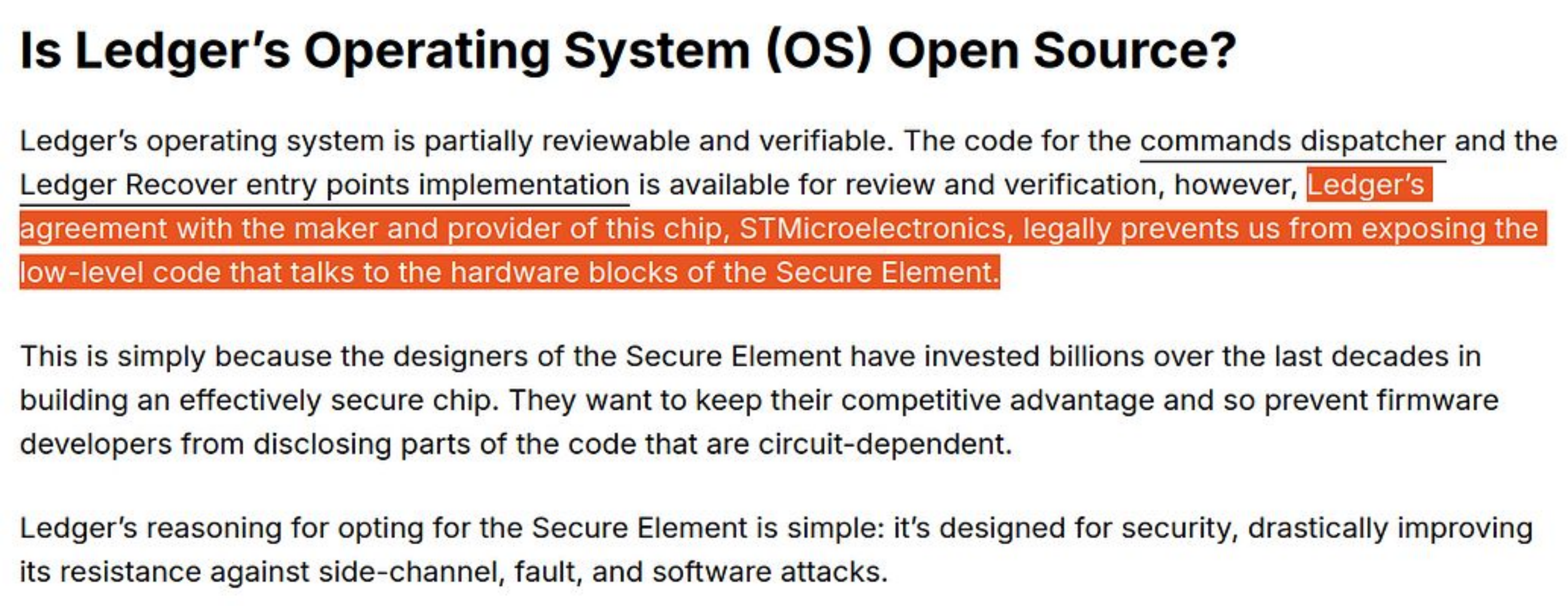
Secure Hardware

From Sand to Stone

Quintus Kilbourn - Flashbots Research



Is Ledger’s Operating System (OS) Open Source?

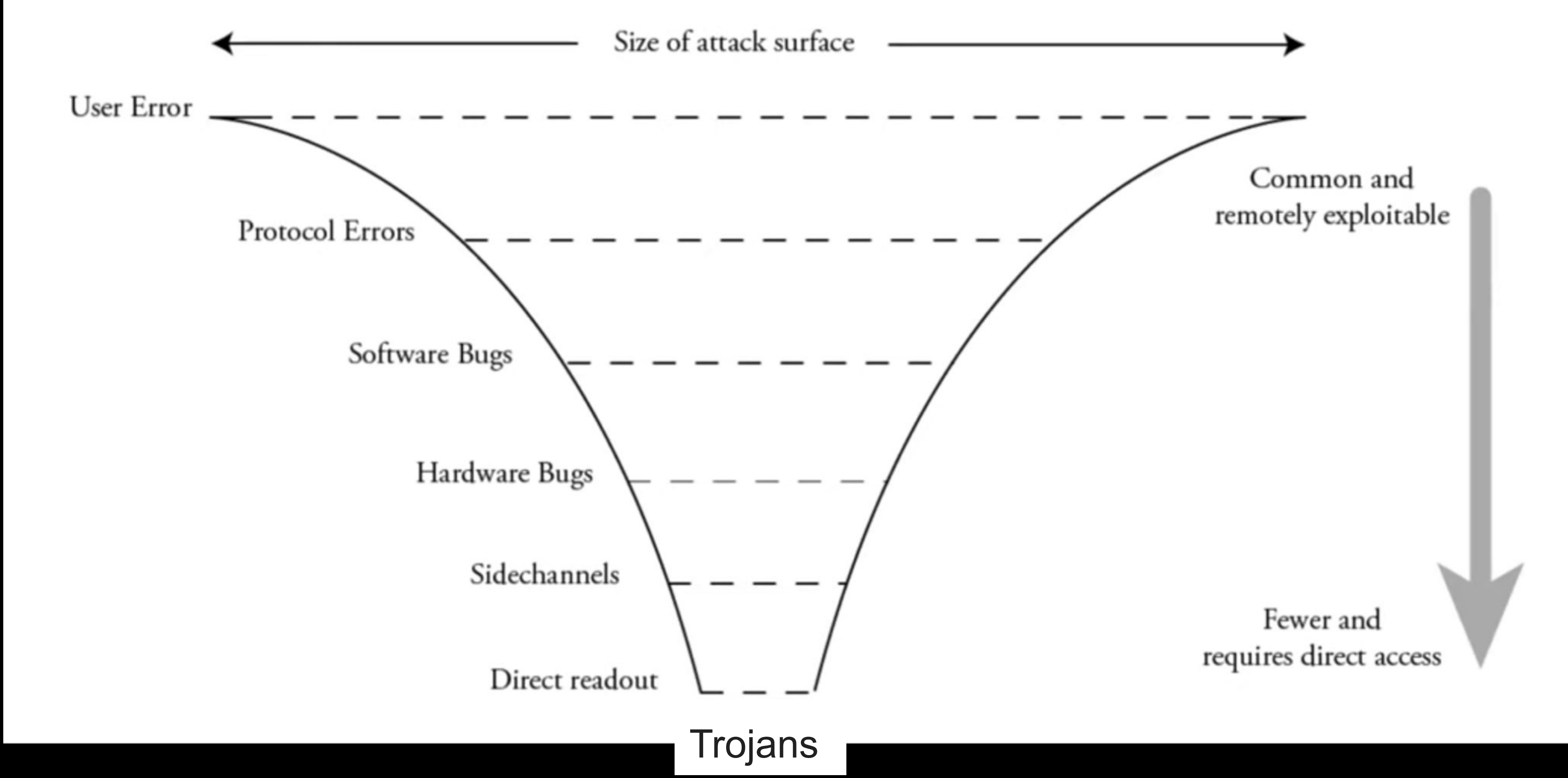
Ledger’s operating system is partially reviewable and verifiable. The code for the commands dispatcher and the Ledger Recover entry points implementation is available for review and verification, however, Ledger’s agreement with the maker and provider of this chip, STMicroelectronics, legally prevents us from exposing the low-level code that talks to the hardware blocks of the Secure Element.

