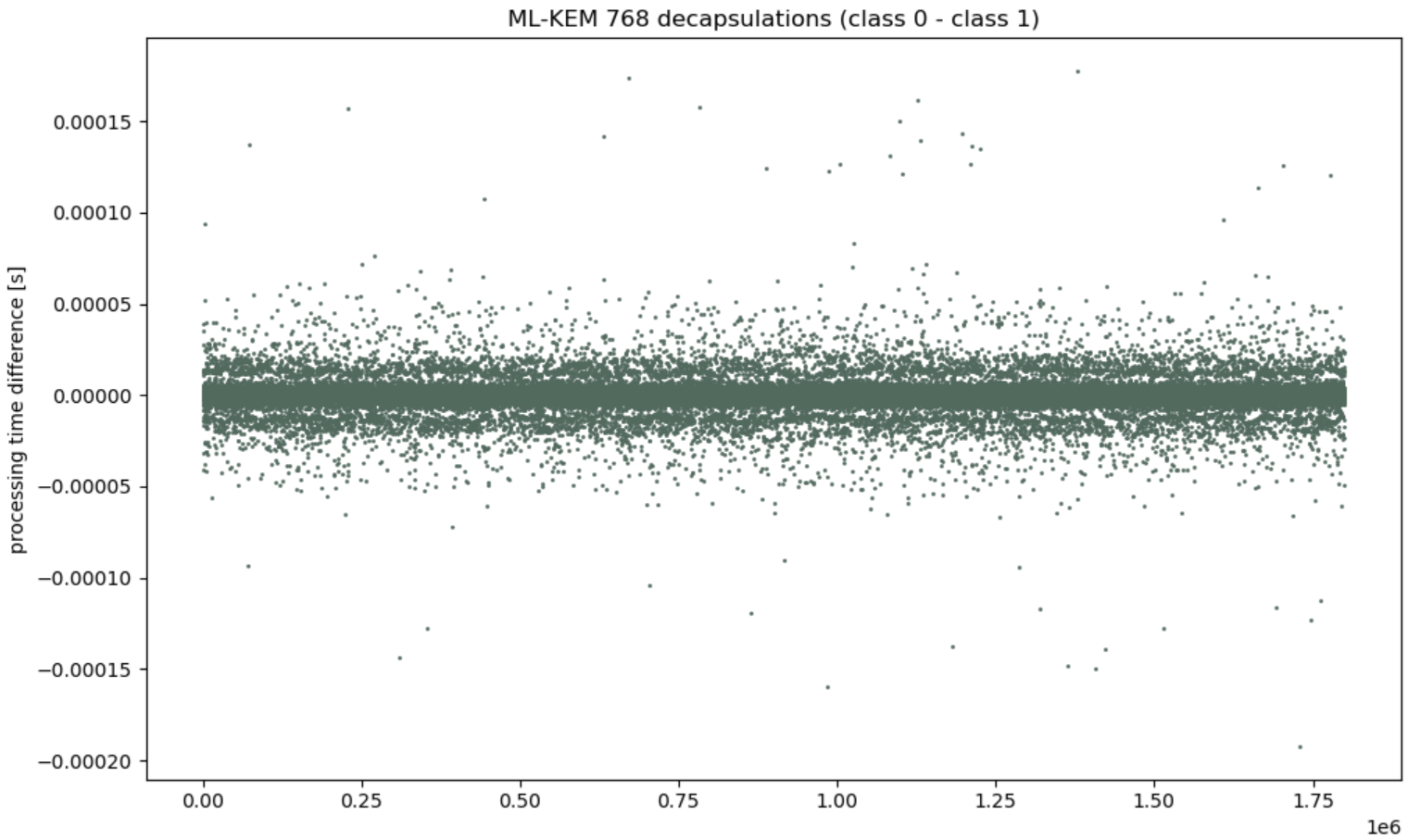

ML-KEM 768 decapsulations
processing time [s]
0.000325
0.000300
0.000275
0.000250
0.000225
0.000200
0.000175
0.000150
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
measurement
1e6

