Hardware for Secure Autonomy

Tanvir Arafin

August 1, 2022

Morgan State University
Baltimore, MD

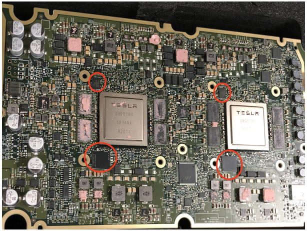
Overview

1. Hardware Security & Autonomous Systems
2. Case Study I: Hardware Root of Trust
3. Case Study II: Physically (Un)cloneable Functions
4. Case Study III: Accelerators for Security
5. Future Research Directions

Hardware Security & Autonomous Systems

Smart Yet Vulnerable Hardware

Smart Yet Vulnerable Hardware



Subaru Cockpit [Image https://www.subaru.com/vehicles/outback/gallery.html]

Tesla Cockpit [Image https://www.tesla.com/tesla-gallery, Courtesy of Tesla, Inc.]

Hardware Security & Smart Systems

- Firmware Extraction
- Architectural Vulnerability Exploitation
- Side-channel Analysis
- Fault Injection

My Research

Hardware Security

- Security is a *full-stack*, *cross-layered* problem
- Hardware: the weakest link

My Research

Hardware Security

- Security is a *full-stack*, *cross-layered* problem
- Hardware: the weakest link
- **Hardware: the strongest link**

Autonomous Systems

Mechanized systems →Automated systems →Autonomous systems

Key Idea

Intelligent machines to sense, plan and act in a changing environment

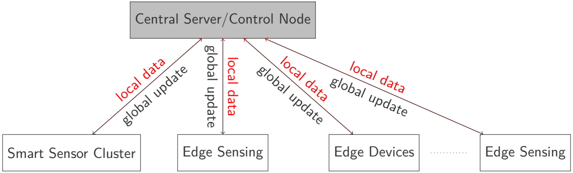


Figure: A simplified system architecture common in autonomous systems

Case Study I: Hardware Root of Trust

Problem

Question

How to verify data at the edge?

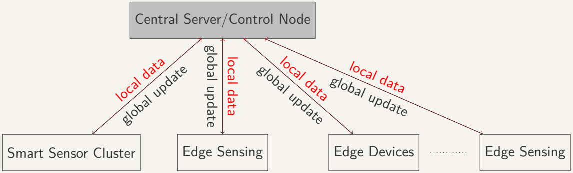
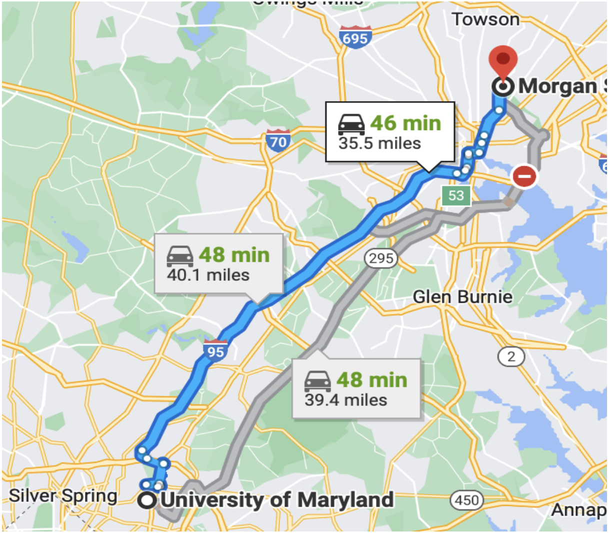


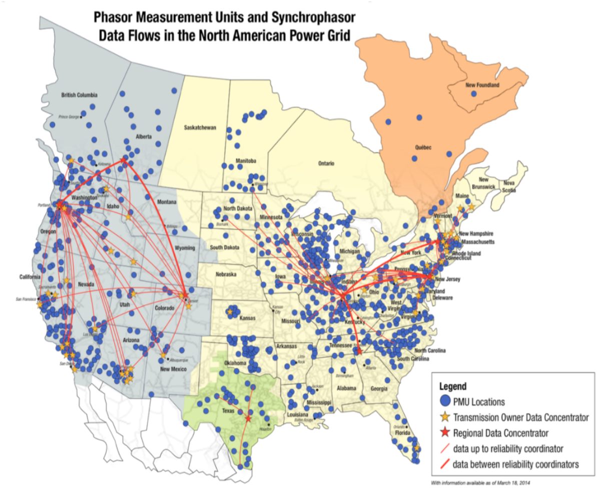
Figure: A simplified system architecture common in autonomous systems

Navigation



Is it secure?

