CONFIDENTIALITY ASSURANCE: A KEY COMPONENT OF HARDWARE SECURITY

Varun Sharma : *Real Intent*
Vikas Sachdeva : *Real Intent*
Vinod Viswanath : *Real Intent*

Abstract: Ensuring confidentiality in hardware is paramount in safeguarding the security triad—confidentiality, integrity, and availability. Modern processors face vulnerabilities such as Spectre-like attacks and transient execution hazards, where improper implementations can inadvertently expose sensitive signals. In our work, we address a critical gap: the absence of a unified, early-stage sign-off method to verify confidentiality at the RTL design phase.

Our approach leverages a novel verification framework that captures Secure Data Transaction Intent across user-defined signals. Through static analysis, our system identifies illegal data flows that could lead to the exposure of architecturally restricted information. The tool, Sentry, enforces these checks robustly at the RTL stage. It verifies that sensitive information is shielded from unauthorized propagation, even under scenarios of aggressive prediction or mis pipelined execution, thereby mitigating risks from stale data forwarding and other architectural nuances.

In summary, our poster presents a scalable, RTL-based verification technique that significantly enhances hardware security by reliably detecting and preventing illegal information flows. This work not only reduces dependency on post-silicon countermeasures but also improves the overall resilience of system-on-chip designs.

I. Motivation

Confidentiality is a key aspect of security triad: Confidentiality, Integrity, Availability

- **Spectre-like vulnerabilities** arise from improper implementation and transient execution. (Scenario-1,2,3)
- **Improper AES implementation** exposes signal. (Scenario-1)

NO reliable single sign-off method exists across industry

- **Verification is complex**, involving multiple stages: Simulation, Formal, and post-Si validation.
- Signing off on confidentiality is **challenging without verification in unknown scenarios**.
- **No single sign-off method exists to help in early detection** at RTL stage.

Proposed New Technology

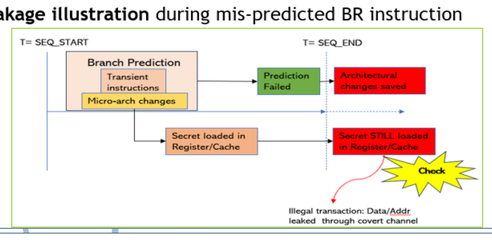
- **Capture Secure Data Transaction Intent across User Defined Signals.**
- Static analysis to identify **Illegal Data Flow causing Security Violations.**
- Facilitates **early detection with minimal constraints during the RTL** design phase.
- **Scales to full SoC sized design**, saving time and effort.

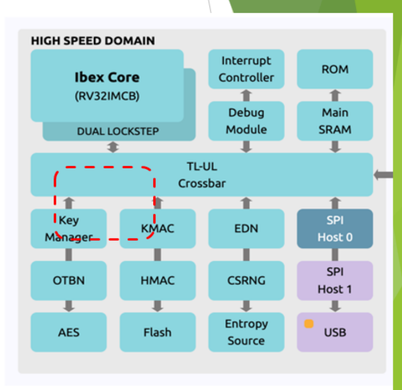
Leakage Scenario

1. Expose architecturally restricted data
2. Allow predictor training in one domain to influence behaviour in another domain
3. Caused by stale data forwarding

II.a. Exposure of Restricted(Sensitive) Information during *Transient Execution*

- In this paper we will utilize open source OpenTitan's sub-block:
  - OpenTitan has a co-processor OTBN (**OpenTitan Big Number accelerator**)
  - **OTBN** is a processor, specialized for the execution of security-sensitive asymmetric (public-key) cryptography code, such as RSA or ECC
- OTBN has preventive measure: NO Transient Execution during Branch and Jump-and-link instructions (BR/JAL/JALR):
  - **Stalls for one cycle after jump and branch instructions** rather than attempt to predict the next PC
  - → For demonstration, we **enabled the transient execution** with RTL modification
    `i.e. insn_prefetch = 1'b1; //Code change`
- **Leakage illustration** during mis-predicted BR instruction



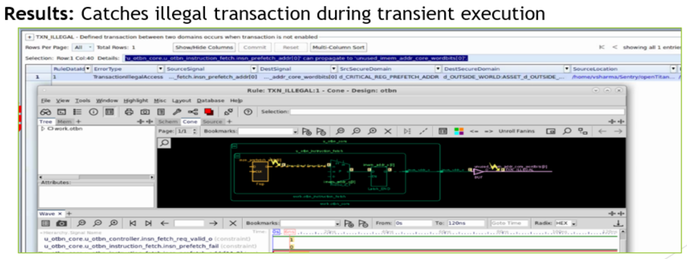


II.b. Catching *Sensitive Information Leak*

- **Define** `prefetch Address registers and` **`legal destinations`**`:`
  `u_otbn_core.u_otbn_instruction_fetch.insn_prefetch_addr`
  `Outside world (Unused signals or POs):`
  `unused_imem_addr_core_wordbits`
- **Define** the mis-prediction Branch sequence:
  `Define a sequence named` **`BRANCH_PRED_FAIL`**`:`

```
Set imem_rvalid_core constraint to 1 at time t=0
Set u_otbn_core.insn_fetch_req_valid constraint to 1 at time t=0
Set u_otbn_core.u_otbn_instruction_fetch.insn_prefetch_fail constraint to:
      0 at time t=0
      1 at time t1 after 20 cycles
```

