Eliminating the Risk of Malicious Counterfeit Chiplets

Scott Best, Sr.Director Rambus Security

sbest@cryptography.com

Rambus solutions serving data-intensive markets

Data Center | Automotive | 5G/Edge | Government | IoT

DDR5 RCD,
TS, SPD HUB
DDR4 RCD,
DB

HBM3
GDDR6
PCIe 4/5/6
CXL 2/3

Chips (Product Sales)

Memory
Interface Chips

Silicon IP (IP Licenses)

Interface IP:
Memory and SerDes
PHYs and Controllers

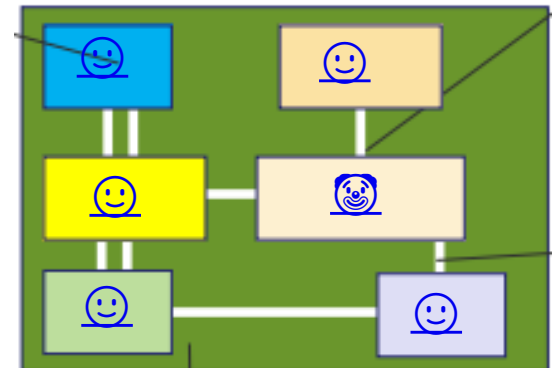
Security IP:
Secure Silicon,
Secure supply-chain

Focus for today

CRYPTOGRAPHY RESEARCH™
a division of Rambus

The Problem

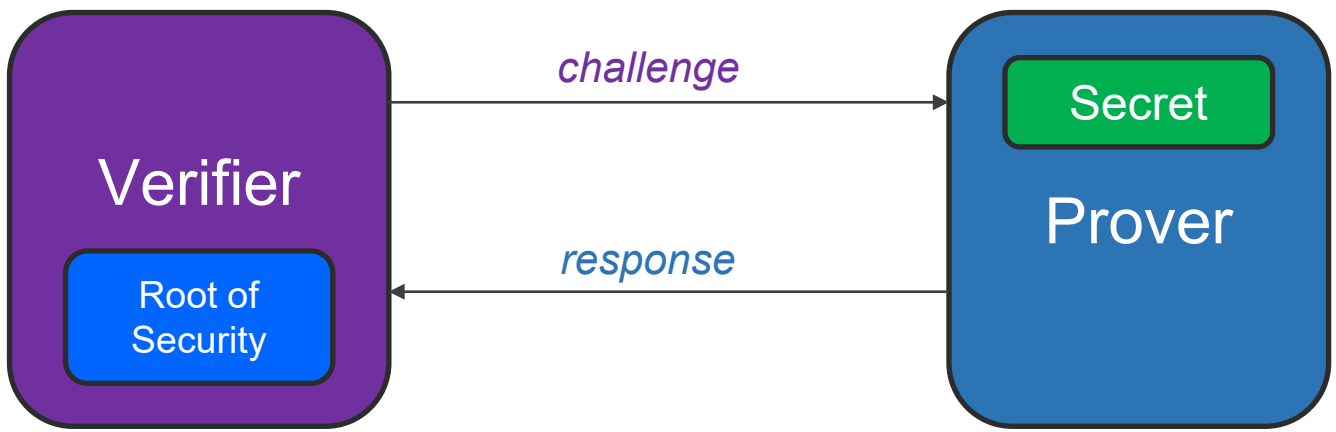
- High-mix heterogeneous SiPs are the ideal breeding ground for malicious “hardware trojan” chiplets
- The overall “quantifiable assurance” of your SiP is only as good as the least secure chiplet



- One of the chiplets in the SiP must be responsible for verifying the authenticity of every other chiplet
  - Good choice for this: the chiplet with the root-of-security (i.e., the one which is responsible for overall secure-boot)
  - See: Caliptra specification for minimum requirements