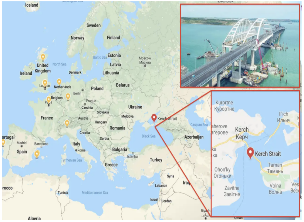
Synchronization in Smart Grid



Source: North American SynchroPhasor Initiative

GPS Spoofing: Evidence

Crimea, 2021



PokeMon GO, 2016

White Rose, 2013

Lockheed RQ-170, 2013

Russia spoofed AIS data. Source://www.theregister.com/2021/06/24/russia_ais_spoofing/

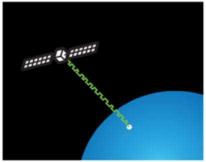
GPS Spoofing: Basics

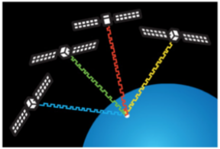
True receiver-to-satellite distance

$$r_{true} = c\ t_{propagation} = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2 + (z_t - z_r)^2} \quad (1)$$

$$r_{pseudo} = r_{true} - ct_r \quad (2)$$

$$t_{sync} = t_{local} + t_r \quad (3)$$





Synchronize transmitter and receiver clocks to calculate $t_{propagation}$

GPS Spoofing Detection: Strategy

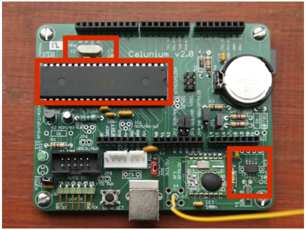
Key Idea

Cross-validate with “something true” or trusted (root of trust)

→Local Clock

Arafin, Anand, & Qu, GLSVLSI 2017. A low-cost GPS spoofing detector design for internet of things (IOT) applications. p 161. [Best Paper Nomination]

Crystal Oscillators



- Ubiquitous

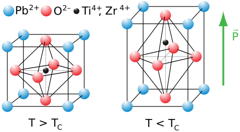
  Piezo-electric quartz crystal

- Intrinsically Unclonable

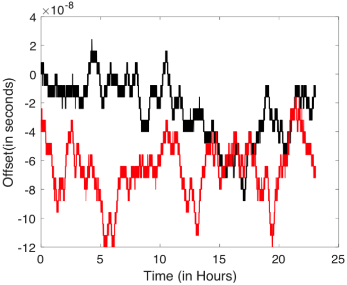
  Imperfect cutting →cutting variations →Physically unclonable time offset

- Reliable

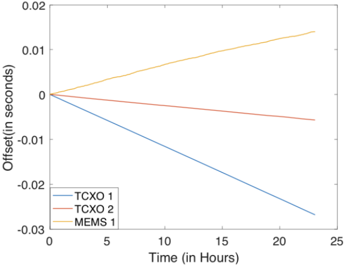
  TCXOs →Correct timing with temperature variation



Fault in Our Clocks



Clock offset between two GPS clocks



Clock offset for TCXO and MEMS clocks

Spoofing Detection

Key Idea

Measure drift (unclonable) against the received GPS signal (untrusted) to detect spoofing

Modeling a Clock

State Space Model

$$\mathbf{X_n} = \mathbf{F_n X_{n-1}} + \mathbf{W_n} \tag{4}$$

$$\xi_\mathbf{n} = \mathbf{H_n X_n} + \mathbf{V_n} \tag{5}$$

Clock state	$X = [x, y, D]$
Time offset	x
Frequency offset	y
Frequency drift	D
State transition matrix	F
Process noise	W

