On Certifying PQC Implementations at **“High” Assurance Level**

Markku-Juhani O. Saarinen
<markku-juhani.saarinen@tuni.fi>

EUCA, Brussels, March 27, 2025

Tampereen yliopisto
Tampere University

Hello! I'm Markku-Juhani O. Saarinen 👋

- *(In cryptography since 1990s. First employee at **PQShield Ltd**, Oxford UK in 2018. Architected, tinkered, prototyped, developed, and helped sell **hardware PQC modules**.)*
- RISC-V since 2019. I designed some of the (now-standard) crypto instructions.
- Returned to **Finland** in 2023-24. **Professor of Practice** at **Tampere University**.
- Chair, **RISC-V International** Post-Quantum Cryptography Task Group (**RVI PQC TG**).
- Finnish representative to CLC TC/47X (Secure Chips Standards related to EU CRA.)
- Program Co-Chair, **PQCrypto 2025** (Taipei, Taiwan April 8-10, 2025): **See you there!**

NIST PQC Standards in effect from August 2024

KEY ESTABLISHMENT

DIGITAL SIGNATURES

Kyber: FIPS 203 ML-KEM (2024)

Primary PQC **key establishment** algorithm to replace Diffie-Hellman (ECDH) key exchange and RSA public-key encryption. Lattice-based.

HQC: ??-KEM (2026?)

Code-based key establishment algorithm, approved from “Round 4” in March 2025.

Hybrid schemes: One still needs to support traditional Elliptic Curve and RSA methods.

Dilithium: FIPS 204 ML-DSA (2024)

Primary "general-purpose" PQC **signature algorithm** to replace ECDSA, RSA signatures. Lattice-based.

XMSS and **LMS**: NIST SP 800-208 (2020)

SPHINCS+: FIPS 205 SLH-DSA (2024)

Hash-based signatures; Firmware signing.

Falcon: FIPS 206 FN-DSA (2025). Lattice-based.

Signature "On-Ramp" Algorithms (2026?)

“It’s the law” (In USA, perhaps in EU soon too)

NSM-8 (Jan 2022): *“On Improving the Cybersecurity of National Security, Department of Defense, and Intelligence Community Systems”*

NSM-10 (May 2022): *“On Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems”*

HR 7535 (Dec 2022): *“Quantum Computing Cybersecurity Preparedness Act”*

<u>These PQC-related National Security Memorandums and the Public Law:</u>

- Mandates transition to Post-Quantum Cryptography in government IT.
- Assigns inventory, reporting responsibilities, sets timelines, etc.
- Outside Government’s own IT systems and some critical sectors, the use of post-quantum cryptography (like most information security) is of course not enforced – mostly just self-regulation and “business best practices” in many industries.

NIST IR 8547 (Current Draft Version)

Table 2: Quantum-vulnerable digital signature algorithms

Digital Signature Algorithm Family	Parameters	Transition
ECDSA [FIPS186]	112 bits of security strength	***Deprecated*** after 2030 ***Disallowed*** after 2035
	≥ 128 bits of security strength	***Disallowed*** after 2035
EdDSA [FIPS186]	≥ 128 bits of security strength	***Disallowed*** after 2035
RSA [FIPS186]	112 bits of security strength	***Deprecated*** after 2030 ***Disallowed*** after 2035
	≥ 128 bits of security strength	***Disallowed*** after 2035

https://doi.org/10.6028/NIST.IR.8547.ipd

UK NCSC in March 2025

By 2028

- Define your migration goals
- Carry out a full discovery exercise (assessing your estate to understand which services and infrastructure that depend on cryptography need to be upgraded to PQC)
- Build an initial plan for migration

By 2031

- Carry out your early, highest-priority PQC migration activities
- Refine your plan so that you have a thorough roadmap for completing migration

By 2035

- Complete migration to PQC of all your systems, services and products

https://www.ncsc.gov.uk/guidance/pqc-migration-timelines

PQC Module Certification **has Started**

- **FIPS 140-3**: Functional testing of FIPS 203, 204, 205 implementations has been active since Aug 2024 and a bunch of modules have been certified.
- **ANSSI**, **BSI** and other EU recommend PQC to be combined with a classical algorithm (RSA or ECDSA). CC can include e.g. side-channel testing.

Random private conversations:

- Most major vendors are upgrading their product lines, *as they have to do*.
- In Jan 2025 BSI said they had not yet received certification requests.
- ANSSI has started working on procedures to test ML-DSA (asked Oct 2024.)
- ANSSI suggested that because of hybridization, it may be sufficient in the beginning for one of the algorithms in the hybrid to resist attacks *(dunno..)*

On Certification of **PQC Modules**

FIPS 140-3 (for PQC)

- FIPS 140-3 is required by U.S. Federal government and many industrial standards.
- Currently focuses only on functional (test vector) and “checklist compliance” testing.
- Random numbers: SP 800-90 still good for PQC.
- Perhaps being introduced: “non-invasive” (ISO 17825) SCA leakage assessment for level 3+.

