MARVELL

2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SANTA CLARA, CA, USA
March 2 - 5, 2026

Security Verification in Practice

Lessons from Pre-Silicon Analysis of SoC Subsystems

Primary author:

Yashwanth Kumar
Senior Product Security Verification Engineer

Co-author:

Rachana Maitra
Senior Principal Corporate SDL

March 5, 2026

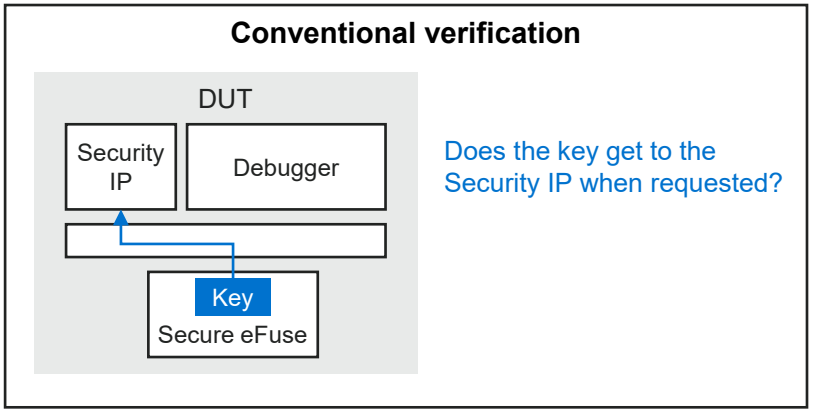
Table of contents

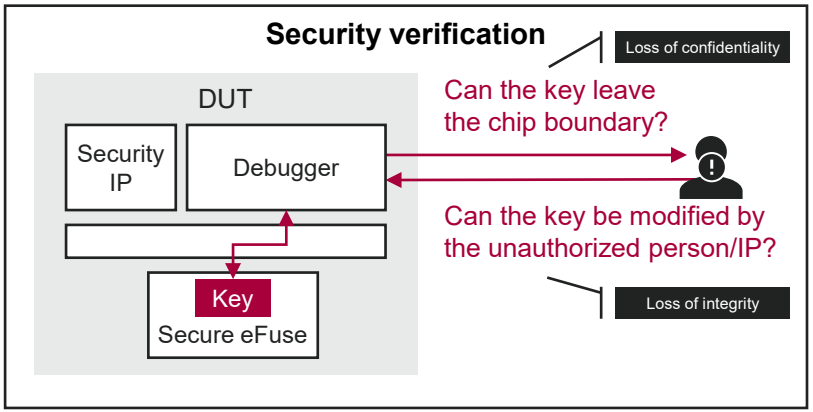
Motivation

- As System on Chip (SoC) complexity increases, the attack surface area increases
- Not a ‘good to have’ anymore
- Functional DV is not Security DV
  - Functional DV catches design correctness but not security issues like leakage or unauthorized access, etc.
- Security bugs are expensive when found late
  - Need for shift-left approach

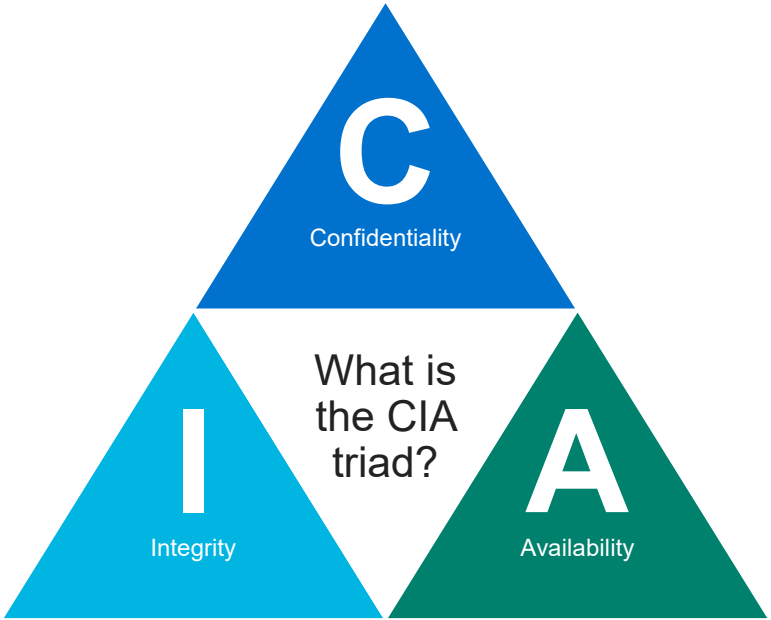
What is security verification?