Results: Meaconing and Replay Attack

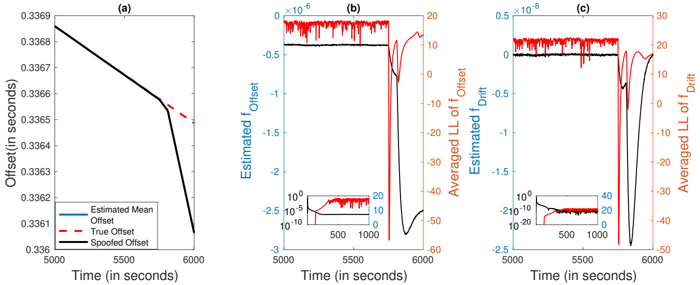


Figure: (a) Spoofing attack at 5130 seconds (b) Estimation of the frequency offset (black curve) and the LL of the frequency offset(red curve) and (c) Estimation of the frequency drift and the LL of the frequency drift.

Arafin, Anand, & Qu, GLSVLSI 2017. A low-cost GPS spoofing detector design for internet of things (IOT) applications. p 161. [Best Paper Nomination]

[Joint work with NIST]

Results: Pose Validation

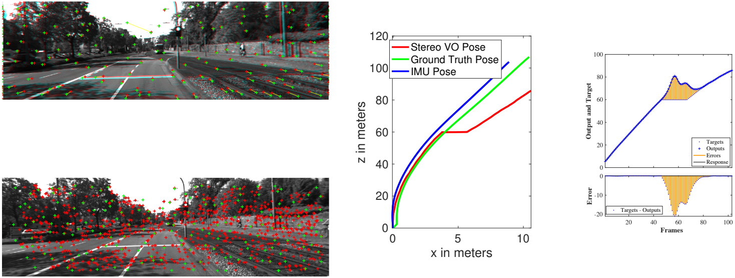


Figure: Flow matching (Left top) and feature selection(left bottom) for stereo-visual odometry. A replay attack on the camera input. Spoofed data on both of the stereo sensors for 20 frames, which results in the large deviation of the stereo odometry pose (red line) from the ground truth (green line).

Arafin, & Kornegay, CISS 2021. Attack Detection and Countermeasures for Autonomous Navigation. p. 1.

Case Study II: Physically (Un)cloneable Functions

Problem

Question

How does a central authority authenticate the client devices or processes and vice-versa?

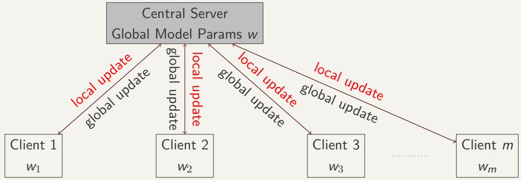


Figure: A simplified system architecture for federated learning.

Solution

PUFs

Physically uncloneable functions to authenticate devices

Issues

- Needs additional circuits
- Power & area constraints

Solution: Voltage Overscaling

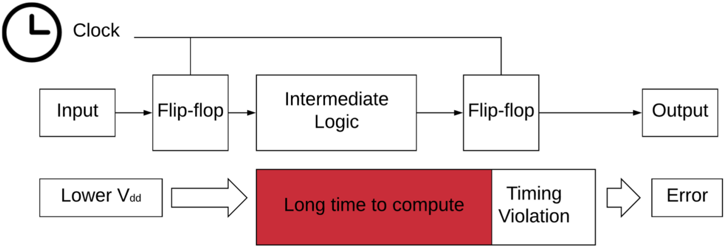
Key Idea

Extract information about the process variation from a physical system using extreme operating condition

Voltage Scaling

- Power Consumption $P = C_{eff} V_{dd}^2 f + V_{dd}(I_{sub} + I_{gate})$
- Critical Voltage
- Scaling Below Critical Voltage →Error due to path delay

How do Faults Occur?