EUCA: Common Criteria and AVA_VAN

- High assurance level (EUCC: AVA_VAN.3+) is required for **Root of Trust IP**, **Smart Cards**, **Secure elements**, many types of **IoT** (SESIP).

CC AVA_VAN and “Attack Potential”

- AVA_VAN assesses real-life security via a “penetration test.” Can be very demanding.
- AVA_VAN security level is determined by "attack potential": A score-based system that measures cost of attack.
- Specialized 3rd party testing laboratories.
- *“Evaluators must have knowledge and experience of [..] side channel attacks (SCA) such as Timing Analysis, Machine Learning based SCA, Simple Power Analysis (SPA), Differential Power Analysis (DPA), Differential EM radiation Analysis (DEMA), Template Attacks (TA); fault injection attacks such as DFA [..]” **-- EUCC documents***

Recap: Protection Profiles for Chip Security

AVA_VAN.3 or 5 is common req. for Root of Trust and Security IC products.
We assume that this will not change (much) with Post-Quantum Cryptography.

[JSADEN011] **“SESIP Profile for PSA Certified™ Level 3”**
Root of Trust (PSA-RoT): 35 person-days of AVA_VAN.3 activities.

[PP-0084] **“Security IC Platform Protection Profile”**
EAL 4 augmented by AVA_VAN.5 and ALC_DVS.2

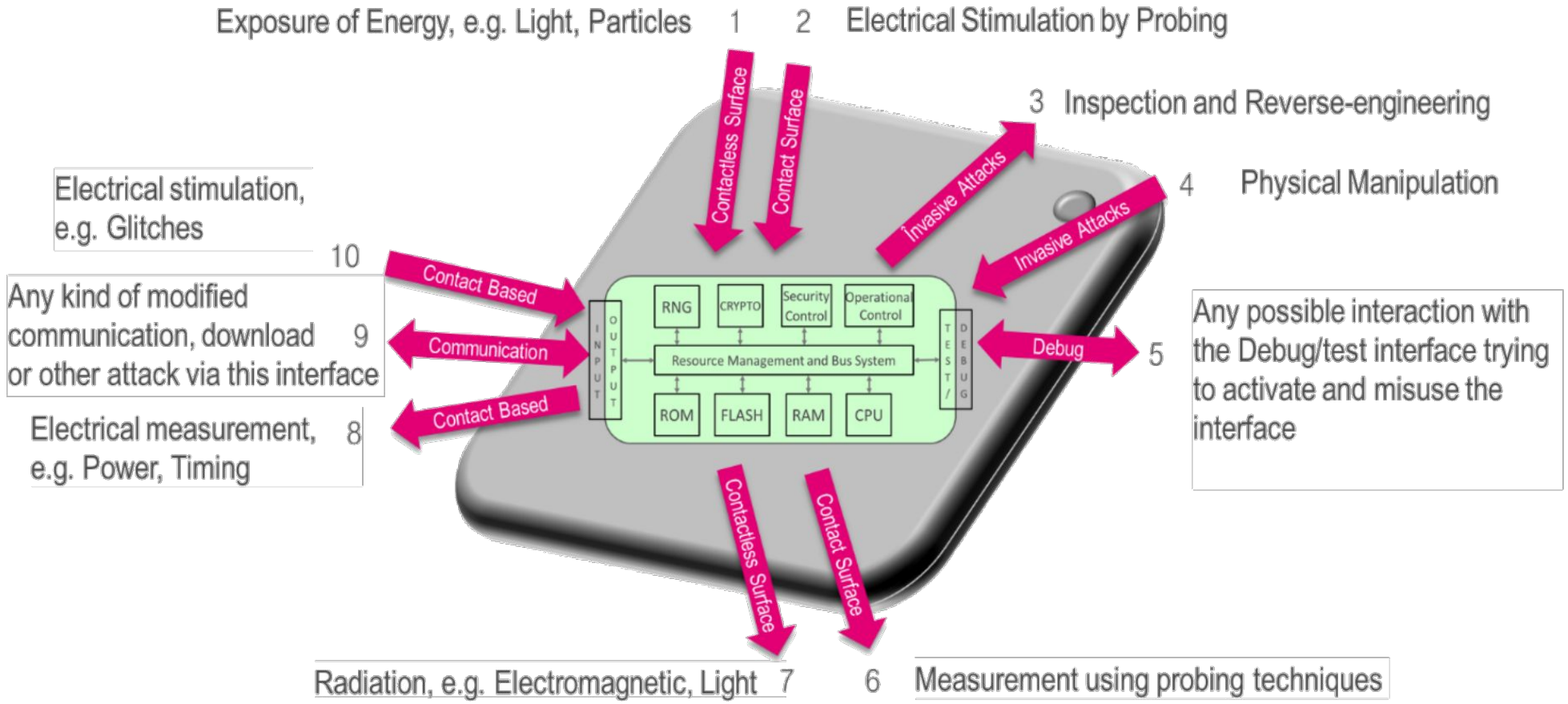
[PP-0117] **“Secure Sub-System in System-on-Chip (3S in SoC)”**
EAL 4 augmented by ATE_DPT.2, AVA_VAN.5, ALC_DVS.2 and ALC_FLR.2