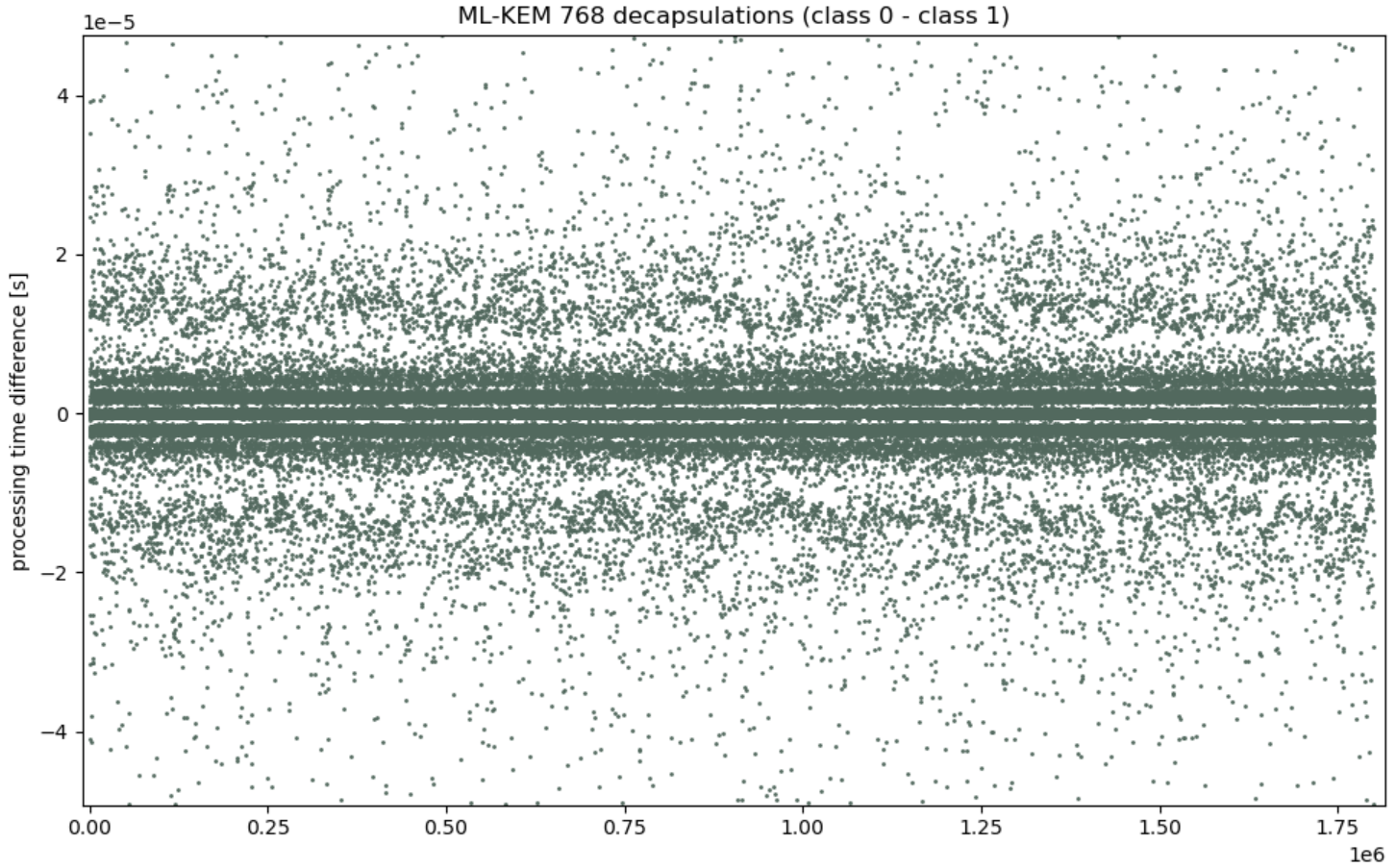

ML-KEM 768 decapsulations
processing time [s]
0.00028
0.00026
0.00024
0.00022
0.00020
0.00018
0.00016
0.00014
0
20000
40000
60000
80000
100000
measurement