- Anticipate the unexpected or unwanted behavior that can be exploited for a security breach
- Test for negative conditions that can lead to a security breach





Security verification methodologies

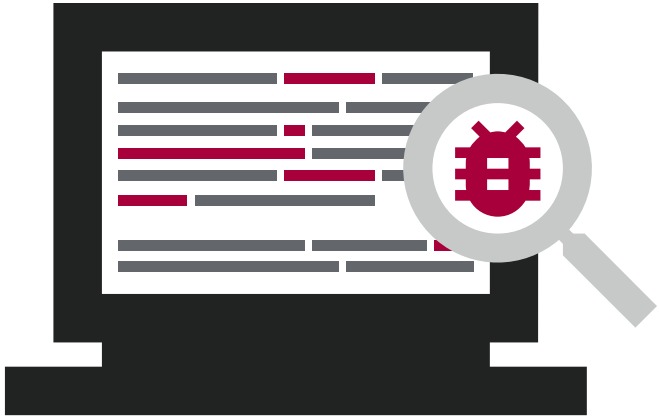


Overview of methodologies

- Static security linting
- Dynamic verification – Information Flow Tracking (IFT)
- Side Channel Analysis (SCA)

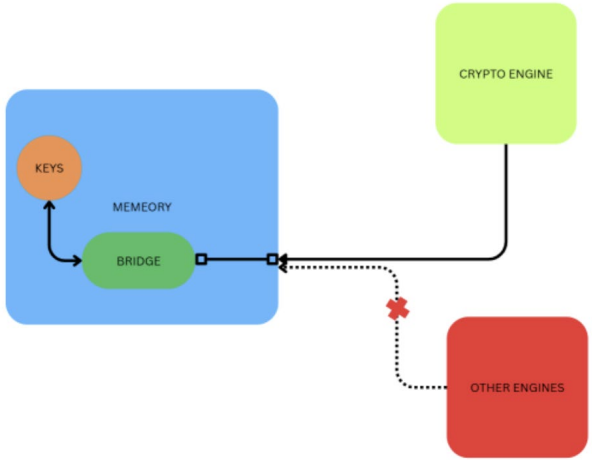
Static security linting

- No simulation required
- Detects issues like unsafe Finite State Machines (FSM) states and transitions, bitwise comparing, etc.
- Mapping to Common Weakness Enumerations list (CWE) from MITRE
- Must be the first line of defense, as it should not be a complicated add-on to existing verification flows



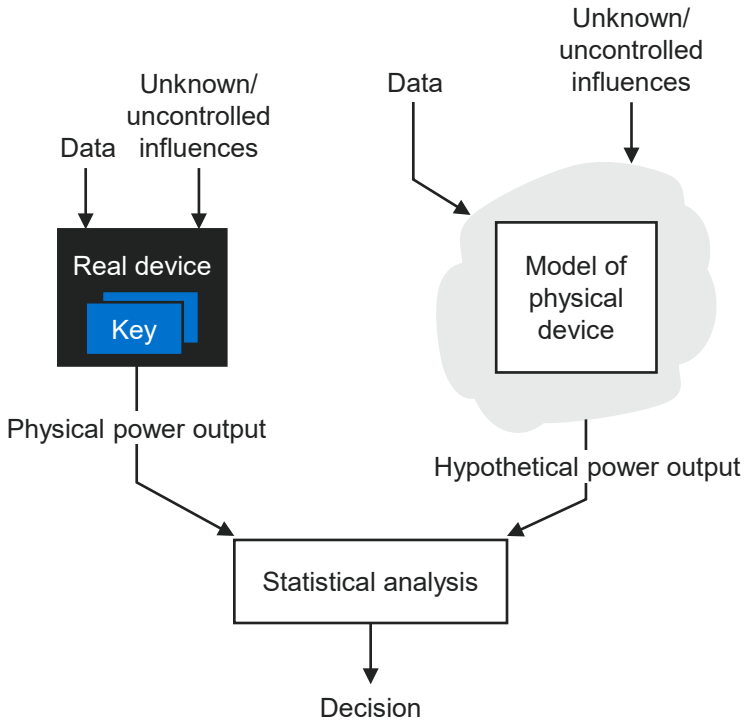
Information flow tracking

- A dynamic verification method that requires simulation
- Tracks assets in terms of their confidentiality and integrity
- Highly useful to verify if control-access or read/write permissions are properly enforced
- Example: Keys accessible only by authorized master