(From PP-0117)

3.2 Threats

The threats described in this section are applicable to the base Protection Profile. For threats related to functional extensions see Chapter 7.

The following figure describes the attacks that are applicable to the TOE. The interactions related to the attacks are marked with red arrows.



https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0117V2b_pdf.pdf

Example Vuln: Processors are “SoCs” too…

AMD: Microcode Signature Verification Vulnerability

High **sirdarckcat** published **GHSA-4xq7-4mgh-gp6w** 2 weeks ago

Package	Affected versions	Patched versions
AMD CPUs	Zen 1-4 CPUs	Naples/Rome/Milan PI 2024-12-13 and Genoa 2024-12-16

Description

Summary

Google Security Team has identified a security vulnerability in some AMD Zen-based CPUs. This vulnerability allows an adversary with local administrator privileges (ring 0 from outside a VM) to load malicious microcode patches. We have demonstrated the ability to craft arbitrary malicious microcode patches on Zen 1 through Zen 4 CPUs. The vulnerability is that the CPU uses an insecure hash function in the signature validation for microcode updates. This vulnerability could be used by an adversary to compromise confidential computing workloads protected by the newest version of AMD Secure Encrypted Virtualization, SEV-SNP or to compromise Dynamic Root of Trust Measurement.

AMD SEV-SNP users can verify the fix by confirming TCB values for SNP in their attestation reports (can be observed from a VM, consult AMD's security bulletin for further details).

Severity

High 7.2 / 10

CVSS v3 base metrics

Attack vector	Local
Attack complexity	High
Privileges required	High
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N

CVE ID

CVE-2024-56161

February 2, 2025: https://github.com/google/security-research/security/advisories/GHSA-4xq7-4mgh-gp6w

AVA_VAN: Common Criteria Vulnerability Analysis

Attack Potential is evaluated with a score-based system that roughly maps to the "**cost of attack**" (think $€£.)

Considers attack **Identification** + **exploitation**, with many factors:

- Elapsed time (hours–months)
- Attacker Expertise (multiple)
- Knowledge (how restricted)
- Access to the TOE (samples)
- Equipment (common/bespoke)

("Application of Attack Potential" docs.)

AVA_VAN.1 Vulnerability Survey
- TOE resistance against BASIC Attack Potential (0-15)

AVA_VAN.2 (Unstructured) Vuln. Analysis
- TOE resistance against BASIC Attack Potential (16-20)

AVA_VAN.3 Focused (Unstructured) Vuln. Analysis
- TOE resistance against ENHANCED-BASIC AP (21-24)

AVA_VAN.4 Methodical Vuln. Analysis
- TOE resistance against MODERATE AP (25-30)

AVA_VAN.5 Advanced Methodical Vuln. Analysis
- TOE resistance against HIGH Attack Potential (31-)

Attack Potential: Example Calculation

AP Component	Identification	Exploitation
Elapsed time	2 (< one week)	6 (< one month)
Expertise	5 (expert)	4 (expert)
Knowledge of the TOE	4 (sensitive)	0 (public)
Access to the TOE	0 (< 10 samples)	0 (< 10 samples)
Equipment	3 (specialized)	4 (specialized)
Open Samples	0 (public)	0 (public)
Total	28 (AVA_VAN.4 / moderate AP range)	

SOG-IS: “Application of Attack Potential to Smartcards and Similar Devices”

https://www.sogis.eu/documents/cc/domains/sc/JIL-Application-of-Attack-Potential-to-Smartcards-v3.2.1.pdf

AVA_VAN vs Dilithium in a **Root of Trust** (SoC RoT)

- **A RoT provides immutable "silicon-rooted" security for SoCs**
  - Secure Boot of Firmware images (checking signatures)
  - Secure Firmware Updates (checking signatures)
  - Platform Attestation (with a signature)
  - Limited additional services such as random numbers, secure key storage
- **Caliptra** is an open-source SoC Root-of-Trust. Developed mainly by Microsoft, AMD, Google, NVIDIA: https://github.com/chipsalliance/Caliptra
- **Adams Bridge** is the **ML-DSA** unit for **Caliptra 2.0**, announced in Oct 2024.