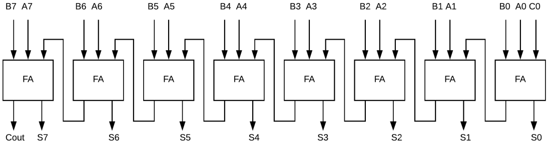


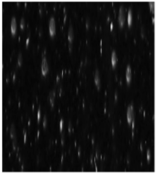
$$\sigma_{\Delta V_t} = A_{\Delta V_t} / \sqrt{WL}$$

$$d_{gate} \propto \frac{V_{DD}}{\beta (V_{DD} - V_t)^{\alpha}}$$

Example

Ripple Carry Adder (45nm)





Example

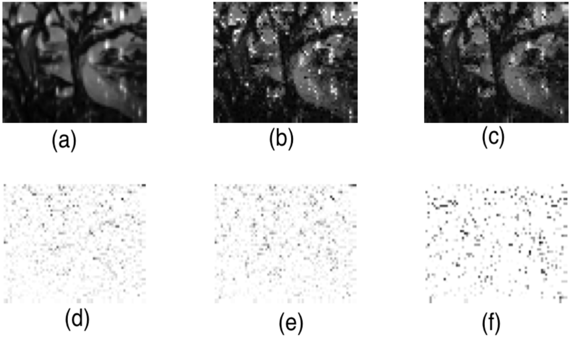


Figure: (a) Vdd = 1V, Adder A and B; (b) Vdd = 0.4V, Adder A; (c)Vdd = 0.4V, Adder B; (d), (e), and (f) Comparison between (a)-(b), (a)-(c) and (b)-(c)

Arafin, & Qu, ASP-DAC 2017. VOLtA: Voltage over-scaling based lightweight authentication for IoT applications. p. 336.

Zhang, Shen, Su, **Arafin**, & Qu, IEEE TC 2021. Voltage over-scaling-based lightweight authentication for IoT security. p. 323. [Featured Paper of the Month]

Single Round Interactive Authentication

Prover$(\mathbf{x_1}, \mathbf{x_2}, H)$

Verifier$(M, \mathbf{x_1}, \mathbf{x_2}, \epsilon)$

$\mathbf{R} \overset{\$}{\leftarrow} \mathbb{Z}_p^{\ell \times n}$

$\xleftarrow{\mathbf{R}}$

Calculate $\mathbf{L} = \mathbf{H}(\mathbf{R}, \mathbf{x_1}) = \mathbf{R} + \mathbf{x_1}$ using the adder and then calculate $\mathbf{z} = \mathbf{L} \oplus \mathbf{x_2} = (\mathbf{R} + \mathbf{x_1}) \oplus \mathbf{x_2}$

$\xrightarrow{\mathbf{z}}$

Calculate $\mathbf{z}' = M(\mathbf{R}, \mathbf{x_1}) \oplus x_2$. If distance $(\mathbf{z}', \mathbf{z}) \leq \epsilon$ accept.

Zhang, Shen, Su, **Arafin**, & Qu, IEEE TC 2021. Voltage over-scaling-based lightweight authentication for IoT security. p. 323. [Featured Paper of the Month]

Case Study III: Accelerators for Security

Accelerators for Security

Question

Can we move privacy-preserving computations at the sensor edge (i.e., near-pixel, near-memory computation)?

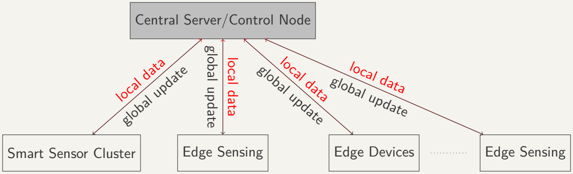
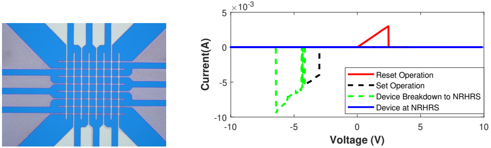


Figure: A simplified system architecture common in autonomous systems

Cryptography Using Memory Devices

Key Idea

Emerging memory device can perform logic and arithmetic computation.



x (Incoming Bit)	y (Resistive State)	z (Memory Content)
0	0(NRHRS)	0
0	1(Writeable)	0
1	0(NRHRS)	0
1	1(Writeable)	1

Figure: Fabricated device, Sample I-V curve for the SET/RESET operation and hard breakdown, and the truth table.

