Mind the Gap: Studying the Insecurity of Provably Secure Embedded Trusted Execution Architectures

Marton Bognar, Jo Van Bulck, Frank Piessens

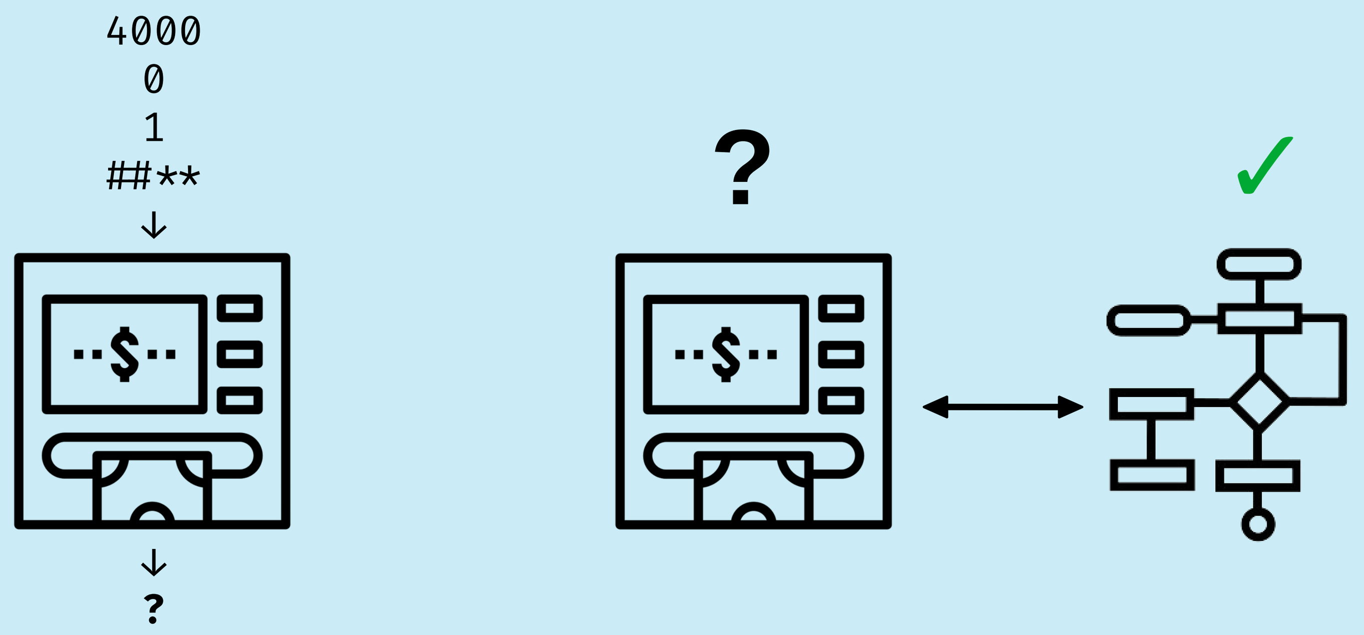
imec-DistriNet, KU Leuven, Belgium

marton.bognar@kuleuven.be

Providing evidence for security

Inductive methods: A successful attack breaks the security claim, a failed attack supports, but does not guarantee it.

Deductive methods: Can guarantee properties of a model, but the connection between the model and the implementation should be strong.



Goal: Narrowing the gap

- Fundamentally impossible to close [1]
- Narrowing the gap: case study approach
  - Deductive + inductive methods
  - Deriving guidelines from experimental evidence
  - Impactful open-source systems with precise security claims, deductive proofs