Our main question:

- Could this open source Dilithium module be EUCC "**high**"? Depends..

Adams Bridge – One way to implement Dilithium

- **Status, March ‘25**: A standalone ML-DSA-87 accelerator, close to RTL freeze?
- Available, 100% SystemVerilog: https://github.com/chipsalliance/adams-bridge
- Only the “Category 5” parameters supported. Nothing related to Kyber visible.
- Self-contained module that does { KeyGen, Sign, Verify } from start to the finish. Includes a SHA3 module etc. Recently memory iface has been moved out.
- **Memory mapped (AHB)**: User writes keys, random, message (hash), sets trigger.

  Waits for status to become <ready> (perhaps intr), then read the signature out.
- **Very fast!** Verify: 20,000 cycles. / Sign: 160,000 cycles (40,000 per round).
- **Very big!** No shared components. Something like 400k GE + memories?

Quoting Adams Bridge Design Documents..

Threat Coverage

Physical Side-Channel Attacks

- **Types Covered:** Power analysis, electromagnetic (EM) analysis, and acoustic analysis.
- **Scope:** All operations involved in key generation and signature generation.
- **Countermeasures:** Combined masking and shuffling techniques to obscure power and EM signatures, and careful design to mitigate acoustic leakage.

(..)

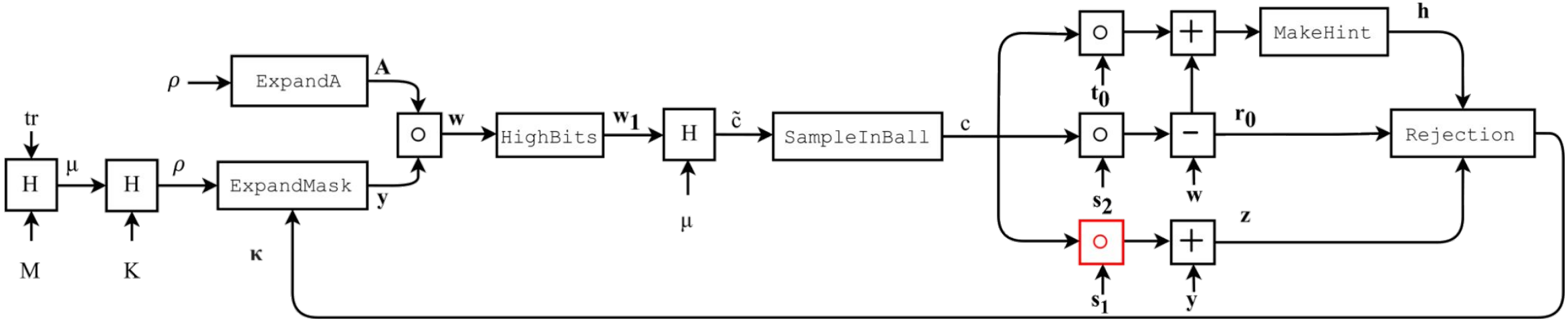
“Although Version 1.0 includes masking countermeasures, this report does not present TVLA results for masking countermeasures. These results will be provided in future releases.”