How to Verify Authenticity

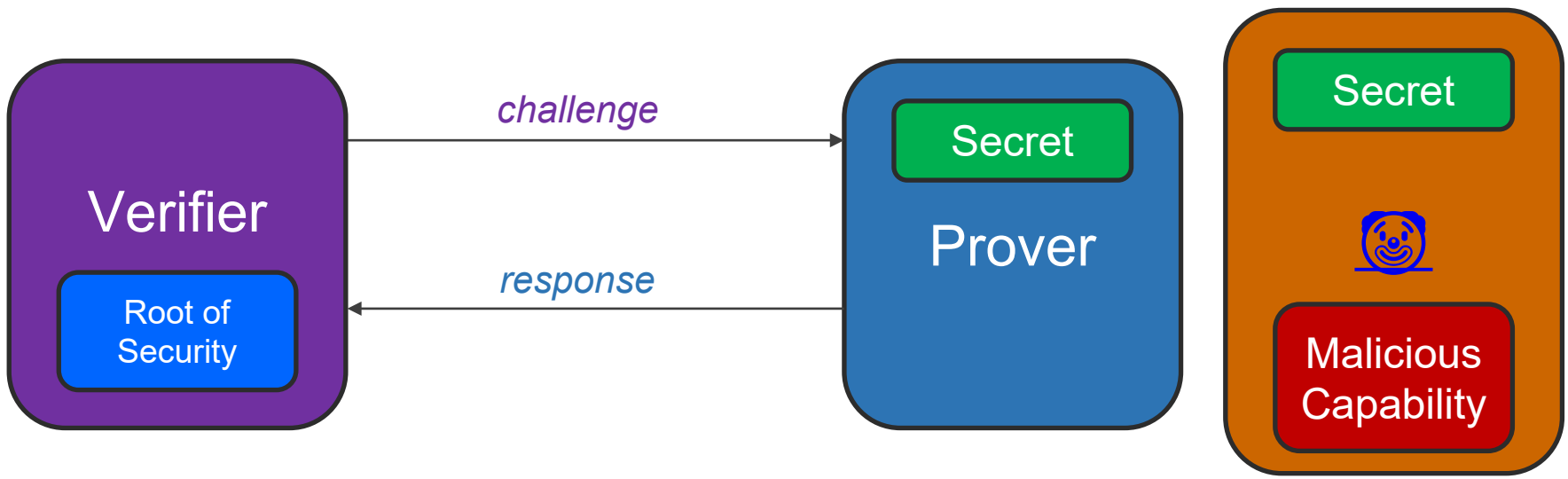
- Generally involves the concept of “challenge response”
  - An authentic chip has a secret that only an authentic chip should be able to prove that it knows



Challenge/Response is tricky

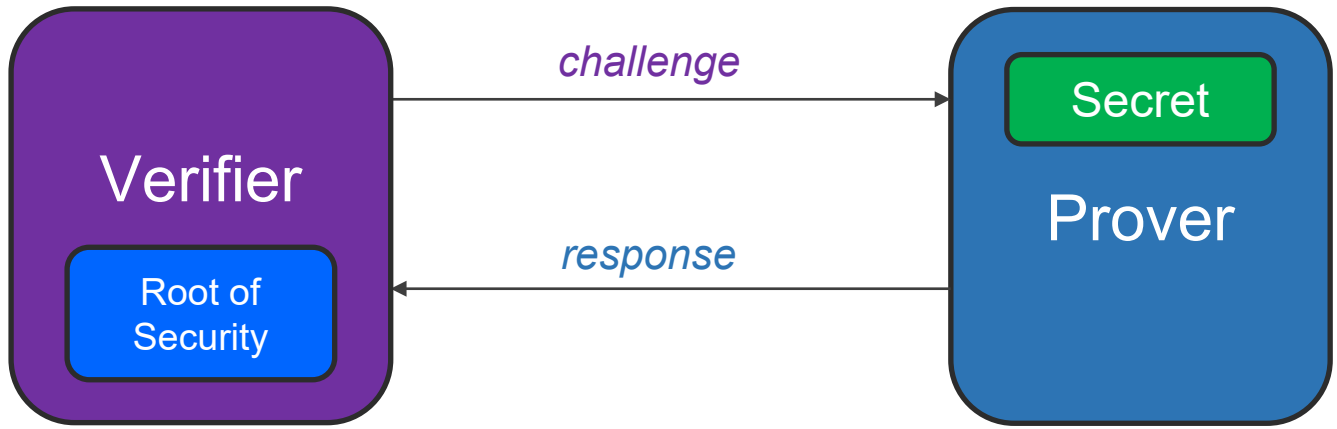
- Testing whether the chiplets in your SiP know the secrets they should know is a good start