Side channel analysis

- Simulation – monitors switching activity
- Mimics a realistic attack scenario
- Detects key-dependent leakage
- Supports different combinations of evaluations – CPA/DPA, etc.
- A good test to ensure the effectiveness of the masking of the design



Case studies and results

1 Static security linting

- The design target was a control-oriented subsystem with multiple FSMs and data/control path logic
- Extensive focus on FSM
- Bitwise compare → risky
- Other CWE-related issues
- Flagged certain critical issues, such as bit-by-bit/byte-by-byte comparison, FIFO over/underflow, counter excess size, etc
- Security-focused linting = deeper insight

```
logic [127:0] aes_key;
logic [7:0] status;

always_ff @(posedge clk) begin
  if (aes_key[0])
    status <= 8'hA5;
  else
    status <= 8'h5A;
end
```

← Direct information leakage

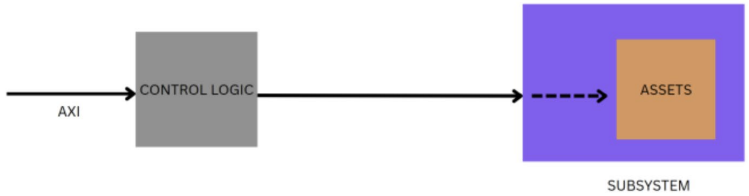
```
// Original RTL

for (int i = 0; i < 8; i++) begin
  if (wr_en && config_select && wr_data[i])
begin
    ctrl_reg[i] <= wr_data[i];
  end
end
```

```
// Fix for the issue

reg config_lock;  // Lock to block config writes until secure state

// Lock logic: starts locked (1), unlocks after secure init
always_ff @(posedge clk or negedge rst_n) begin
  if (!rst_n)
    config_lock <= 1'b1;              // Locked after reset
  else if (secure_init_done)
    config_lock <= 1'b0;              // Unlock after secure init
end

// Protected config write: only if unlocked (!config_lock)
for (int i = 0; i < 8; i++) begin
  if (wr_en && config_select && !config_lock && wr_data[i]) begin
    ctrl_reg[i] <= wr_data[i];
  end
end
```

Case studies and results

2 Information Flow Tracking (IFT)

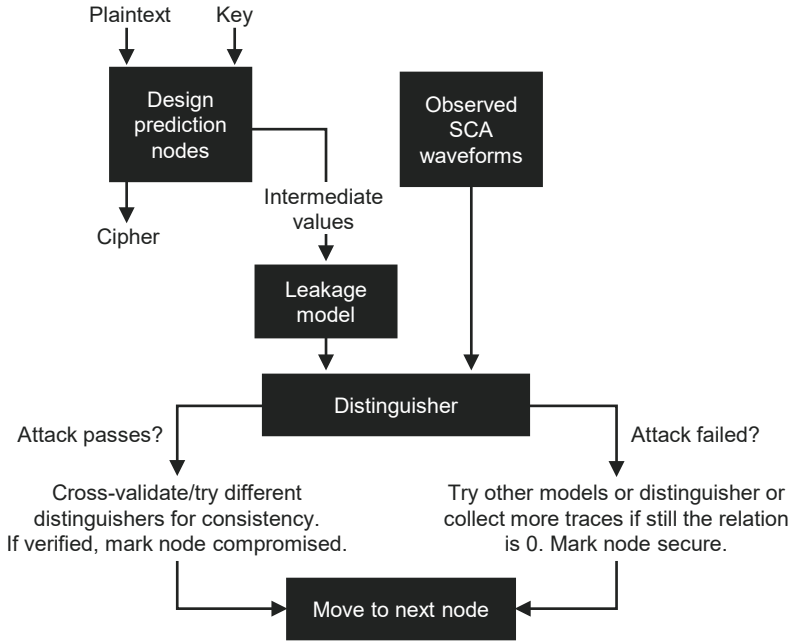
- Validated AC policy behavior
- Emphasized context-driven rule creation
- Necessitated additional protection for eHSM assets
- Revealed architectural improvement
- Confirms design resilience and uncovers opportunities for proactive security hardening
- IFT was used to verify the CIA in an asset-centric approach