This is simply because the designers of the Secure Element have invested billions over the last decades in building an effectively secure chip. They want to keep their competitive advantage and so prevent firmware developers from disclosing parts of the code that are circuit-dependent.

Ledger’s reasoning for opting for the Secure Element is simple: it’s designed for security, drastically improving its resistance against side-channel, fault, and software attacks.

Tamper Resistance
Performance
Cost

Trust Assumptions



Size of attack surface

User Error

Protocol Errors

Software Bugs

Hardware Bugs

Sidechannels

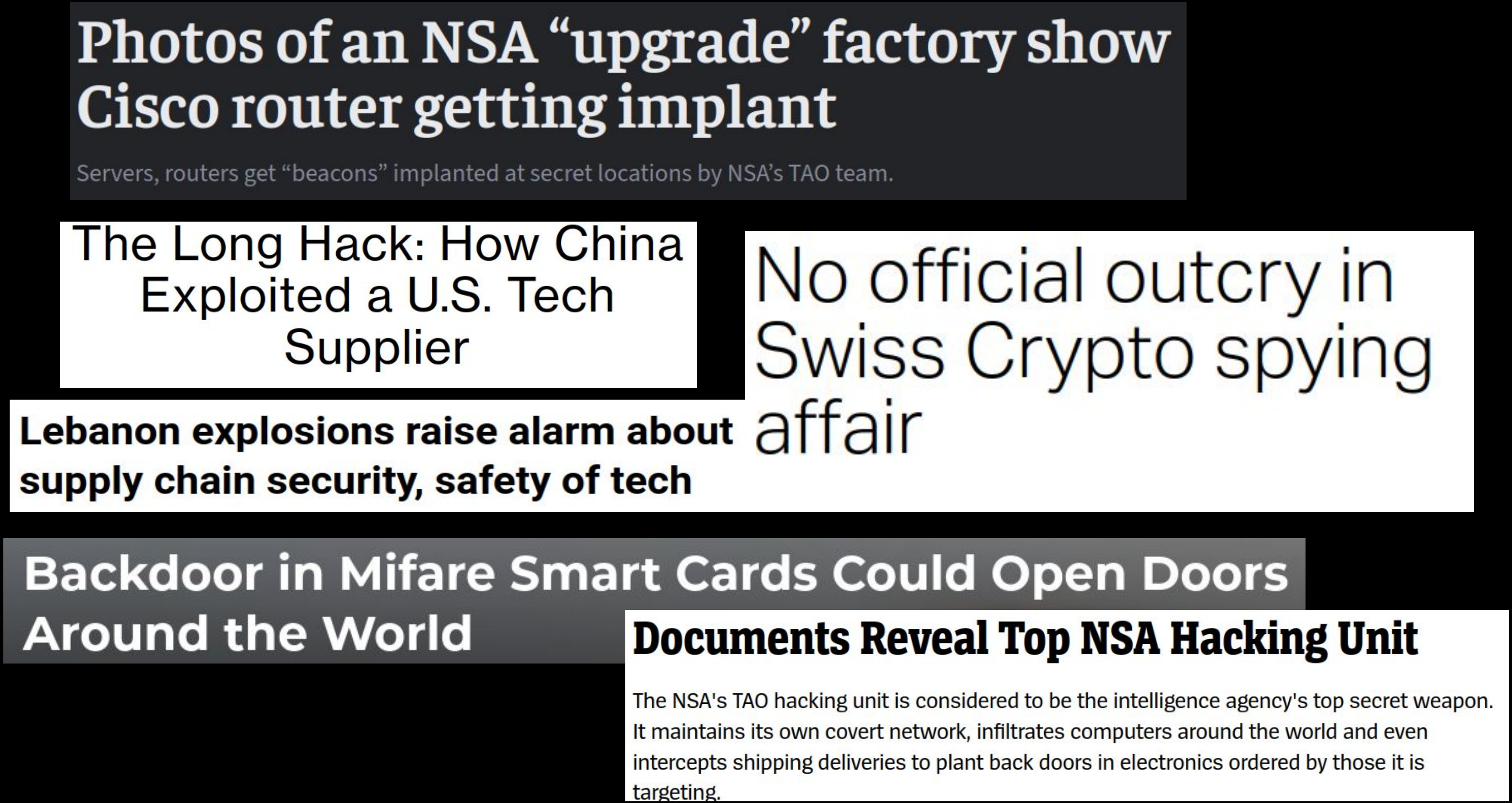
Direct readout

Trojans

Common and remotely exploitable

Fewer and requires direct access

Bunnie’s Talk At Silicon Salon III



Photos of an NSA "upgrade" factory show Cisco router getting implant

Servers, routers get "beacons" implanted at secret locations by NSA's TAO team.