https://github.com/chipsalliance/adams-bridge/blob/main/docs/AdamsBridgeSCA.md

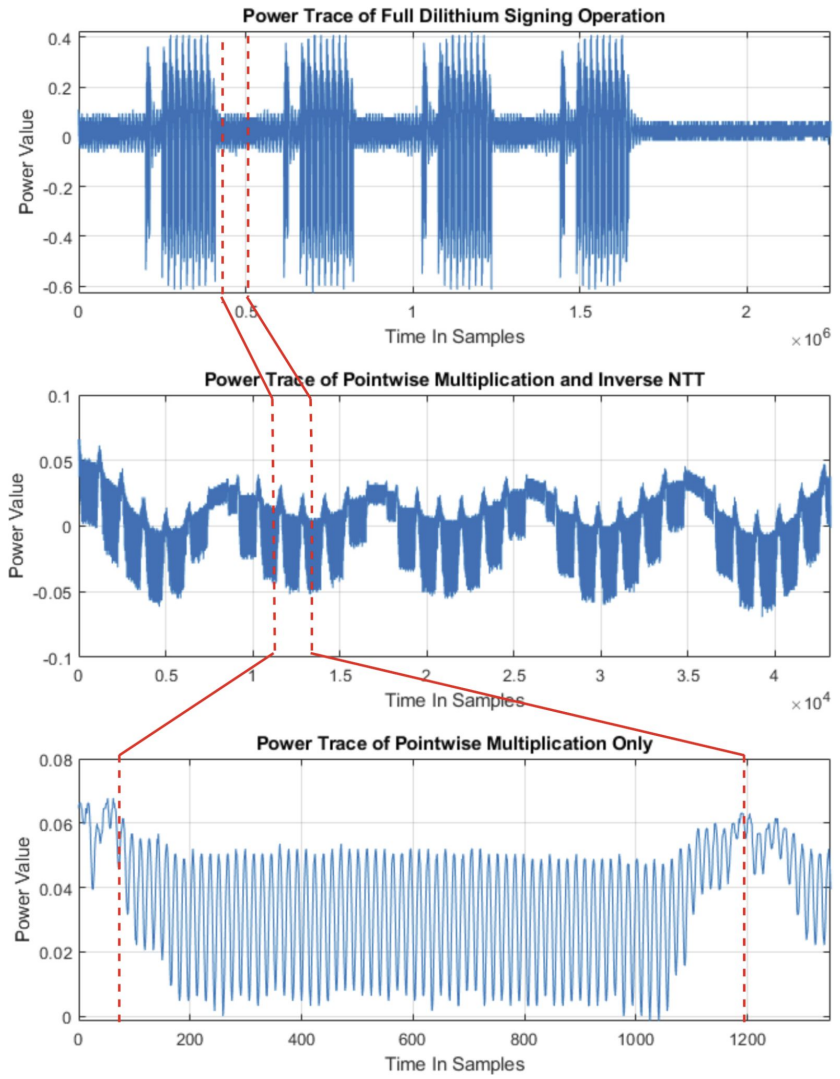
FAU was first-to-publish on Adams Bridge

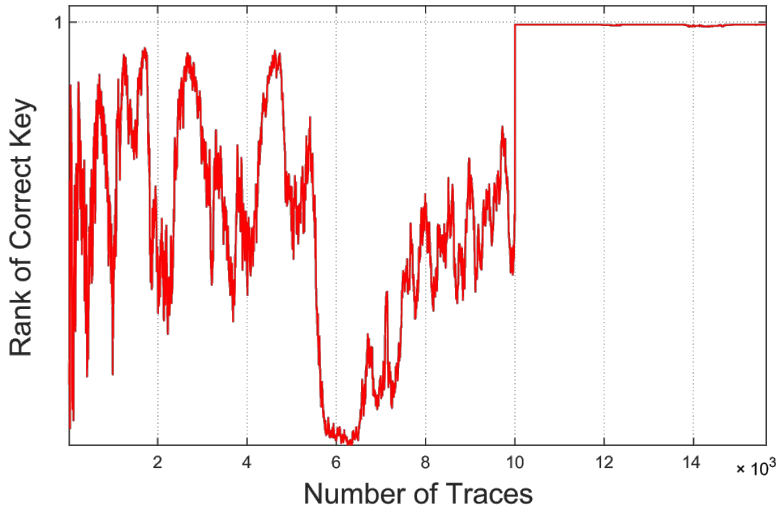
- A Correlation Power Analysis (CPA) attack on a secret-key multiplication step in late October ‘24 version of Caliptra’s Adams Bridge (“ABR”) Dilithium IP.
- Tested on CW305 A7 FPGA target. 10,000 traces to recover the secret key. This version didn’t have all of the countermeasures of the current ABR.

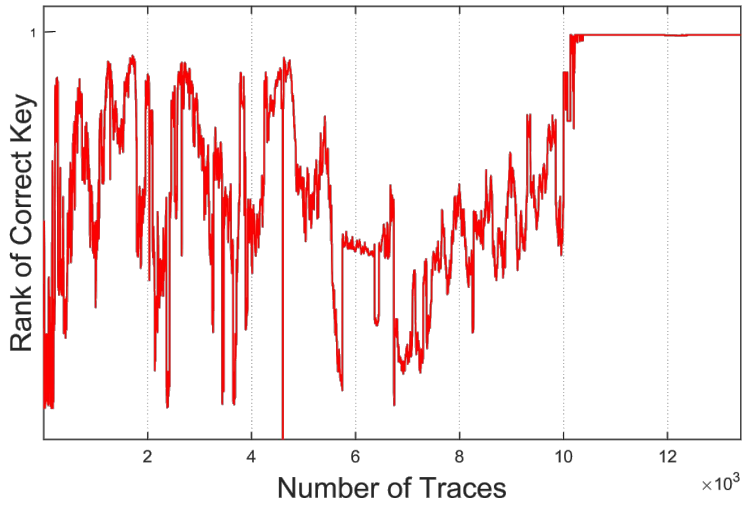
M. Karabulut, R. Azarderakhsh, *“Efficient CPA Attack on Hardware Implementation of ML-DSA in Post-Quantum Root of Trust.“* HOST 2025. https://ia.cr/2025/009