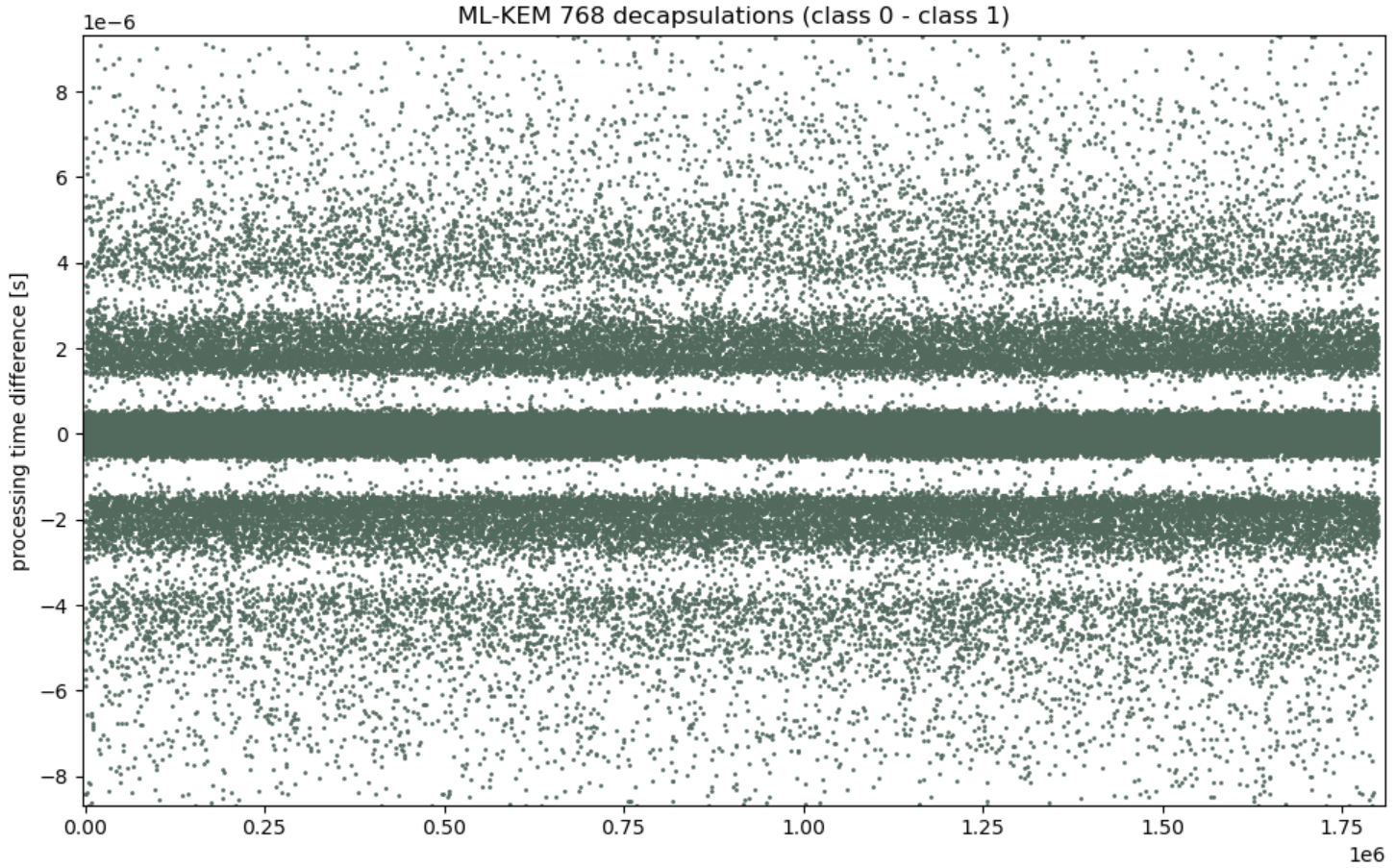

ML-KEM 768 decapsulations
processing time [s]
0.000149
0.000148
0.000147
0.000146
0.000145
0.000144
0.000143
0.000142
0
20000
40000
60000
80000
100000
measurement