The Long Hack: How China Exploited a U.S. Tech Supplier

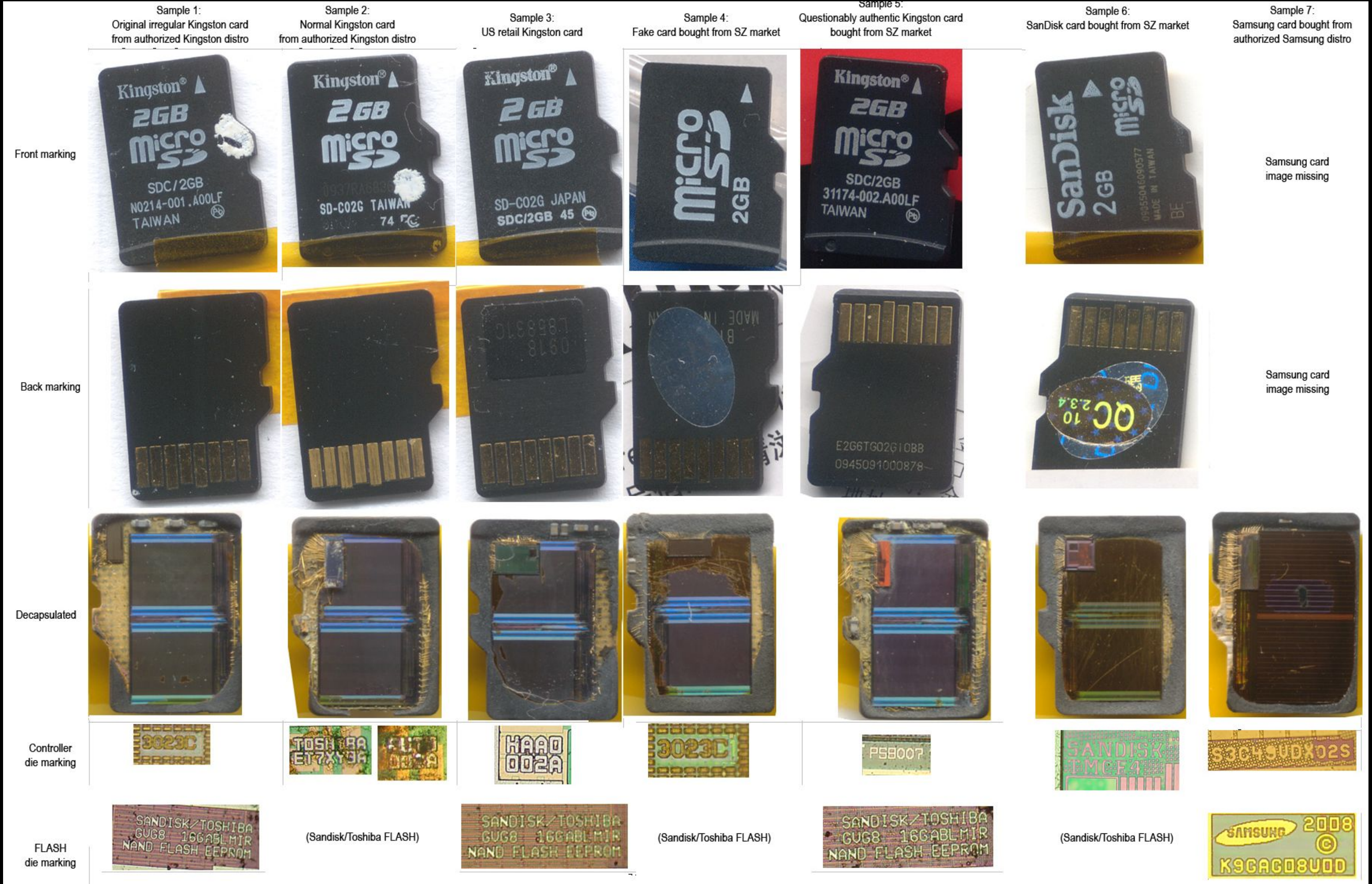
Lebanon explosions raise alarm about supply chain security, safety of tech

No official outcry in Swiss Crypto spying affair

Backdoor in Mifare Smart Cards Could Open Doors Around the World

Documents Reveal Top NSA Hacking Unit

The NSA's TAO hacking unit is considered to be the intelligence agency's top secret weapon. It maintains its own covert network, infiltrates computers around the world and even intercepts shipping deliveries to plant back doors in electronics ordered by those it is targeting.



