THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY: CEDA IEEE Council on Electronic Design Automation, SIGDA ACM

A Remote Side-Channel Attack on Post-Quantum Key Encapsulation Mechanisms

A Case Study of ML-KEM on Intel x86

Mona Hashemi[1] · Qianmei Wu[2,3] · Fan Zhang[2] · **Shivam Bhasin**[3,4] · Trevor E. Carlson[1]

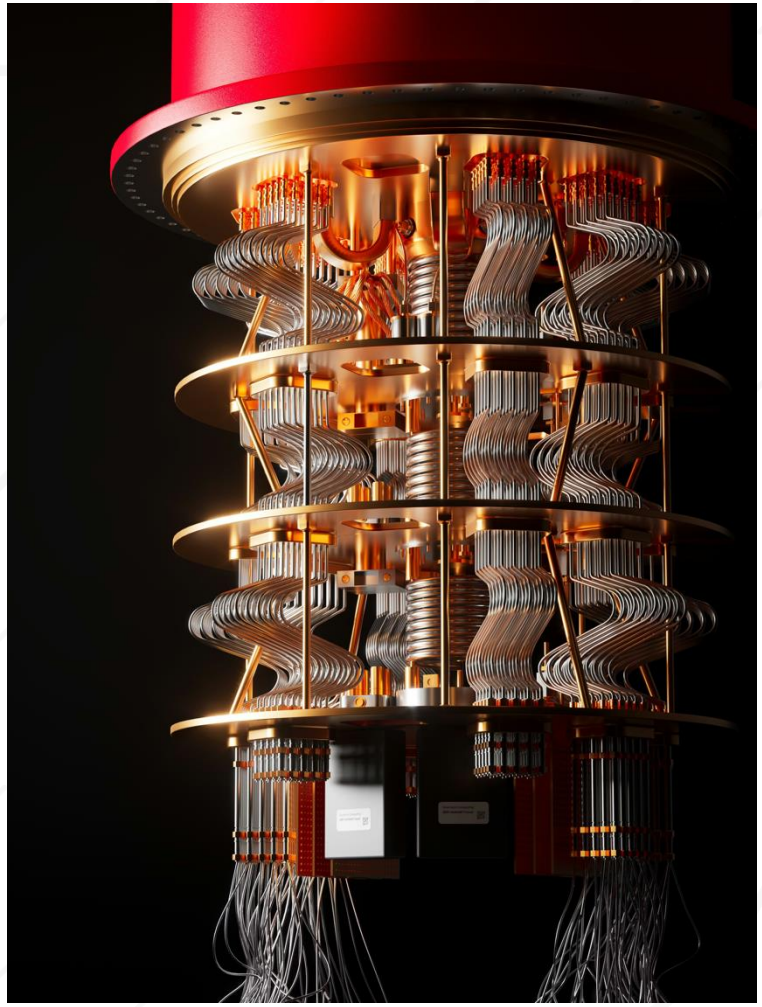
[1]National University of Singapore · [2]Zhejiang University · [3]Nanyang Technological University. [4]PQStation

ZHEJIANG UNIVERSITY · NUS National University of Singapore · NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE · PQSTATION

Motivation

- **Quantum threat is real** — NIST has standardized ML-KEM (FIPS 203) as the primary PQC KEM
- **ML-KEM is already deployed** — TLS, cloud, OS-level crypto on modern x86
- **Problem** — All SCA studies are on embedded targets (AVR, ARM Cortex-M). No attack on x86. No remote attack.