ML-KEM 768 decapsulations
processing time [s]
0.00014450
0.00014425
0.00014400
0.00014375
0.00014350
0.00014325
0.00014300
0.00014275
0
2000
4000
6000
8000
10000
measurement

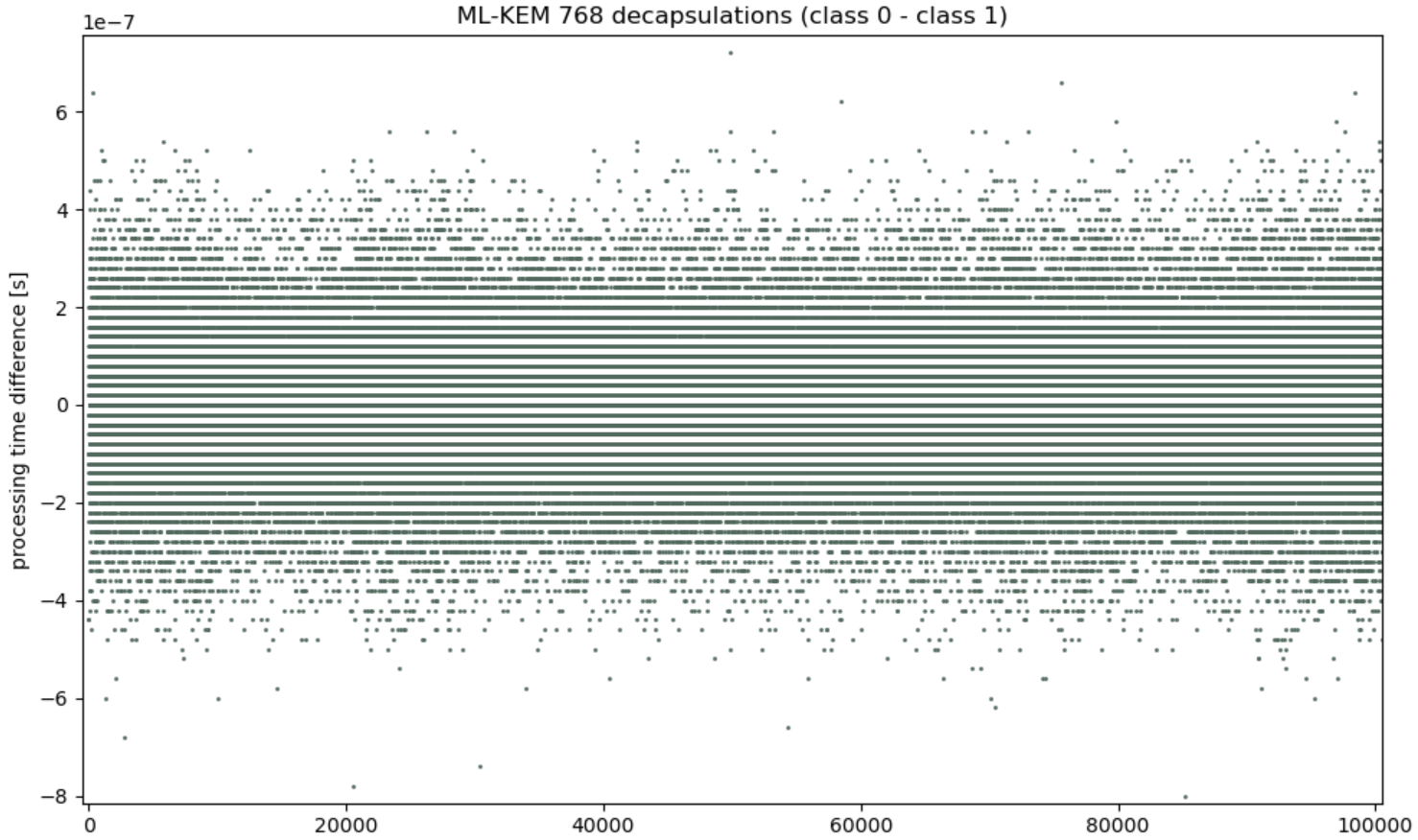

ML-KEM 768 decapsulations
processing time [s]
0.00014450
0.00014425
0.00014400
0.00014375
0.00014350
0.00014325
0.00014300
0.00014275
0
2000
4000
6000
8000
10000
measurement