- However … if an adversary can learn these secrets, they can manufacture a clone that impersonates authentic chiplets



How to protect on-chip secrets

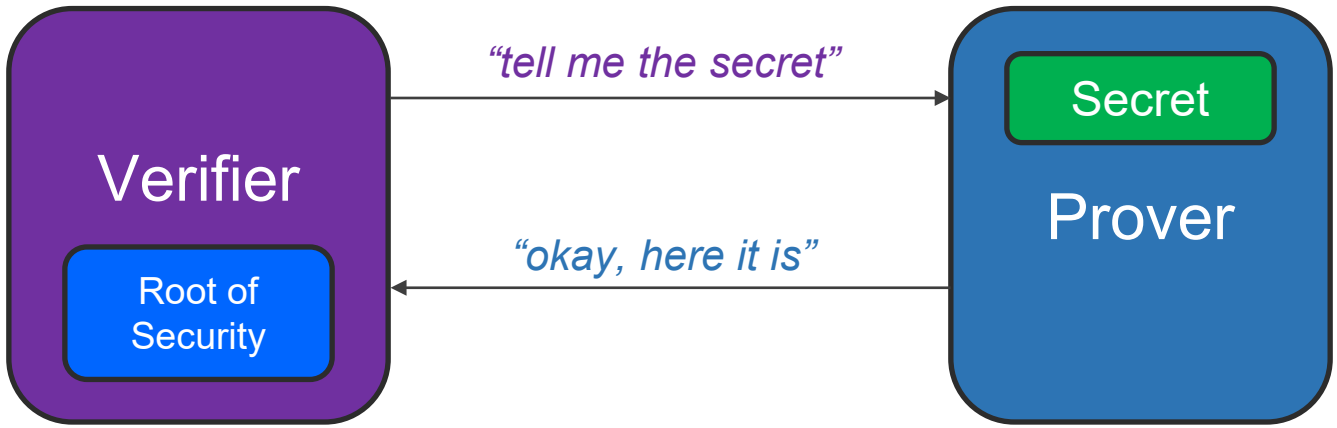
- First question: how should secrets be stored on chip?



- Ideally, the secret is split into several pieces (“keysplits”)
  - Some in the netlist (can be RE’d, but it’s difficult)
  - Some in the embedded NVM (easiest attack: re-enable mfg mode)
  - Maybe a PUF? Data disappears when the chip is powered off.
  - All of those, combined in a secure way, and only when needed