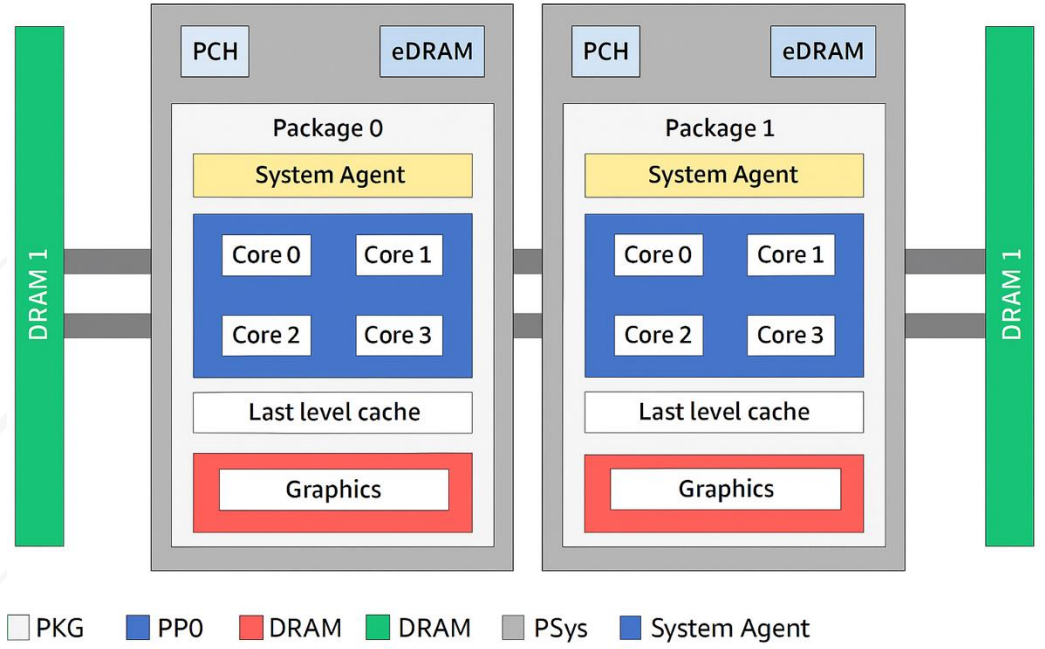
This work — First remote power SCA on full ML-KEM on Intel x86 (99.5% key recovery success rate)



https://unsplash.com/s/photos/quantum-computer

Running Average Power Limit (RAPL)

- **Intel RAPL** — Software-accessible energy telemetry, readable without physical access
- **3 Linux interfaces** — MSR driver, perf_event_open(), PowerCap sysfs
- **Why it matters** — Accessible in cloud containers and multi-tenant deployments; exploitable in some AWS EC2 types (CVE-2020-8694/8695)



https://hubblo-org.github.io/scaphandre-documentation/explanations/rapl-domains.html

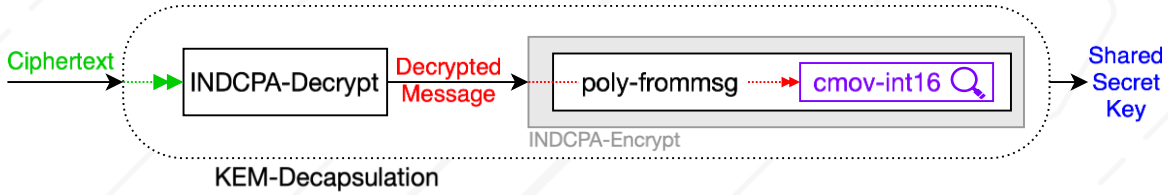
Threat Model

- **Adversary:** No physical access; remotely submits chosen ciphertexts to ML-KEM decapsulation oracle

- **Capabilities:**
  1. Trigger repeated decapsulations,
  2. Read RAPL energy counters,
  3. Perform offline statistical analysis

- **No specialized hardware required:** Attack relies solely on RAPL software interface at default 1 ms resolution