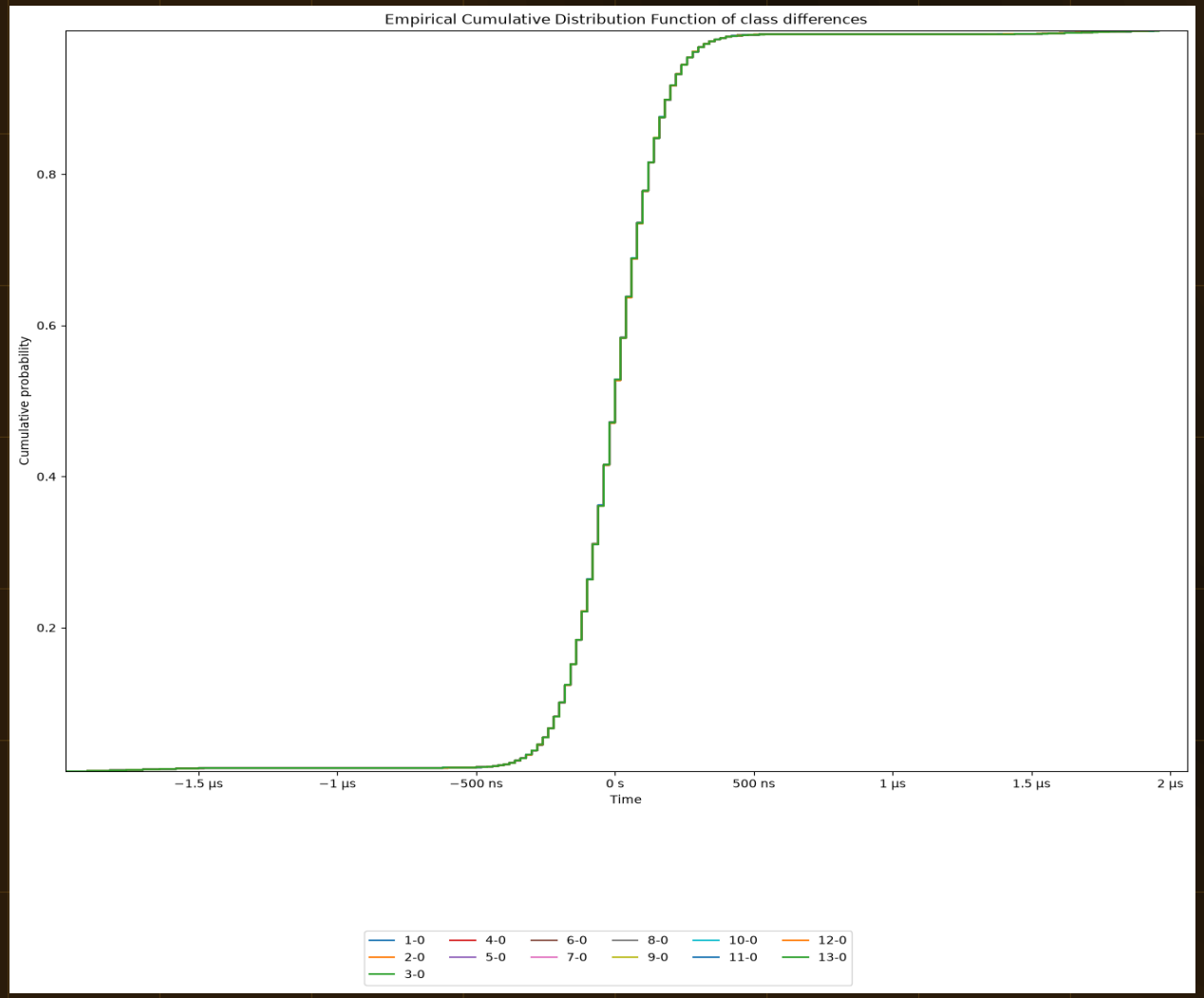

ML-KEM 768 decapsulations (class 0 - class 1)
processing time difference [s]
0.00015
0.00010
0.00005
0.00000
−0.00005
−0.00010
−0.00015
−0.00020
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6


ML-KEM 768 decapsulations (class 0 - class 1)
1e−5
processing time difference [s]
4
2
0
−2
−4
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6

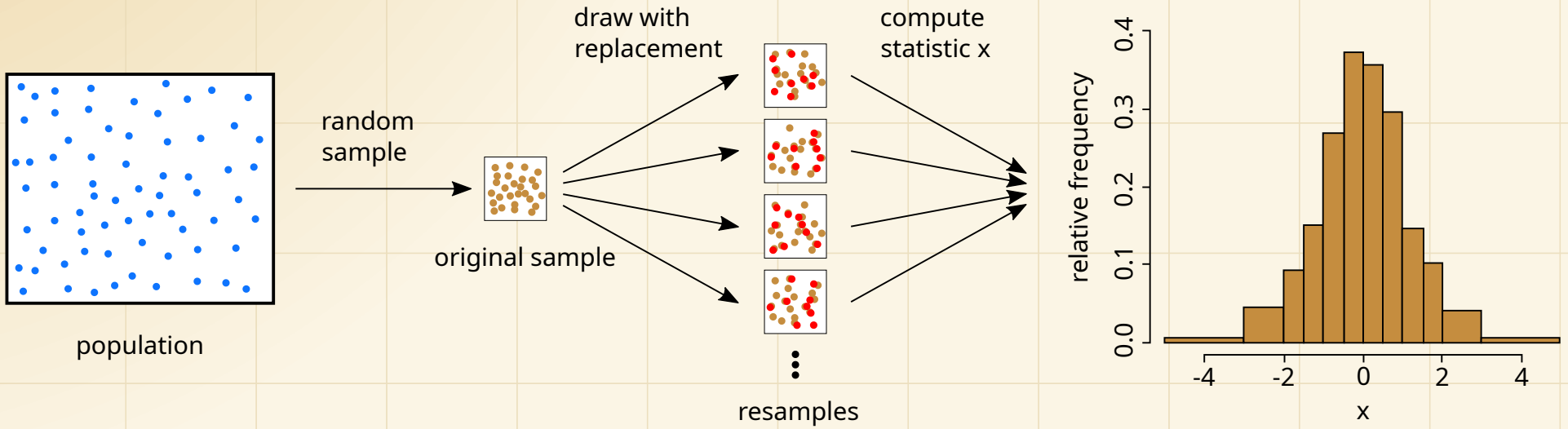

ML-KEM 768 decapsulations (class 0 - class 1)
1e−6
processing time difference [s]
8
6
4
2
0
−2
−4
−6
−8
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6

ML-KEM 768 decapsulations (class 0 - class 1)
1e−6
processing time difference [s]
8
6
4
2
0
−2
−4
−6
−8
0
20000
40000
60000
80000
100000


ML-KEM 768 decapsulations (class 0 - class 1)
1e−7
processing time difference [s]
6
4
2
0
−2
−4
−6
−8
0
20000
40000
60000
80000
100000