Case studies and results

3 Side Channel Analysis (SCA)

- Target – crypto engine/operation simulation
- Focus on power-based leakage
- Leakage models – Hamming distance and Hamming weight
- Distinguishers – SPA, DPA, CPA, ML
- First round most vulnerable
- Unmasked shows leakage at 5k+ traces
- No leakage for masked design



Test case	Number of traces	Working distinguisher (DPA/CPA)	Subkeys recovered (out of 16)	Leakage detected
Unprotected AES	500	None	0	No
	1000	CPA	7	Partial
	5000	Both	16	Yes
	10000	Both	16	Yes
Masked AES	500	None	0	No
	1000	None	0	No
	10000	None	0	No

Challenges and observations

Technical challenges

- Static false positives
- IFT stimulus dependency
- SCA limitations (AES modes / non-AES engines)

Collaboration and mindset observations

- Security DV ≠ functional DV mindset
- Asset-centric approach essential
- Requires design ↔ DV collaboration

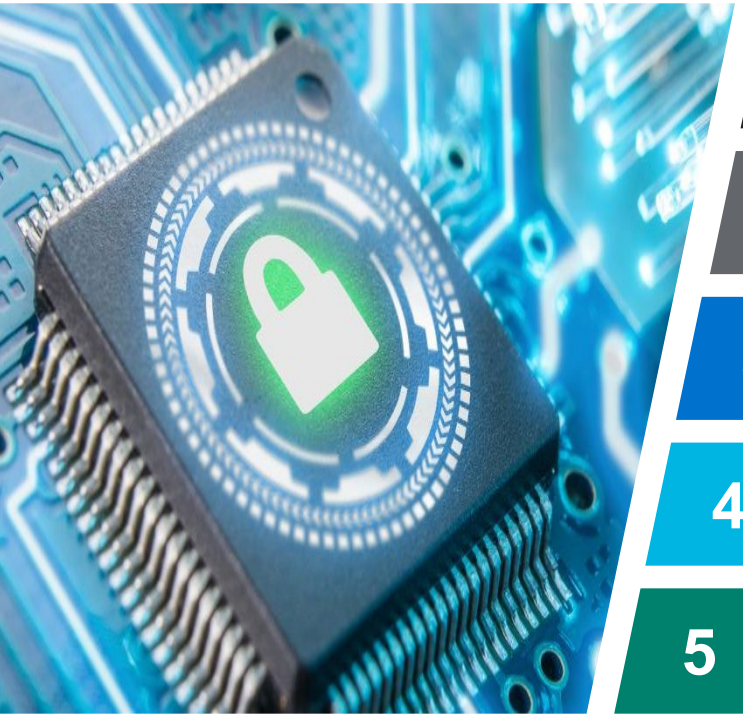
Analysis challenges

- Asset scoping and threat modeling
- Requires contextual understanding
- Pre/post analysis designer discussions

Industry evolution observations

- Templates reusable for future IPs
- Companies need internal BLUE-team style security DV groups
- Industry and EDA vendor collaboration required

Conclusion



1. Practical, scalable methodology
2. Flag subtle security bugs early
3. Valid across multiple subsystems
4. Enable shift-left security
5. Future: scope increase, automation and AI inclusion

Appendix

- DV – Design Verification
- IFT – Information Flow Tracking
- SCA – Side Channel Analysis
- FSM – Finite State Machines
- CWE – Common Weakness Enumerations
- SPA – Simple Power Analysis
- CPA – Correlation Power Analysis
- DPA – Differential Power Analysis
- ML – Maximum Likelihood
- FIFO – First In First Out
- AC – Access Control
- AES – Advanced Encryption Standard
- AI – Artificial Intelligence

MARVELL

2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

Thank You

Essential technology, done right™