FPGA Target and FAU Key Extraction





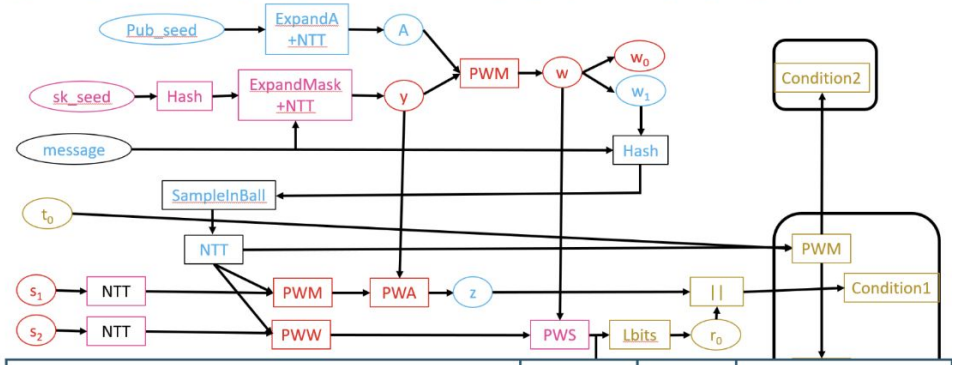


Design – Past tense: “Has it been exploited?”

E. Karabulut, K. Upadhyayula, **"Side-Channel Countermeasures for the Adams Bridge Accelerator"**, 2024 OCP Global Summit

Developing a Comprehensive SCA Threat Model

- Reviewing literature and listing existing SCAs
- Extending attack scope to include new and novel attacks
- Performing vulnerability assessment over data and control flows of our implementation
- Categorizing the attacks and setting up a priority list
- Revisiting our threat model after each RTL code review



Vectors	PWM	NTT	Protection
Does it have a secret input causing variation in the execution time?	No	No	Constant time design
Has it been exploited with profiled SCAs?	Yes	Yes	Hiding |Masking
Does it involve a sequential execution and have high signal-to-noise ratio?	Yes	Yes	Hiding |Masking
Has it been exploited with CPA/DPA?	Yes	No	Masking
Does it receive an observable public input while another operand is a fixed secret?	Yes	No	Masking

Not really masked (as researchers understand it)

- **Secret keys are not masked.**

  “Operations Protected with Masking: Point-wise multiplication and the first state of inverse NTT.”

- **Key generation is not protected at all**.

  “The key generation operation does not have a non-profiled attack vector since its nature is inherently secure against CPA-style attacks. This is because non-profiled attacks require multiple traces captured while constant secret or private values are being processed.”

  Dilithium may be used in a mode where secret keys are stored as short “seeds” and always expanded before use. Adams Bridge supports this..

Presilicon Testing of **Current Adams Bridge**

- Get VCD traces from verilator, DUT doing signing operations

- Presilicon VCD-to-Trace program reads VCD file, keeps track of all state bits and records Hamming distance for each clock cycle.

- Since the signal is very “clean”, not nearly as many traces are required than from FPGA-oscilloscope setup (rule-of-thumb, perhaps 10%).

- Very precise; we get exact cycle of leak points and can check (from VCD) the names of wires and signals that were active and causing it.

Dirty details: Dilithium Secret Key TVLA

Not everything in the secret key is secret!

- The basic TVLA fix-vs-random is really only suitable for symmetric ciphers

- Dilithium secret key has six components, two of which are actually secret:

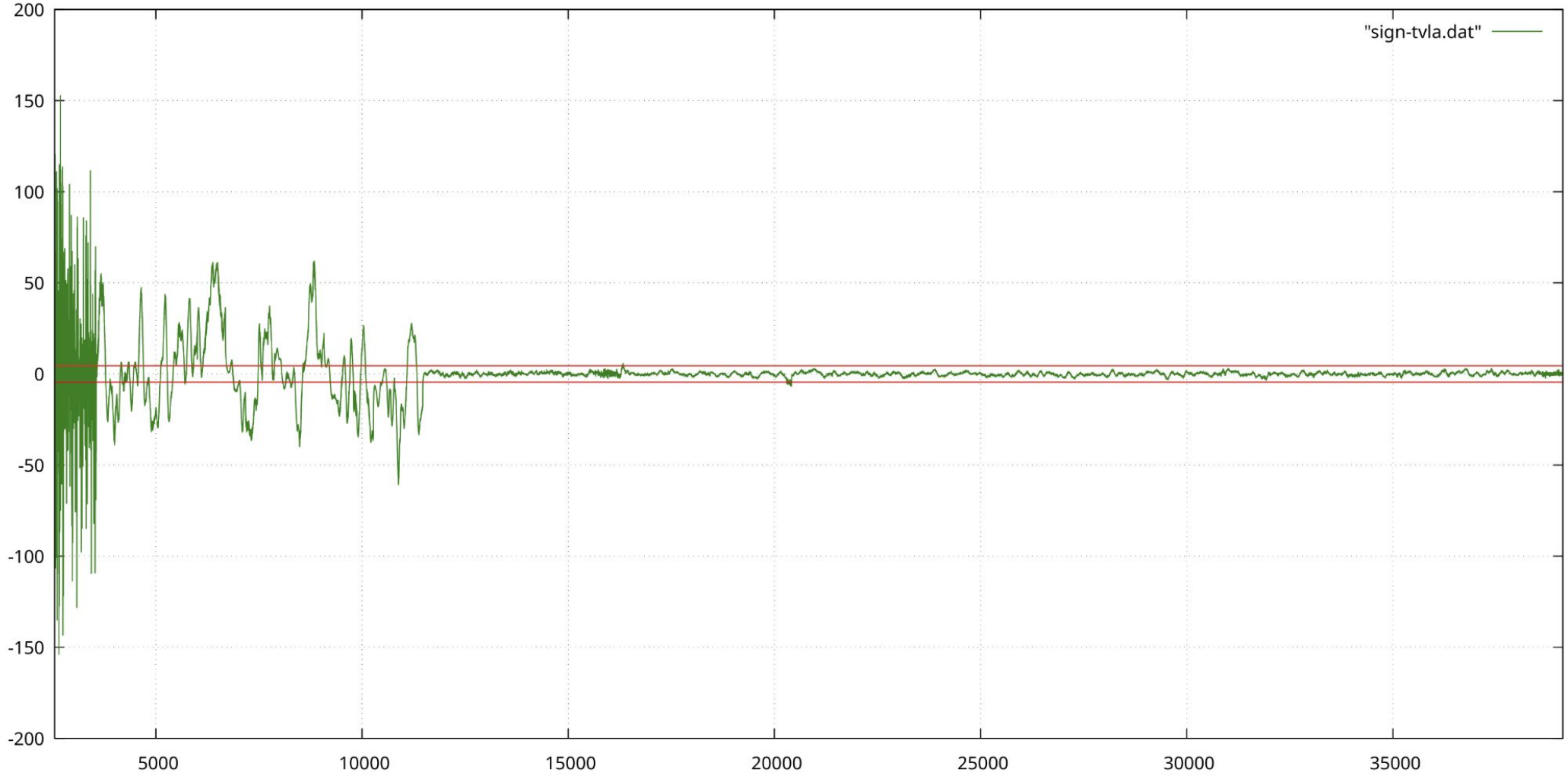
$$\mathbf{SK} = (\ \rho, K, \mathrm{tr}, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0\ )$$

- The public parts, e.g. matrix A expansion from symmetric seed ρ do not need protection. So one can easily get false positives in fix-vs-random