Empirical Cumulative Distribution Function of class differences
Cumulative probability
0.8
0.6
0.4
0.2
−1.5 µs
−1 µs
−500 ns
0 s
500 ns
1 µs
1.5 µs
2 µs
Time
1-0
2-0
3-0
4-0
5-0
6-0
7-0
8-0
9-0
10-0
11-0
12-0
13-0

Bootstrapping



Proper statistical approach

- Use test harness that is oblivious to test vector order
- Randomize the test vector execution order
- Use blocking (compare the measurements only within a group)
- Use non-parametric methods
- Estimate precision of the measurement

Proper statistical approach

- *Paired* Student t-test
- Sign test, Wilcoxon signed-rank test
- Friedman test
- *Repeated Measures* ANOVA on ranks
- Skillings-Mack test

ML-KEM Testing

Test vectors for decapsulation

10/09/2026

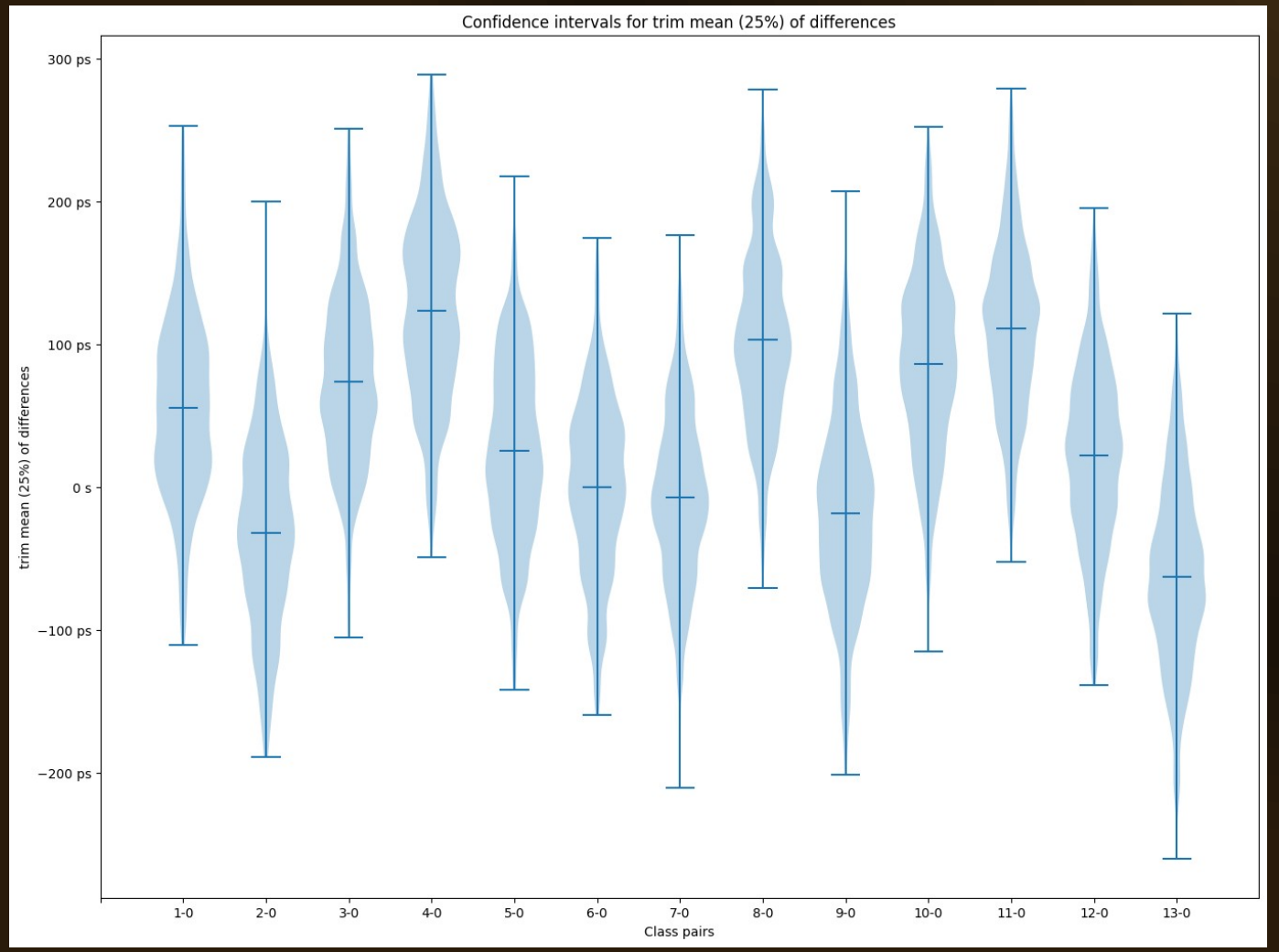
- Correct (valid) random messages
- Random bytes (invalid)
- Modified one whole *u* or *v* polynomial
- Modified one specific coefficient in one of the *u* or *v* polynomials