How to reveal on-chip secrets

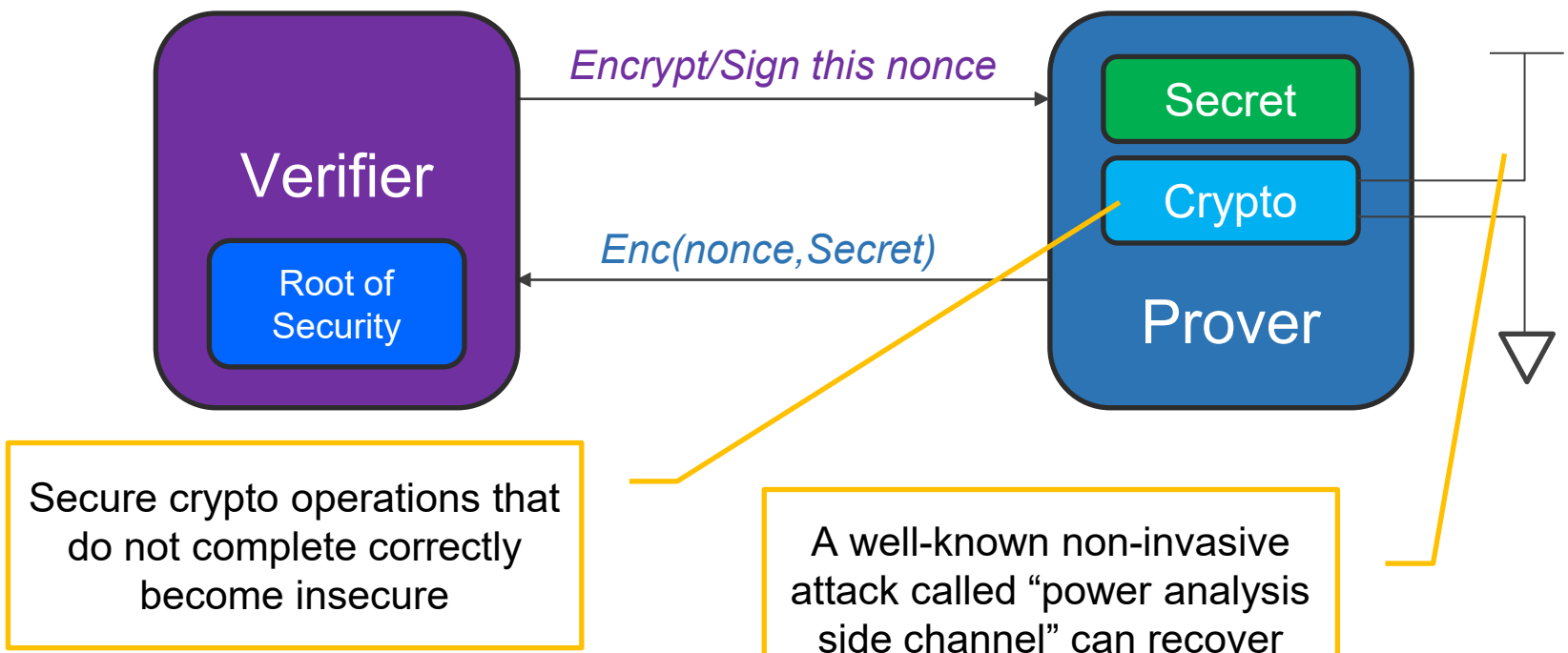
- Second question … what makes a good C/R protocol?
  - Not all challenge response protocols are good ones…



- Reminder: the main thing preventing a malicious clone of a chiplet is knowledge of the Secret value
  - Assume your adversary will collect ~1M C/R pairs to learn what you're doing, and (if possible) determine that Secret value

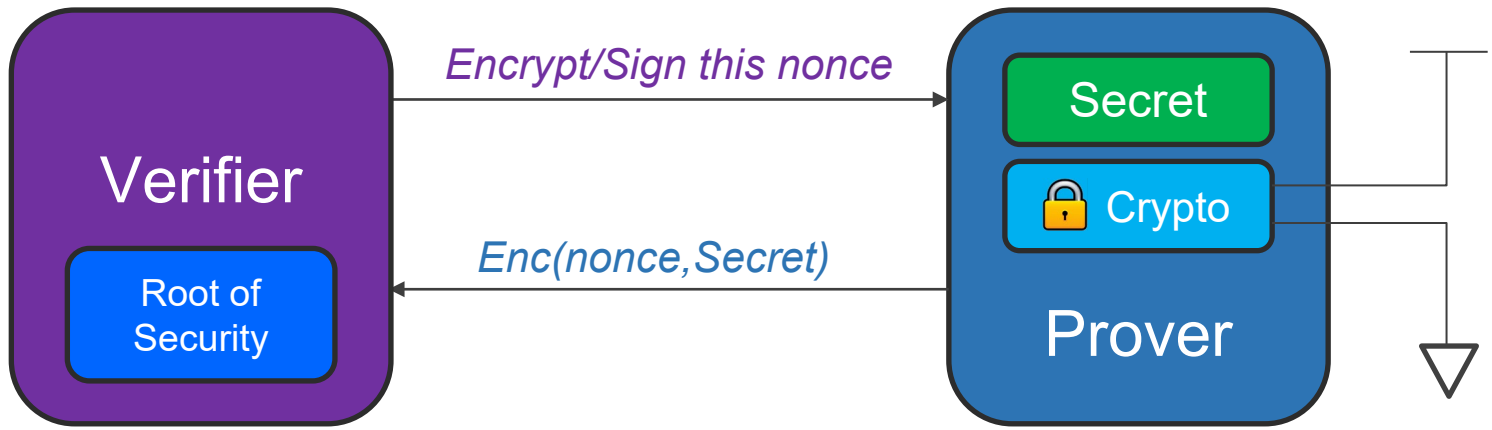
How to reveal *knowledge* of on-chip secrets