Arafin, Shen, Tehranipoor & Qu, GLSVLSI 2019. LPN-based Device Authentication Using Resistive Memory. p 9.

Cryptography Using Memory Devices

Key Idea

Simple error correction technique (i.e., parity) can lead to lightweight yet quantum resistant cryptography (LPN, LWE, etc).

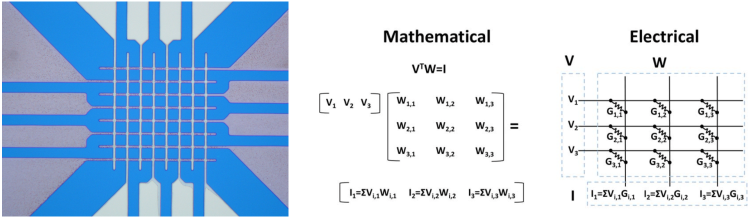

Mathematical
$V^TW=I$
$\begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix} \begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} =$
$\begin{bmatrix} I_1=\Sigma V_{i,1}W_{i,1} & I_2=\Sigma V_{i,2}W_{i,2} & I_3=\Sigma V_{i,3}W_{i,3} \end{bmatrix}$
Electrical
V
W
V_1
V_2
V_3
$G_{1,1}$ $G_{1,2}$ $G_{1,3}$
$G_{2,1}$ $G_{2,2}$ $G_{2,3}$
$G_{3,1}$ $G_{3,2}$ $G_{3,3}$
I
$I_1=\Sigma V_{i,1}G_{i,1}$ $I_2=\Sigma V_{i,2}G_{i,2}$ $I_3=\Sigma V_{i,3}G_{i,3}$

Figure: Fabricated device and basic matrix-vector computation

Arafin, Shen, Tehranipoor & Qu, GLSVLSI 2019. LPN-based Device Authentication Using Resistive Memory. p 9.

Energy-Efficient In-Memory Architecture for Cryptography

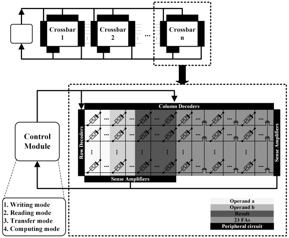


Figure: Implementation of a RIME computation unit

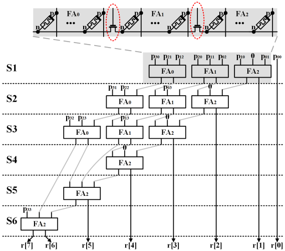

FA₀
FA₁
FA₂
p
n
S1
S2
S3
S4
S5
S6
p_{30} p_{21} p_{12} p_{20} p_{11} p_{02} p_{10} 0 p_{01} p_{00}
p_{31} p_{22} p_{03} 0
p_{32} p_{23} p_{13} 0
0
p_{33}
r[7] r[6] r[5] r[4] r[3] r[2] r[1] r[0]

Figure: Implementation of a 4-bit Wallace-tree multiplier in RIME.

Lu, **Arafin**, & Qu, ASP-DAC 2021. RIME: A scalable and energy-efficient processing-in-memory architecture for floating-point operations. p. 120.

Energy-Efficient In-Memory Architecture for Cryptography

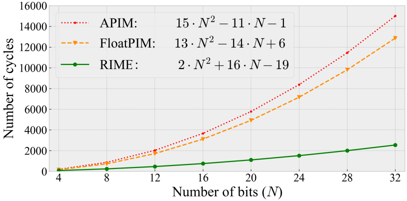


Figure: Latency of *N*-bit fixed-point multiplier.