Overall: 14 classes with 3 duplicates (for sanity checks)

OpenSSL CONFERENCE

ML-KEM Results

OpenSSL 3.5.5 on x86_64
ML-KEM-768



10/09/2026

OpenSSL 3.5.5 on x86_64 ML-KEM-768

OpenSSL CONFERENCE

tlsfuzzer analyse.py version 9 analysis

Sign test mean p-value: 0.4412, median p-value: 0.445, min p-value: 0.000378

Friedman test (chisquare approximation) for all samples

p-value: 0.06185529844390088 4

Worst pair: 2(one_u_remain_2), 10(xor_u_coefficient_0_1)

Mean of differences: 1.46978e-10s, 95% CI: -7.22323e-13s, 2.959107e-10s (±1.483e-10s)

Median of differences: 2.92600e-10s, 95% CI: 0.00000e+00s, 5.852000e-10s (±2.926e-10s)

Trimmed mean (5%) of differences: 1.28179e-10s, 95% CI: 5.59161e-12s, 2.424555e-10s (±1.184e-10s)

Trimmed mean (25%) of differences: 2.00841e-10s, 95% CI: 7.54539e-11s, 3.206342e-10s (±1.226e-10s)

Trimmed mean (45%) of differences: 2.45975e-10s, 95% CI: 1.05779e-10s, 3.801618e-10s (±1.372e-10s)

Trimean of differences: 1.46302e-10s, 95% CI: -2.50000e-15s, 3.657500e-10s (±1.829e-10s)

Layperson explanation: Implementation most likely not providing a timing side-channel signal

ML-KEM Results

OpenSSL 3.5.5

10/09/2026

- Tested ML-KEM-512, ML-KEM-768, ML-KEM-1024
- On: x86_64, aarch64, ppc64le, s390x
- All 95% confidence intervals <1 ns
- All negative

ML-KEM Results

NSS 3.112.0

- Tested ML-KEM-768, ML-KEM-1024 (no ML-KEM-512)
- On: x86_64, aarch64, ppc64le, s390x
- All 95% confidence intervals <1 ns
- All negative

10/09/2026

Materials

Base library/toolkit: https://github.com/tlsfuzzer/tlsfuzzer

Toy example: https://securitypitfalls.wordpress.com/2023/10/16/experiment-with-side-channel-attacks-yourself/

Toolkit for RSA: https://github.com/tomato42/marvin-toolkit/

Toolkit for ECDSA: https://github.com/GeorgePantelakis/minerva-toolkit/

Toolkit for arbitrary precision arithmetic: https://github.com/tomato42/ctmpi

Toolkit for ML-KEM: https://github.com/tlsfuzzer/mlkem-sct-toolkit

Toolkit for ML-DA: https://github.com/rvala02/mldsa-toolkit (WIP)

Methodology: https://eprint.iacr.org/2026/2023

My contact: hkario@redhat.com