- Challenge/Response using crypto is (of course) a good idea, but everyday crypto can be attacked…



How to *safely* reveal knowledge of on-chip secrets

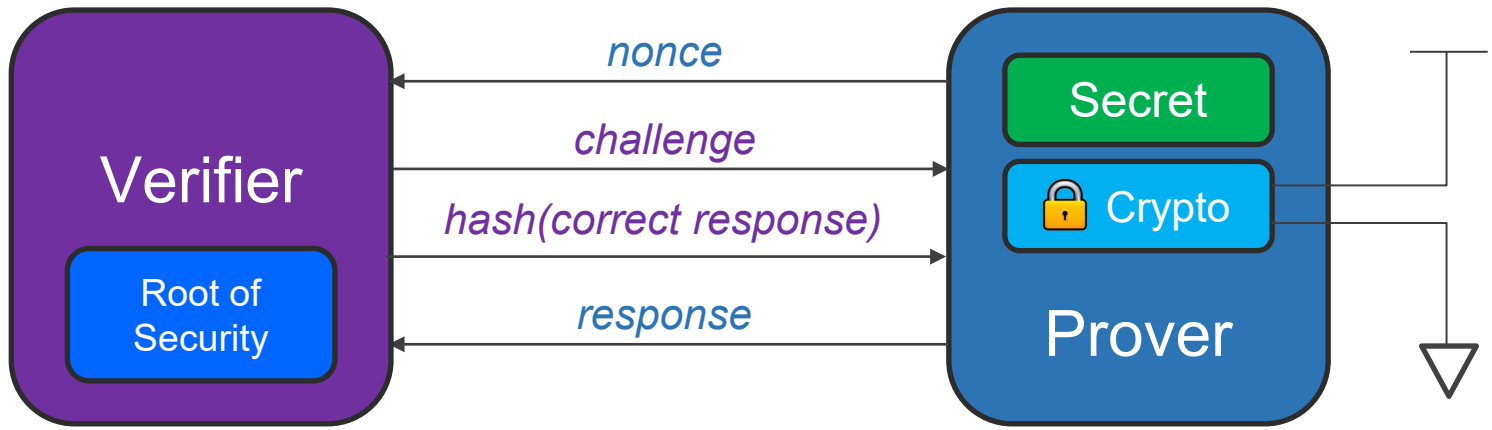
- What's needed for the prover is *tamper-resistant crypto*



- Tamper-resistance includes:
  - Countermeasures against power-analysis side channel
  - Countermeasures against fault attacks, both "glitch" and laser-fault
  - Countermeasures against "environmental attacks" (over/under voltage/clock)

Lastly … should the Prover trust the Verifier?

- Mutual authentication prevents “harvesting” of Prover



- The Verifier knows (at least something about) the correct response before Prover responds
  - Before the Prover releases the response, it waits until the Verifier sends proof that it knows it already

Summary

- High-mix heterogeneous SiPs are the ideal breeding ground for malicious “hardware trojan” chiplets
- One chiplet must be the Verifier, every other chip must be a Prover
- A Challenge/Response protocol is what binds Verifier and Prover
- Essential C/R ingredients:
  1. Many and varied keysplits
  2. Tamper-resistant crypto
  3. Mutual authentication