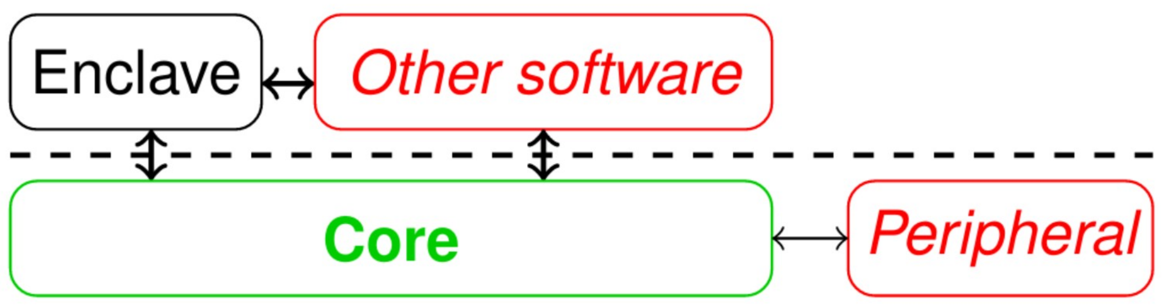
Methodology

- Identify falsifiable assumptions
- Validate the implementation
- Identify missing attacker capabilities
- Check proofs

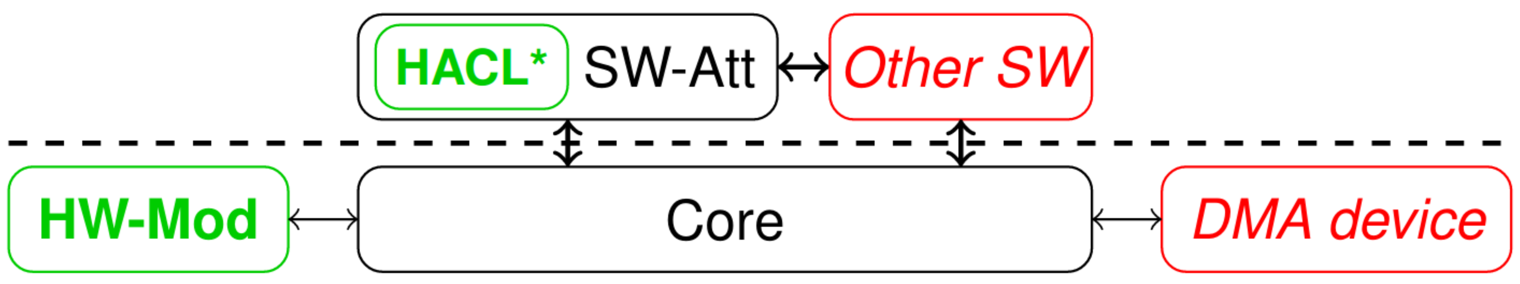
Three attack classes:
- Implementation-model mismatches
- Missing attacker capabilities
- Deductive errors

Case study systems: $Sancus_V$ [2], VRASED [3]

$Sancus_V$: secure interrupt handling
- Verilog hardware implementation
- Operational semantics, pen-and-paper proof



VRASED: secure remote attestation
- Hybrid architecture: *HW-Mod* in Verilog + *SW-Att* based on HACL*
- State machine model extracted from Verilog, mechanized proofs

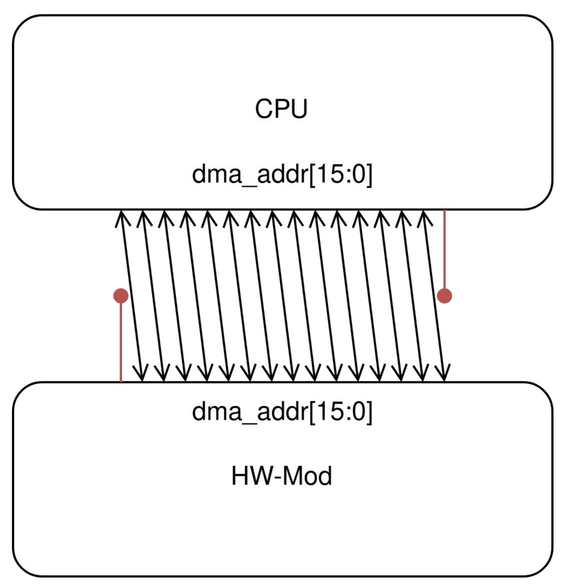


Implementation-model mismatches

Successful attacks on the implementation that fail in the model

$Sancus_V$	
V-B1	Context-free instruction lengths
V-B2	Maximum instruction length
V-B3	reti from outside an ISR
V-B4	Restarting the enclave from ISR
V-B5	Number of enclaves
V-B6	Accessing unprotected memory
V-B7	Protected interrupt functionality