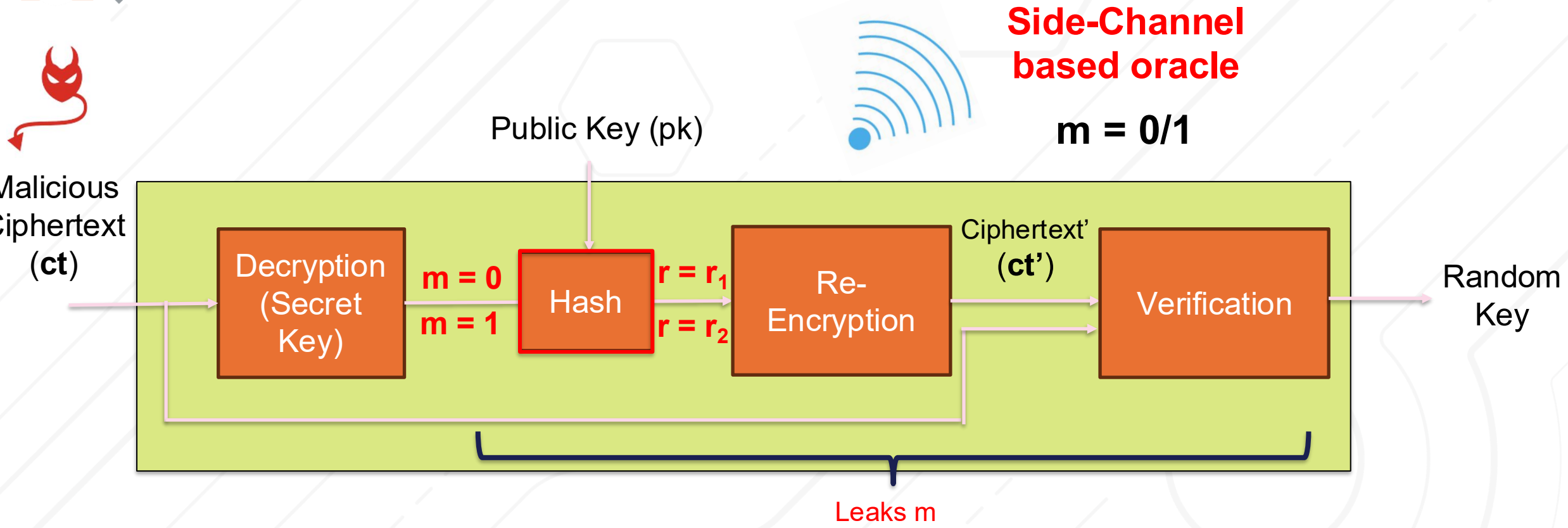
Target Leakage




```
void PQCLEAN_MLKEM512_CLEAN_cmov_int16(int16_t *r, int16_t v, uint16_t b)
{
    b = -b;
    *r ^= b & ((*r) ^ v);
}
```

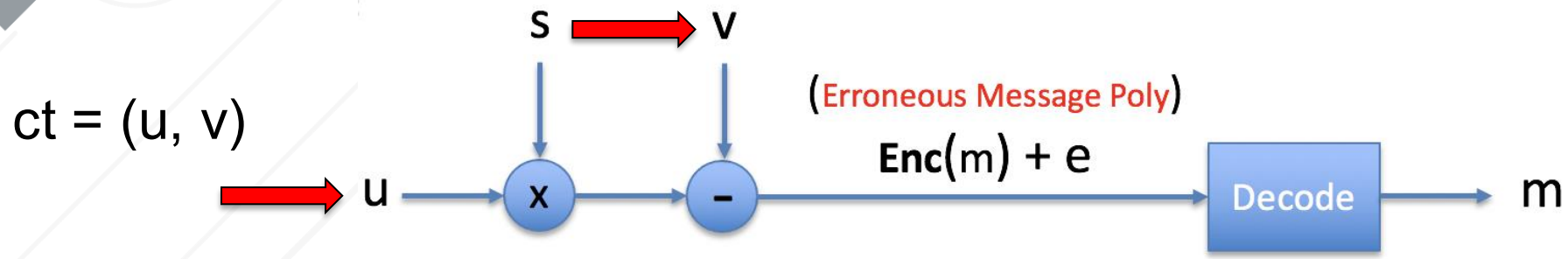
- **Attack path:** Decapsulation → Re-Encryption → poly-frommsg → cmov-int16
- **cmov(r, v, b):** Constant-time conditional copy in PQClean; uses two's-complement negation: **b = -b**
  - b=0 → HW=0 (low power);
  - b=1 → 0xFFFF → HW=16 (high power)
- **Amplification:** Sleep-induced spikes bracket target; act as temporal anchors and amplify residual leakage in RAPL traces