Sample 1: Original irregular Kingston card from authorized Kingston distro

Sample 2: Normal Kingston card from authorized Kingston distro

Sample 3: US retail Kingston card

Sample 4: Fake card bought from SZ market

Sample 5: Questionably authentic Kingston card bought from SZ market

Sample 6: SanDisk card bought from SZ market

Sample 7: Samsung card bought from authorized Samsung distro

Front marking

Back marking

Decapsulated

Controller die marking

FLASH die marking

Samsung card image missing

Samsung card image missing

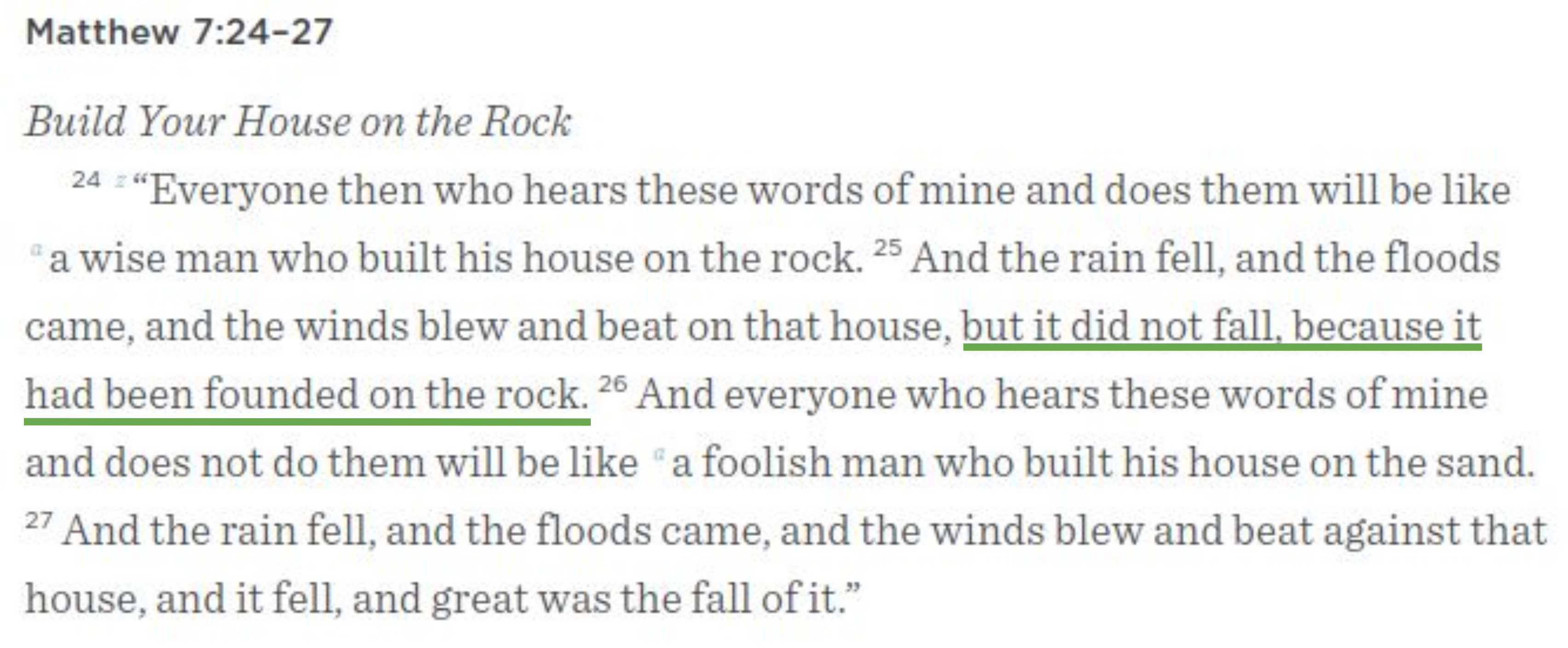
(Sandisk/Toshiba FLASH)

(Sandisk/Toshiba FLASH)

(Sandisk/Toshiba FLASH)

Kingston® 2GB microSD SDC/2GB N0214-001.A00LF TAIWAN
Kingston® 2GB microSD SD-C02G TAIWAN 74
Kingston® 2GB microSD SD-C02G JAPAN SDC/2GB 45
microSD 2GB
Kingston® 2GB microSD SDC/2GB 31174-002.A00LF TAIWAN
SanDisk 2GB microSD MADE IN TAIWAN BE
E2G6TG02G10BB
0945091000878
QC 10 2.3.4
3023C
TOSHIBA ET7XY9A
HAAD 002A
3023C
P5B007
SANDISK TMCF4
S3C49UDX02S
SANDISK/TOSHIBA GUG8 16GABLMIR NAND FLASH EEPROM
SANDISK/TOSHIBA GUG8 16GABLMIR NAND FLASH EEPROM
SANDISK/TOSHIBA GUG8 16GABLMIR NAND FLASH EEPROM
SAMSUNG 2008 © K9GAG08U0D

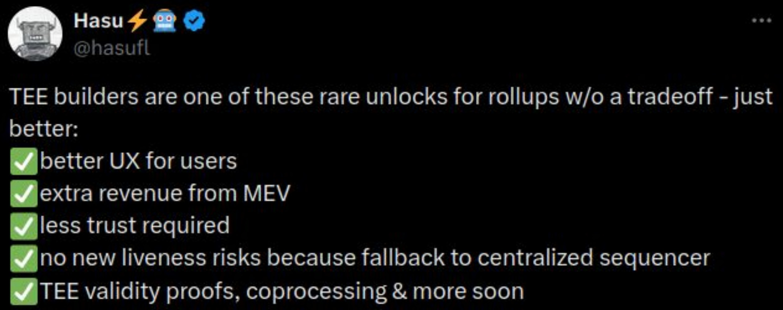
On MicroSD Porblems



Matthew 7:24–27

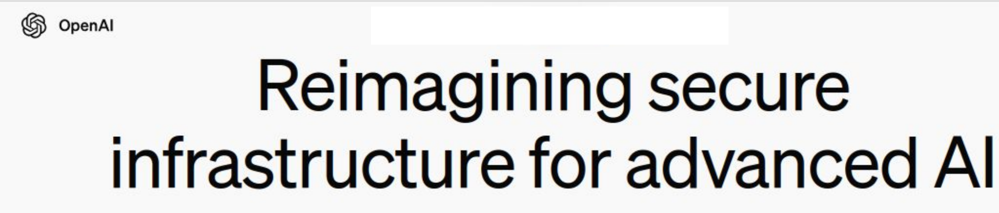
Build Your House on the Rock

[24] [z] "Everyone then who hears these words of mine and does them will be like [a] a wise man who built his house on the rock. [25] And the rain fell, and the floods came, and the winds blew and beat on that house, but it did not fall, because it had been founded on the rock. [26] And everyone who hears these words of mine and does not do them will be like [a] a foolish man who built his house on the sand. [27] And the rain fell, and the floods came, and the winds blew and beat against that house, and it fell, and great was the fall of it."