VRASED	
VI-B1	DMA address bus
VI-B2	Consistent key size



Guideline: Maintain a systematic connection between the implementation and the model

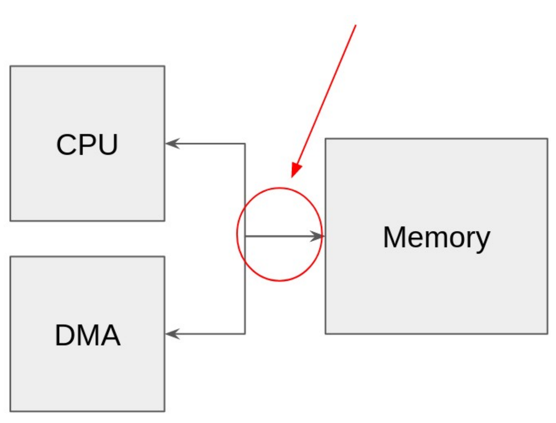
VRASED: model derived from implementation, fewer errors!

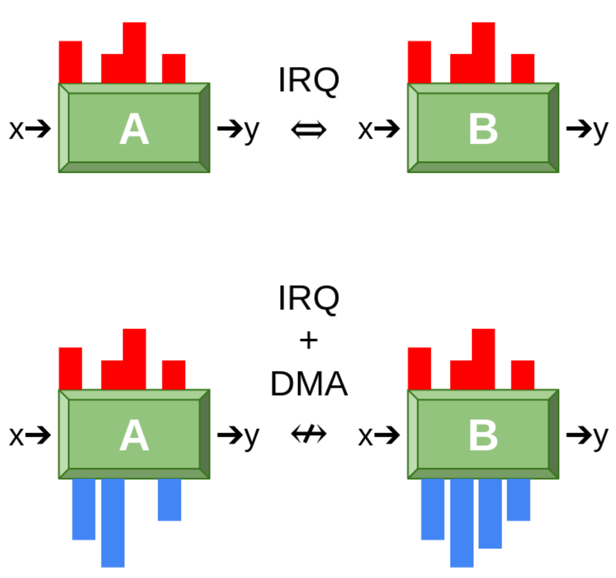
Missing attacker capabilities

Attacks that cannot be represented in the model

VRASED	
VI-C1	Shared peripheral bus
VI-C2	Secure stack initialization
VI-C3	End-to-end timing of trusted software
VI-C4	Interrupt latency timing
VI-C5	DMA side-channel

$Sancus_V$	
V-C1	DMA side-channel
V-C2	Watchdog timer IRQ





Guidelines:
- Study attack literature
- Model attacker capabilities + composition
- Audit interfaces between verified/unverified, trusted/untrusted components

Results

$Sancus_V$:
- Implementation-model mismatches: **7**
- Missing attacker capabilities: **2**
- Deductive errors: **0**

VRASED:
- Implementation-model mismatches: **2**
- Missing attacker capabilities: **5**
- Deductive errors: **1**

Resources:
- Repository: https://github.com/martonbognar/gap-attacks
  - Including a CI pipeline for the attacks
- Paper: https://mici.hu/papers/bognar22gap.pdf

References

[1] C. Herley and P. C. van Oorschot, "Science of security: Combining theory and measurement to reflect the observable," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 12–22, 2018.

[2] M. Busi, J. Noorman, J. Van Bulck, L. Galletta, P. Degano, J. T. Muhlberg, and F. Piessens, "Provably secure isolation for interruptible enclaved execution on small microprocessors," in *33rd IEEE Computer Security Foundations Symposium (CSF)*, Jun. 2020, pp. 262–276.

[3] I. D. O. Nunes, K. Eldefrawy, N. Rattanavipanon, M. Steiner, and G. Tsudik, "VRASED: A verified hardware/software co-design for remote attestation," in *28th USENIX Security Symposium*, 2019, pp. 1429–1446.