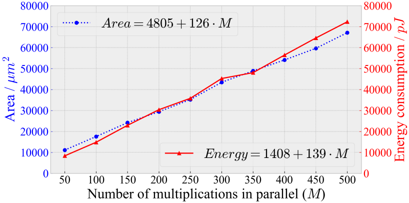
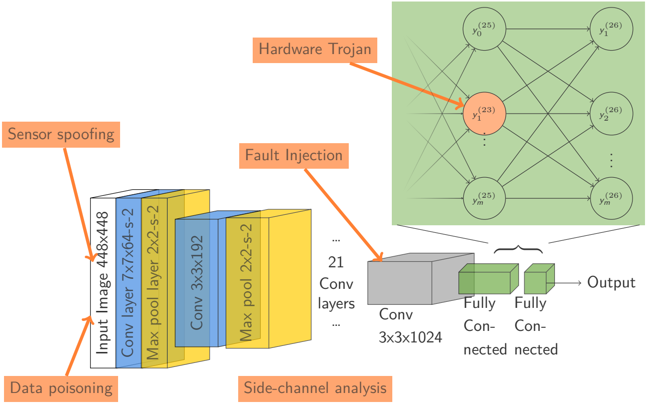


Figure: Area / μm^2 & energy consumption / *pJ* for a single 32-bit floating-point multiplier

Lu, **Arafin**, & Qu, ASP-DAC 2021. RIME: A scalable and energy-efficient processing-in-memory architecture for floating-point operations. p. 120.

Future Research Directions

Hardware Security of AI/ML Tools

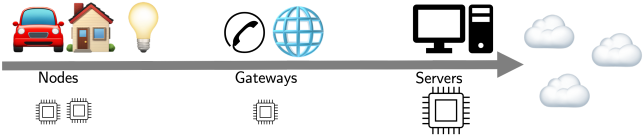


Xu, **Arafin**, Qu, ASP-DAC 2021, *Hardware Security of neural networks from hardware perspective: A survey and beyond*

YOLO v1 [**CVPR16.Redmon.YOLO**].

Funded by ARLIS

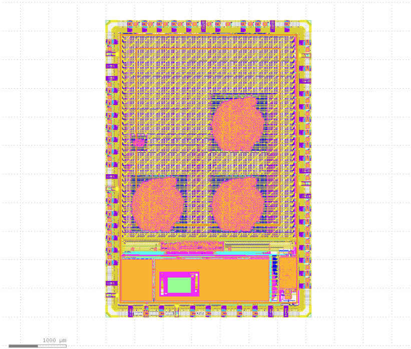
Hardware Security: From Edge to Cloud



Funded by NCAE-C

Opportunities in Data-centric Hardware Accelerators

- Quantum Resistant Algorithms & Hardware Accelerators
- Security Challenges of Processing-In-Memory Systems
- Scalable & Energy-Efficient In Memory Computation



Fabrication support by Apple

Long Term Vision

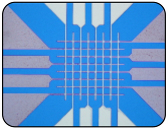
- Device level

  Security from nano-electronic device primitives
- Architecture level

  Secure hardware-software co-design
- System level

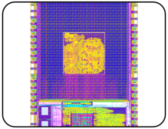
  Hardware vulnerabilities in critical embedded systems

Contributions



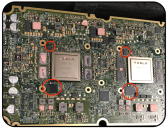
Device and Circuits

- PUFs [TC 2021 🏆, ASP-DAC 2017, ICCAD 2015]
- Approximate Computing [Computer 2017, GLSVLSI 2017 🥈]
- Supply Chain Integrity [ISCAS 2017]



Architecture

- Accelerators [ASPDAC 2021, SOCC 2020, GLSVLSI 2019]
- In-memory Computation [ASPDAC 2022, TVLSI 2018]
- Vulnerability [GLSVLSI 2020]



Systems

- ROT [CISS 2021, IOTSMS 2020, ASIAN-HOST 2018 🏆]
- ML Security [ASPDAC 2021, ASIAN-HOST 2020]
- Hardware Reverse Engineering

THANK
YOU

QUESTIONS
COMMENTS