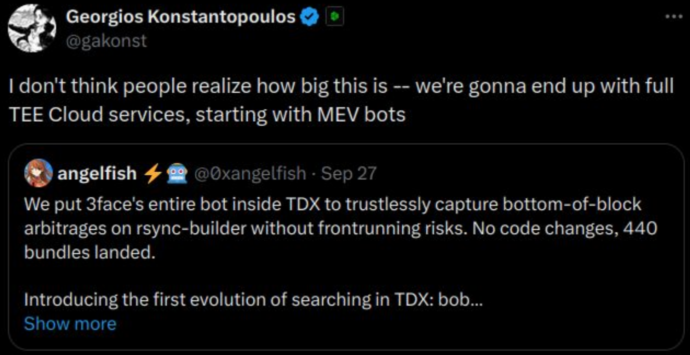
Hasu⚡🤖
@hasufl

TEE builders are one of these rare unlocks for rollups w/o a tradeoff - just better:
✅better UX for users
✅extra revenue from MEV
✅less trust required
✅no new liveness risks because fallback to centralized sequencer
✅TEE validity proofs, coprocessing & more soon



OpenAI

Reimagining secure infrastructure for advanced AI

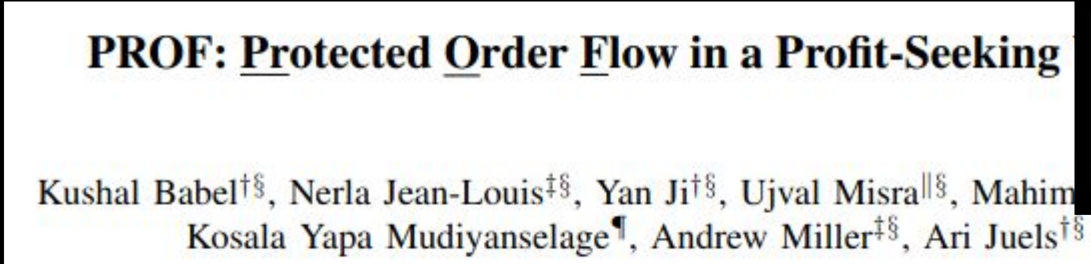


Georgios Konstantopoulos
@gakonst

I don't think people realize how big this is -- we're gonna end up with full TEE Cloud services, starting with MEV bots

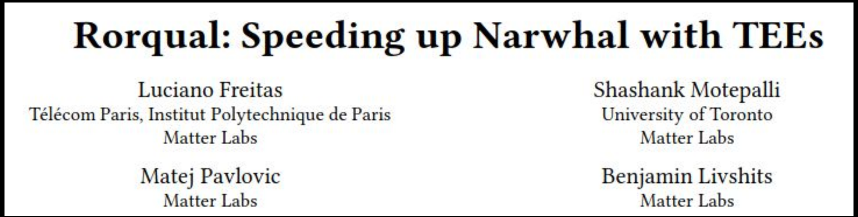
angelfish⚡🤖 @0xangelfish · Sep 27
We put 3face's entire bot inside TDX to trustlessly capture bottom-of-block arbitrages on rsync-builder without frontrunning risks. No code changes, 440 bundles landed.

Introducing the first evolution of searching in TDX: bob...
Show more



PROF: Protected Order Flow in a Profit-Seeking

Kushal Babel[†§], Nerla Jean-Louis[‡§], Yan Ji[†§], Ujval Misra[‖§], Mahim
Kosala Yapa Mudiyanselage[¶], Andrew Miller[‡§], Ari Juels[†§]



Rorqual: Speeding up Narwhal with TEEs

Luciano Freitas
Télécom Paris, Institut Polytechnique de Paris
Matter Labs

Shashank Motepalli
University of Toronto
Matter Labs

Matej Pavlovic
Matter Labs

Benjamin Livshits
Matter Labs

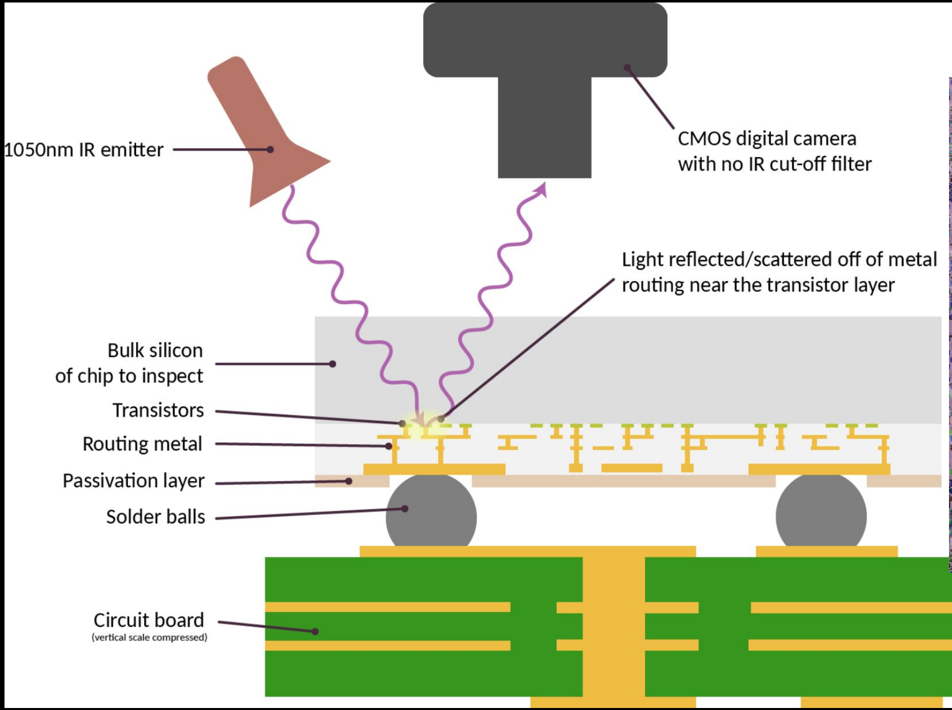
Image: How do we analyse a chip?