Side-Channel Assisted PC Oracle



- **Why is it easy to distinguish**:
  - Use of Hash Functions: **H(0,pk)** and **H(1,pk)** ➔ **Leakage Amplification**
  - **Thousands of Leakage Points**

Constructing Malicious Ciphertext

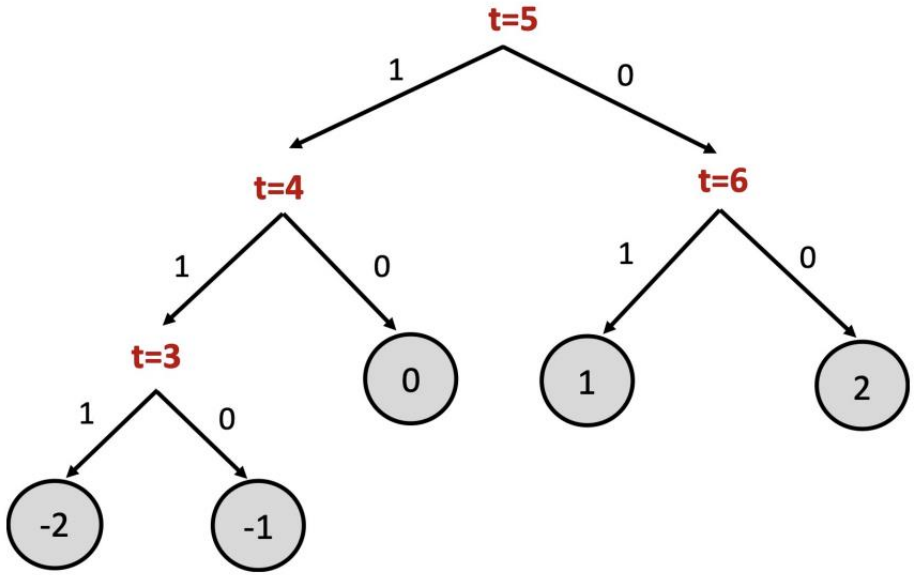


Chosen u	k	0	0	0	0	0	0
u.s	$k.s_0$	$k.s_1$	$k.s_2$	$k.s_3$	$k.s_4$	$k.s_5$	$k.s_6$
Chosen v	t	0	0	0	0	0	0
m' = u.s - v	$k.s_0$-t	$k.s_1$	$k.s_2$	$k.s_3$	$k.s_4$	$k.s_5$	$k.s_6$
m = Decode(**m**)	$f(s_0)$	0	0	0	0	0	0
	m_0	m_1	m_2	m_3	m_4	m_5	m_6

Constructing Malicious Ciphertext

m'=Decode(**m'**)

$f(s_0)$	0	0	0	0	0	0



Binary Distinguisher for secret coeffs. of MLKEM-512 coeffs in [-3,3]