- **Check:** It is Illegal that during branch prediction computation, prefetch address registers leak to outside-world
- **Results:** Catches illegal transaction during transient execution



III.a. Confidential signals should NOT be *exposed w/o Encryption*

- Here we are using OpenTitan's **A**dvanced **E**ncryption **S**tandard (AES) hardware IP block
  - The AES unit is a cryptographic accelerator that accepts requests from the processor to encrypt or decrypt 16 byte blocks of data.

Confidential Signal

- Contains secret information
- `keymgr_key_i.key[keymgr_pkg::Shares-1:0][keymgr_pkg::KeyWidth-1:0]`
- `reg2hw.key_shares[aes_reg_pkg::NumRegsKey-1:0].q[31:0]`
  - *where s is in [0:keymgr_pkg::Shares-1]*
- `reg2hw.iv[aes_reg_pkg::NumRegsIv-1:0].q[31:0]`
- `reg2hw.data_in[aes_reg_pkg::NumRegsData-1:0].q[31:0]`

Exempt Signals

- The following signals are exempt to contain encrypted information.
- `hw2reg.data_out[aes_reg_pkg::NumRegsData-1:0].d[31:0]`
- **Encryption is ensured if goes through XOR gates**

Exposed Signals

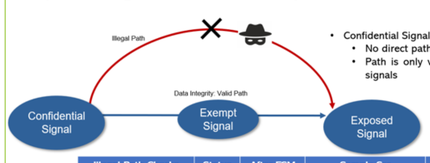
- All output ports are observable either by SW or by other HW modules:
- `idle_o`
- `tl_o`
- `alert_tx_o`

Data transfer from confidential to exposed via exempt signals(ensures encryption) is **NOT** a violation.

III.b. Catching *Confidential Signal propagation w/o Encryption*

- Define the **Secure Domains:**
  - Confidential
  - Exempt
  - and Exposed
- Define the **two Checks:**
  - chk_CONF*_to_EXPOSED: Illegal transaction from Confidential to Exposed NOT going through Exempt signals
  - chk_CONF*_to_EXEMPT: Illegal transaction from Confidential to Exempt NOT going through XOR gates

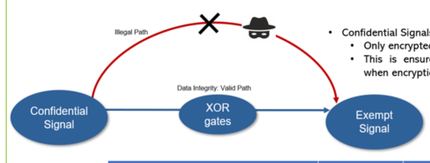
Exposed Signals can not Compromise Secure Data



- Confidential Signals to Exposed signals
  - No direct path should be possible
  - Path is only valid through exempt signals

Illegal Path Checks	Status	After FSM constraints	Sample Src	Sample Dest
chk_CONF1_to_EXPOSED	FAIL	PASS	keymgr_key_i.key[1][199]	tl_o.d_user.data_intg[0]
chk_CONF2_to_EXPOSED	FAIL	PASS	reg2hw.iv[3].q[18]	tl_o.d_user.data_intg[0]

Data Visible on Exempt Signals is Always Encrypted



- Confidential Signals to Exempt signals
  - Only encrypted data should reach
  - This is ensured by data only reaches when encryption is valid

Illegal Path Checks	Check	Status
chk_CONF_to_EXEMPT_illg *s* where *s* ∈ [1:7] when u_aes_core.cipher_out_valid !='d3	TXN_NO_ILLEGAL	PASS
Valid Path Checks	**Check**	**Status**
chk_CONF_to_EXEMPT_vld when u_aes_core.cipher_out_valid =='d3	TXN_VALID	PASS

IV. Summary and Conclusion

- **Challenges in Security Countermeasure Verification**:
  - Traditional formal and simulation-based approaches face significant risks, including scalability, missing relevant corner cases and relying on incorrect assumptions.
- **Effectiveness of New Technology (Tool Name: Sentry)**:
  - Sentry ensures no unexpected information flow from internal registers to exposed points in the design, regardless of the applied stimulus.
  - Provides robust verification of security countermeasures with negative cases.
  - Reduces risks of relying on incorrect assumptions.
  - Scalable to multi-million gate design at RTL design stage.
  - Enhances the reliability and security of designs.
- **Demonstration of Sentry's Capabilities with OpenTitan**:
  - **OTBN Co-Processor**: Sentry identifies information leakage if inserted RTL bug leads to transient execution.
  - **AES Accelerator**: Sentry ensures that secret key material only propagates to the register interface in encrypted form.

Benchmarking of Sentry with HRoT OpenTitan's blocks:

Design	GateCount (Nand2)	Check #	Performance/Capacity
OTBN	352K	2	<1min/1M
AEs	118K	8	<1min/568K
Earl Grey	13.3M	16	45min/58G