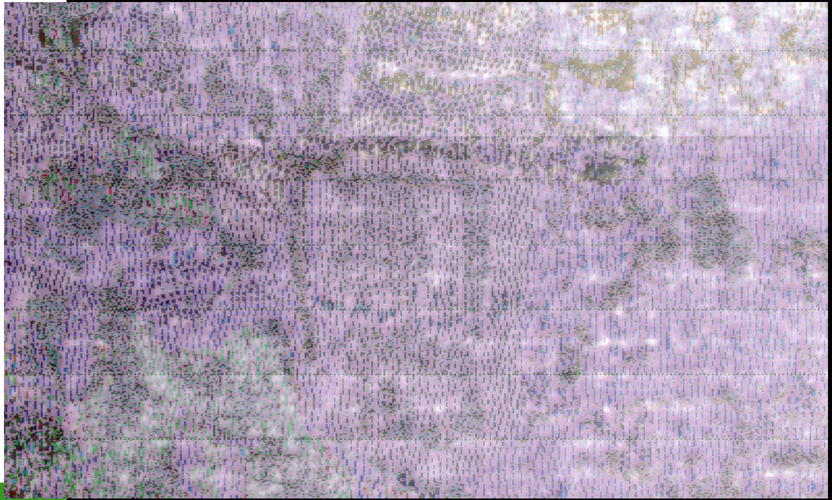
Reference: What does a secure chip look like?

Image: How do we analyse a chip?

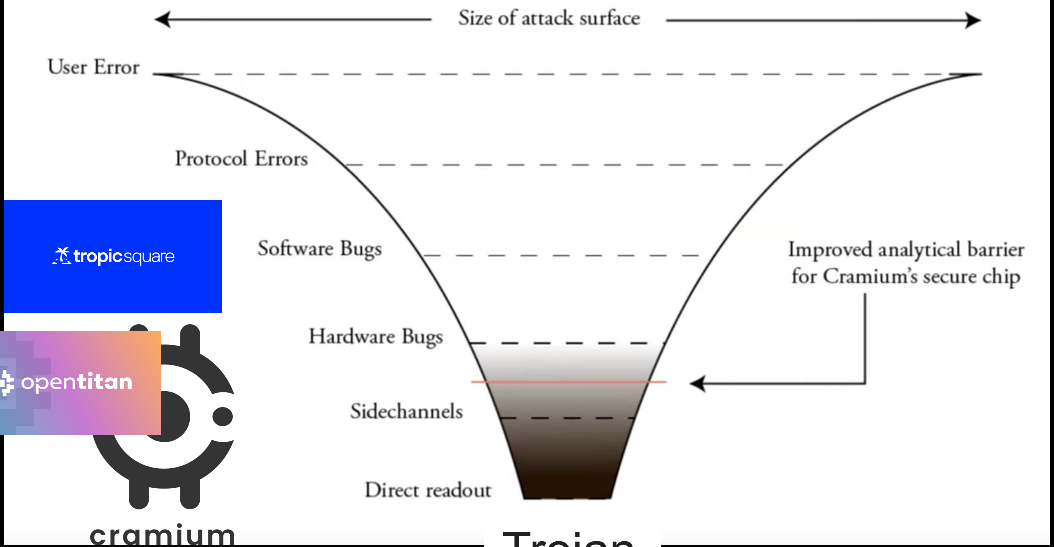
Infra-Red, *In-Situ* (IRIS) Inspection of Silicon

Andrew ‘bunnie’ Huang





Reference: What does a secure chip look like?



Size of attack surface

User Error

Protocol Errors

Software Bugs

Hardware Bugs

Sidechannels

Direct readout

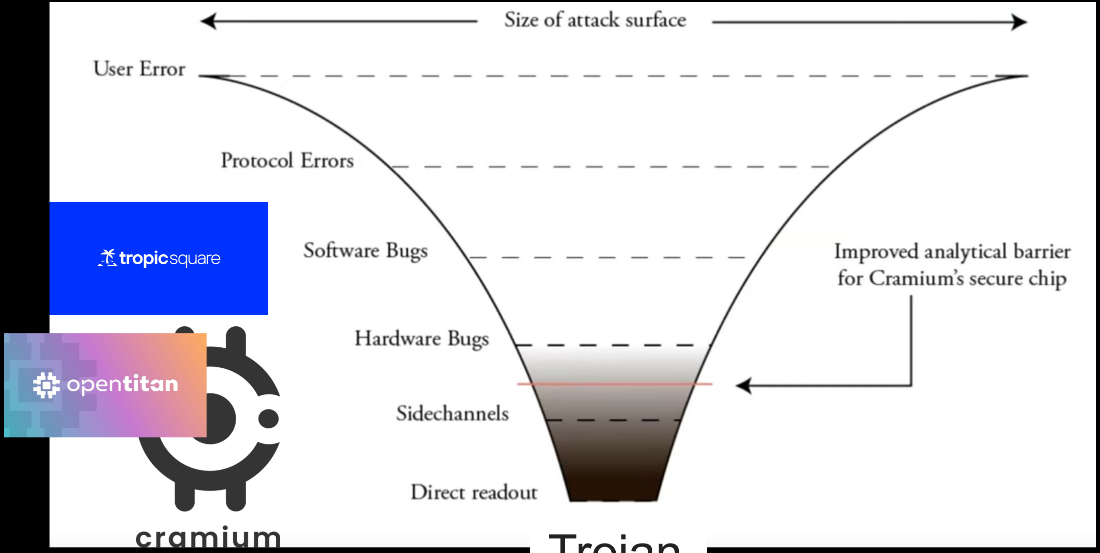
Improved analytical barrier for Cramium's secure chip