ct = (k, t) k = 208

t	Oracle Resp.
5	1
4	1
3	0

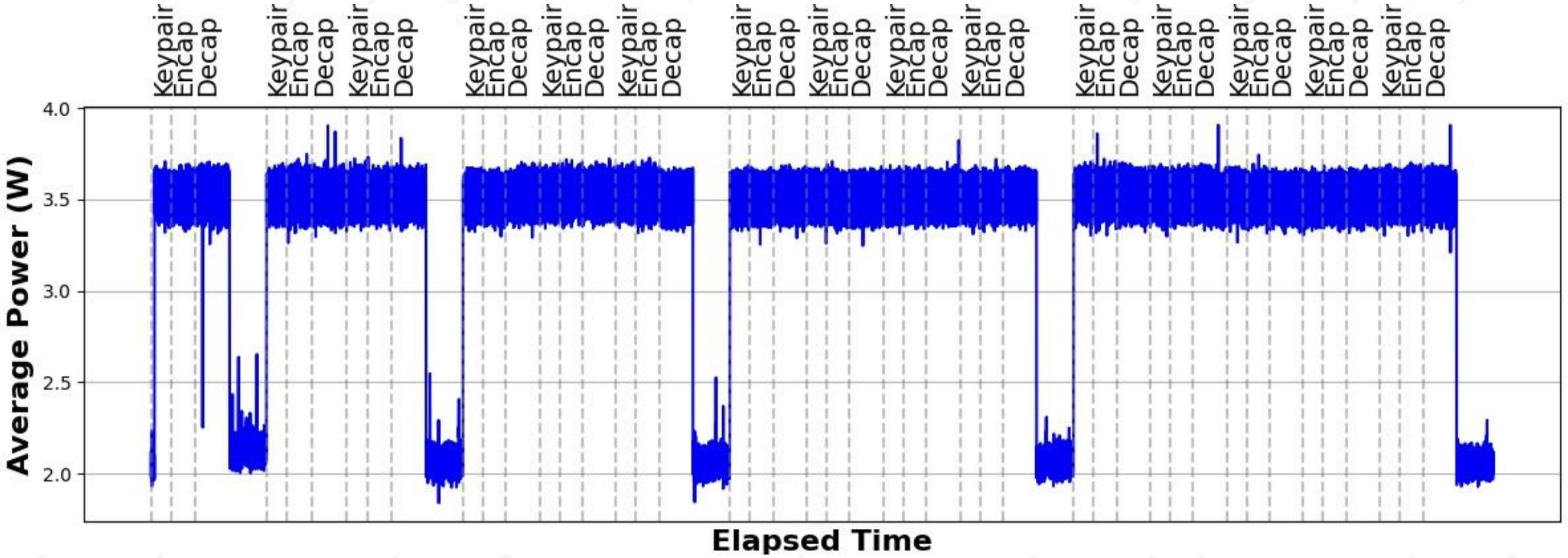
→ $s_0 = -1$

Recover 1 secret coeff. in not more than 3 queries

Experimental Setup

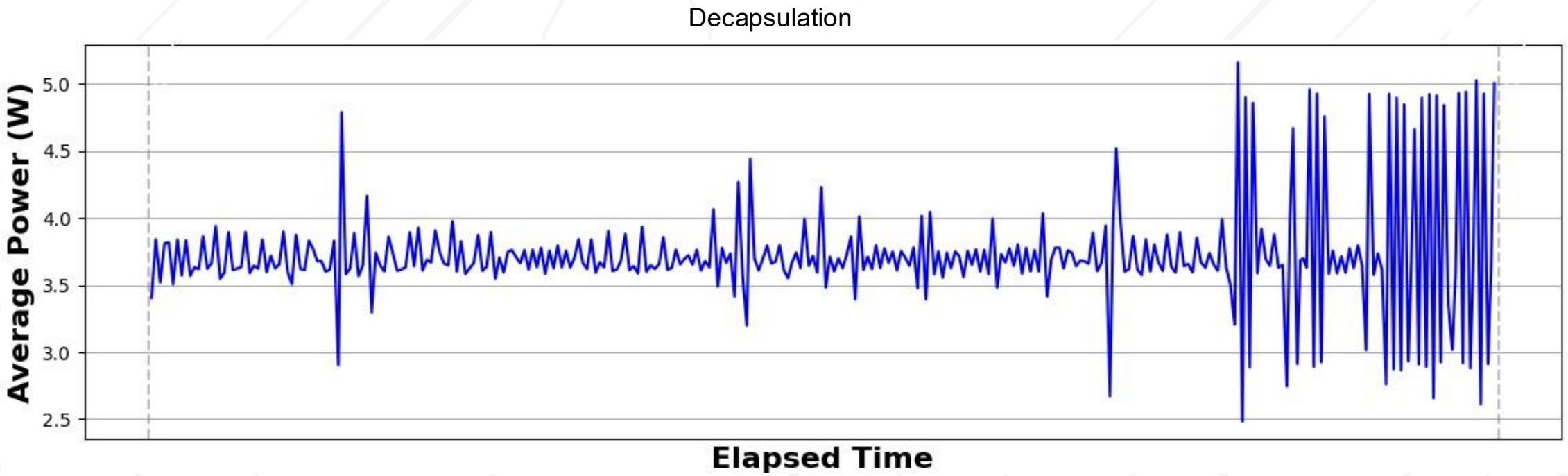
- **Target Platform** — Intel Core i7-12700H (Alder Lake); Ubuntu 22.04.2, kernel 5.19.0-35
- **Cryptographic Target** — **ML-KEM-512 (FIPS 203, k=2, CBD η=3);** PQClean clean implementation; GCC 11.3 -O3
- **RAPL Measurement** — MSR driver; PP0-ENERGY-STATUS (0x639); energy unit = 61.04 μJ; ~1ms sampling