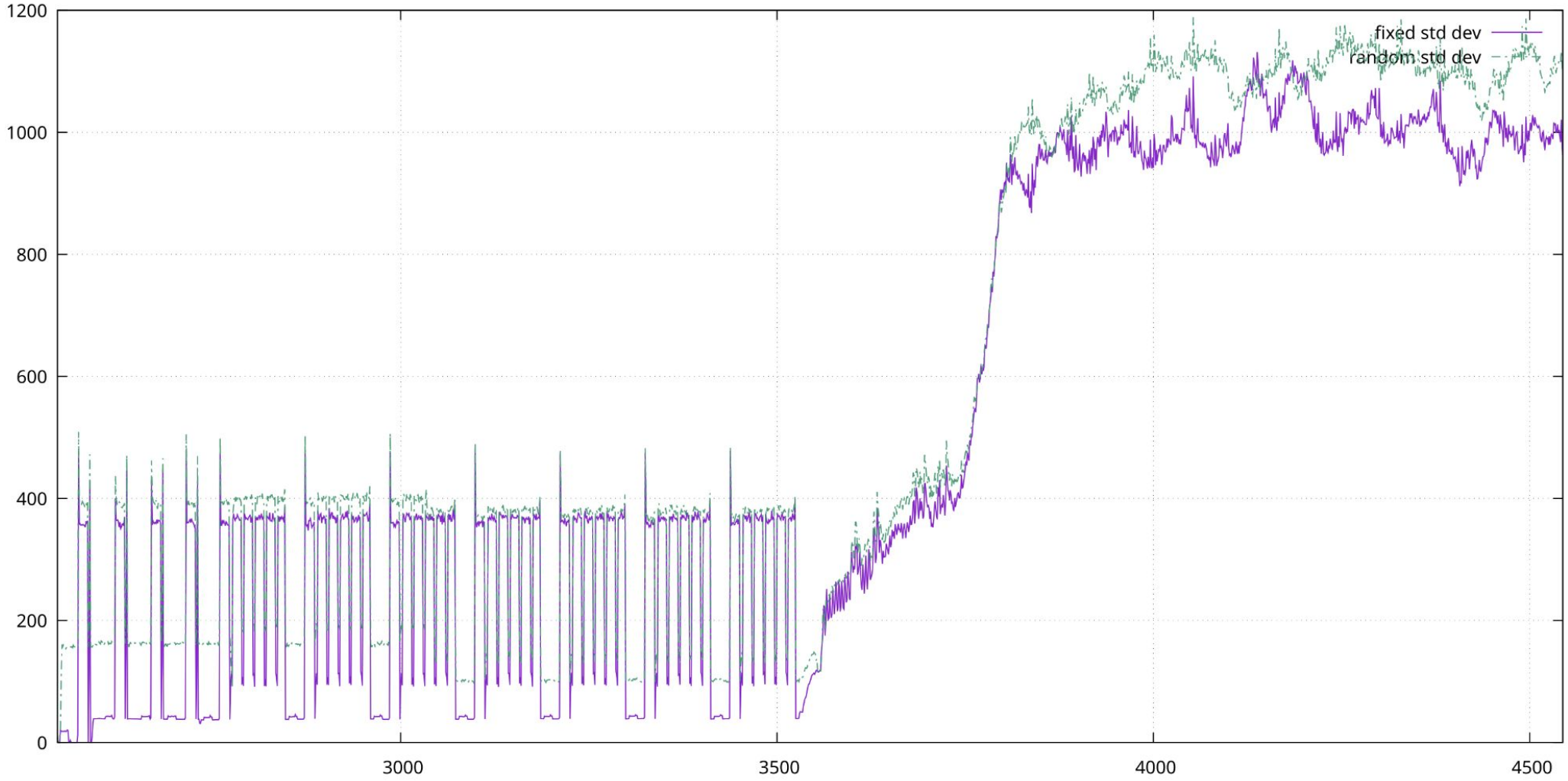
- One creates the test vectors for TVLA so that the random set is not entirely random, but just bits of the secret key bits are varied between traces.

- Alternative: randomize fully and just fix some secret key bits.

Well, Adams Bridge TVLA Doesn’t Look Good



Zoom into the leakage points



Leakage points: What would happen in AVA_VAN?

- **No surprise**: Leakage happens during early phases when the “plaintext” secret key is being moved about and transformed (NTT(s1), NTT(s2) ..)

- Partial masking is (by definition) considered “broken” by the theory.

  But leakage alone does not imply efficient key recovery or forgery attacks.

- **For AVA_VAN *perhaps* saved by wide data paths** – large chunks are being moved in each cycle so one learns the total hamming weight or distance.

- Further questions: **Where do the keys come from? How are they stored?**

Lattice Countermeasures are **Complicated**

- Masking splits secrets into “shares.” Successful measurement of an individual share does not leak secret info. One needs to convert sensitive arithmetic into masked operations.

Type	Relationship	Algebraic Object
Algebraic / Prime Field	$X = X0 + X1 \pmod{q}$	Mod 3329 (Kyber) or 8380417 (Dilithium)
Algebraic / Power-of-2	$X = X0 + X1 \pmod{2^n}$	Some Lattice Crypto, SHA2, etc
Boolean / Binary Field	$X = X0 \oplus X1$	Nonlinear Functions, shifts, symmetric Crypto

- **Most cryptographers agree**: Masking and other attack mitigation techniques for PQC algorithms are much more complex than countermeasures for older cryptography.

- **Why?** The algorithms are not homogenous like RSA or ECC but contain a number of dissimilar steps. One may have to design a dozen different gadgets for one algorithm.

My Humble Conclusions and Recommendations

- Attack papers do not even claim to describe *all of the vulnerabilities*, often just what happened to be “enough” (the low hanging fruit) to break particular target.

- Researchers know that many side-channel attacks work against Dilithium, but there has not been attack papers because there has not been *attack targets*.

  Lattice crypto countermeasure “theory” work has been going on for many years.

- I recommend taking a theoretically sound **masking approach** as a basis – must be complemented with ad hoc countermeasures, and **adversarial in-house analysis**.

- **Importantly:** Masking and other countermeasures **impact architecture**. Don’t try to “patch” countermeasures into an unprotected implementation!