tropicsquare

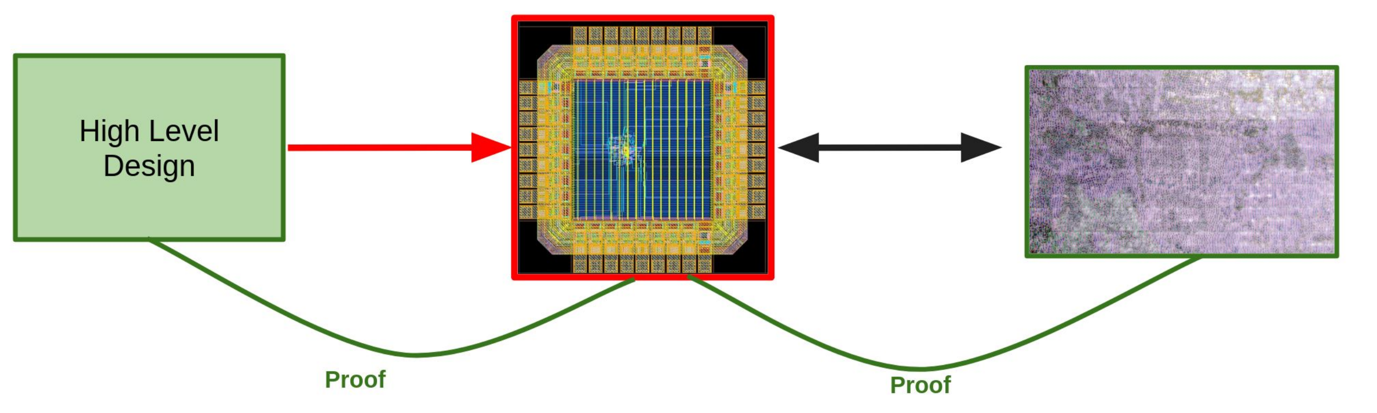
opentitan

cramium

Trojans


Size of attack surface
User Error
Protocol Errors
Software Bugs
Hardware Bugs
Sidechannels
Direct readout
Improved analytical barrier for Cramium's secure chip
tropicsquare
opentitan
cramium

Trojan


High Level Design
Proof
Proof

Easy case

Image: How do we analyse a chip?

Reference: What does a secure chip look like?

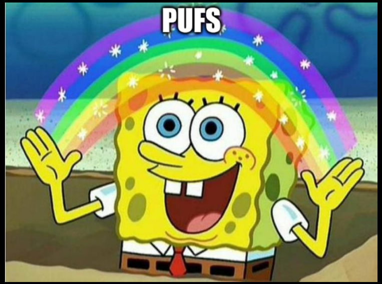
- RA
  - Easy case
    - **Image**: How do we analyse a chip?
    - **Reference**: What does a secure chip look like?
  - **Key**: How do we know hardware keys are not compromised?
  - **Attestation**: How do we remove trust in remote attestation services?

RA

Easy case

Image: How do we analyse a chip?

Reference: What does a secure chip look like?



Key: How do we know hardware keys are not compromised?

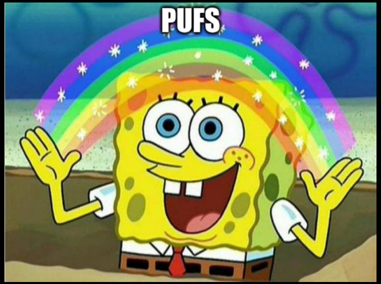
Attestation: How do we remove trust in remote attestation services?

RA

Easy case

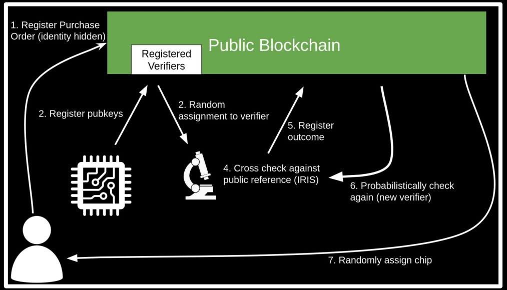
Image: How do we analyse a chip?

Reference: What does a secure chip look like?



Key: How do we know hardware keys are not compromised?

Attestation: How do we remove trust in remote attestation services?