Localizing Decapsulation Leakage



Periodic Keypair/Encap/Decap phases visible

Localizing Decapsulation Leakage



Visibly structured power profile despite noise from speculative execution and OoO, targeting cmov-int16.

Stage 1: Message Recovery via RAPL

- **Dataset:** 4 datasets × 1 million RAPL power traces per dataset; mean-variance normalized
- **Classifier:** SVM with RBF kernel (γ=0.1, C=100) for binary b=0/1 classification of decrypted message bits
- **Results:** Message-bit recovery rate **97% – 99.83%** across all datasets
- **Key insight:** Coarse-grained RAPL (1 ms) retains sufficient information to distinguish secret-dependent cmov executions despite speculative execution noise

Stage 2: Secret Key Recovery

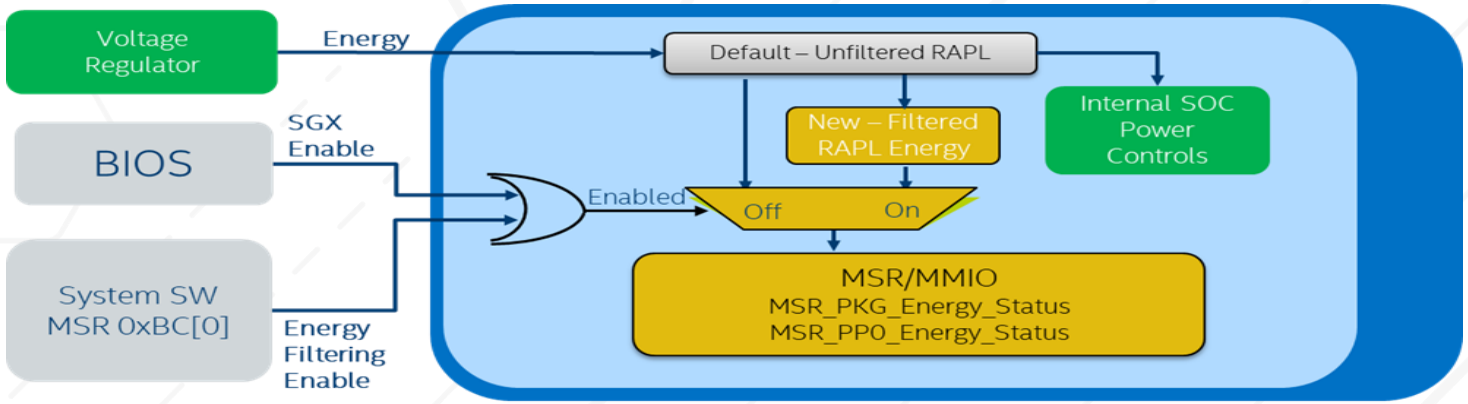
Dataset	Message Recovery Success Rate (%)	No of Attack Traces for Secret Key Recovery	Secret Key Recovery Success Rate (%)
1	97-98.4	1k	94.98
2	97.78	1k	90.25
3	98.18	1k	95.01
3	98.18	10k	95.02
4	99.83	1k	98.90
4	99.83	5k	99.50

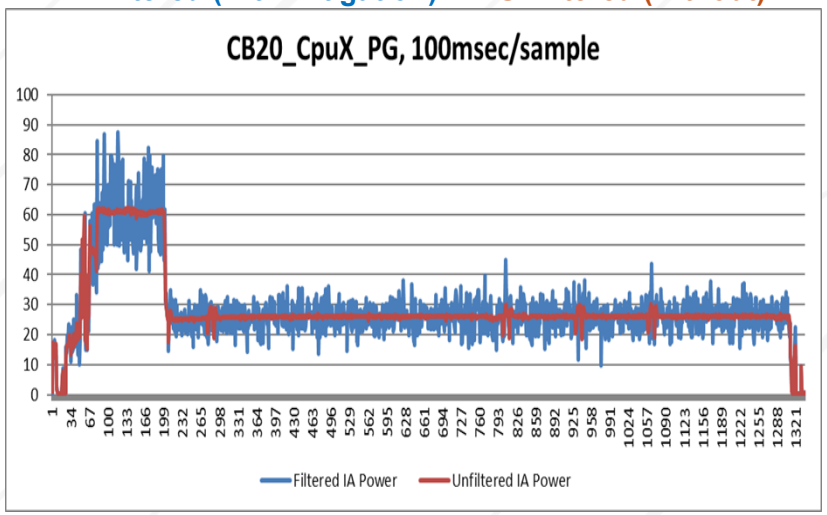
Conclusions & Takeaways

- **First remote power SCA on full FIPS 203 (ML-KEM) on Intel x86** using Intel RAPL — no physical access required
- **Message-bit recovery: up to 99.83%**
- **Secret key recovery: 94.98% – 99.5%**
  - ❖ This can be further improved using theoretical lattice reduction
- **Key insights:**
  - ❖ Coarse-grained (1 ms) RAPL leakage survives speculative execution, OoO, and DVFS noise on modern x86
  - ❖ Constant-time code ≠ No Leakage (Microarchitectural power behavior leaks)
  - ❖ Mitigation requires cross-layer defense **(**masking, hardware power obfuscation etc)

Bonus: Targeting RAPL Mitigations

- Introduced against CVE-2020-8694, CVE-2020-8695
- Adds random energy noise and reduces update frequency, lowering data-energy correlation
- Enabled through MSRs






```
SVM Classification Results for Power_Oversampled_PP0 ---
Accuracy: 0.9071

Classification Report:
              precision    recall  f1-score   support

        msg=0       1.00      0.81      0.90      9844
        msg=1       0.84      1.00      0.91      9843

    accuracy                           0.91     19687
   macro avg       0.92      0.91      0.91     19687
weighted avg       0.92      0.91      0.91     19687


Confusion Matrix:
[[8015 1829]
 [   0 9843]]
```

Before mitigation (Acc: 91.1%)

```
--- SVM Classification Results for Power_Oversampled_PP0 ---
Accuracy: 0.9110

Classification Report:
              precision    recall  f1-score   support

        msg=0       1.00      0.82      0.90      9859
        msg=1       0.85      1.00      0.92      9859

    accuracy                           0.91     19718
   macro avg       0.92      0.91      0.91     19718
weighted avg       0.92      0.91      0.91     19718


Confusion Matrix:
[[8105 1754]
 [   0 9859]]
```

After mitigation (Acc: 90.7%)

Mitigation reduces leakage but attack remains feasible