Public Blockchain

Registered Verifiers

1. Register Purchase Order (identity hidden)

2. Register pubkeys

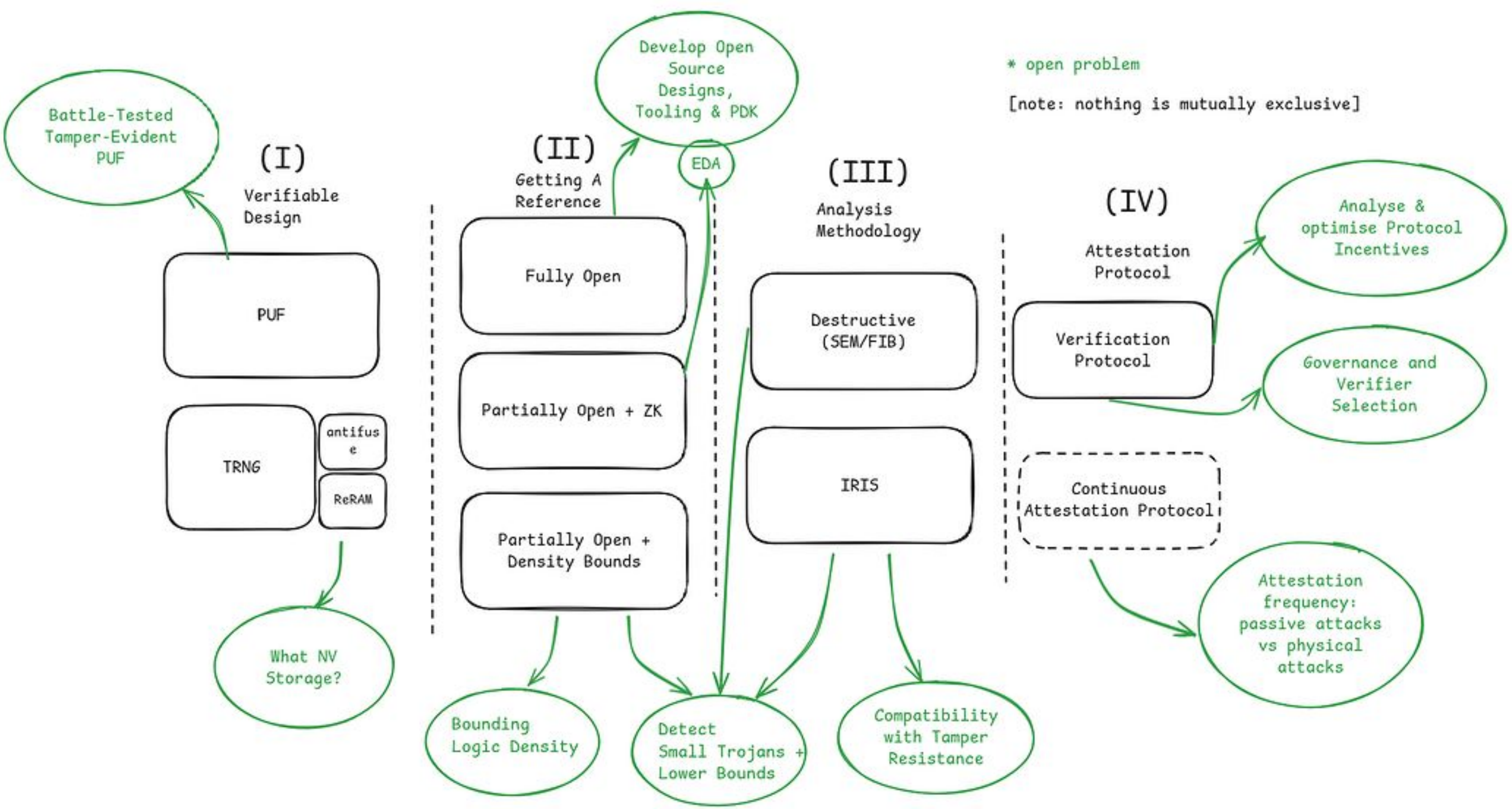
2. Random assignment to verifier

4. Cross check against public reference (IRIS)

5. Register outcome

6. Probabilistically check again (new verifier)

7. Randomly assign chip



* open problem

[note: nothing is mutually exclusive]

(I)
Verifiable Design

PUF

TRNG

antifuse

ReRAM

Battle-Tested Tamper-Evident PUF

What NV Storage?

(II)
Getting A Reference

Fully Open

Partially Open + ZK

Partially Open + Density Bounds

Develop Open Source Designs, Tooling & PDK

EDA

Bounding Logic Density

Detect Small Trojans + Lower Bounds

(III)
Analysis Methodology

Destructive (SEM/FIB)

IRIS

Compatibility with Tamper Resistance

(IV)
Attestation Protocol

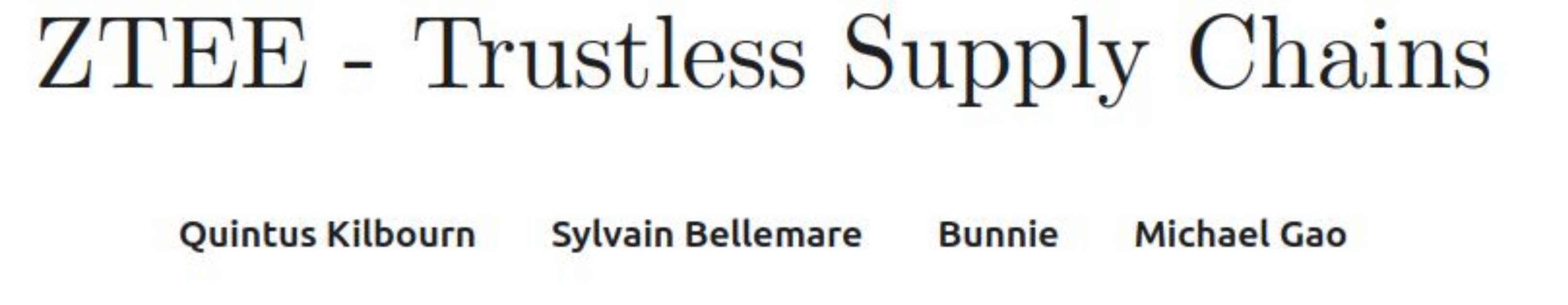
Verification Protocol

Continuous Attestation Protocol

Analyse & optimise Protocol Incentives

Governance and Verifier Selection

Attestation frequency: passive attacks vs physical attacks



ZTEE - Trustless Supply Chains

Quintus Kilbourn Sylvain Bellemare Bunnie Michael Gao

*with a lot of help from Thorben Moos