USC Viterbi
School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

Secure and Trustworthy Cyber-Physical System Design: A Cross-Layer Perspective

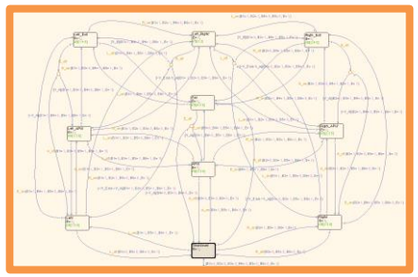
Pierluigi Nuzzo
Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California, Los Angeles
nuzzo@usc.edu

In Honor of Alberto Sangiovanni-Vincentelli

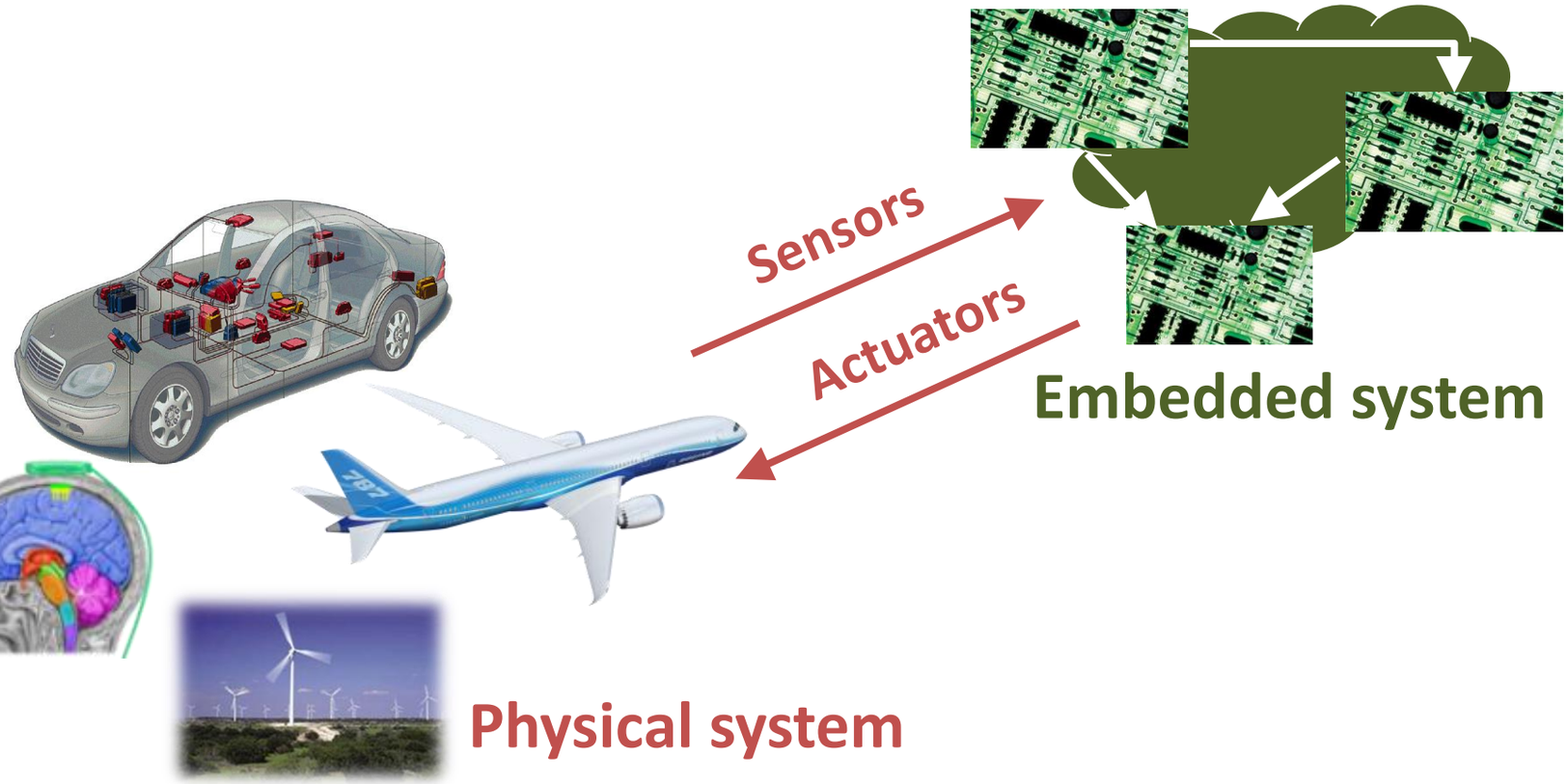
International Symposium on Physical Design, San Francisco, April 16, 2019

What is a Cyber-Physical System (CPS)?

A system characterized by the tight ***integration*** *of* ***computation, communication, and control*** *with* ***physical processes*** *via feedback loops where physical processes affect computation and vice versa*



Controller



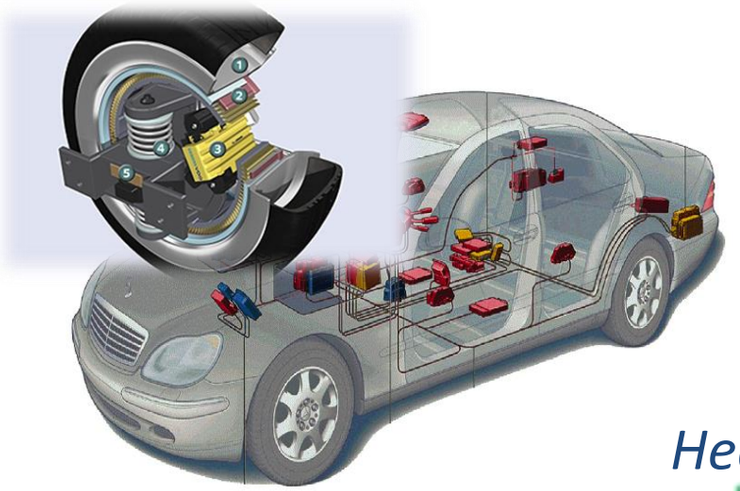
Networking

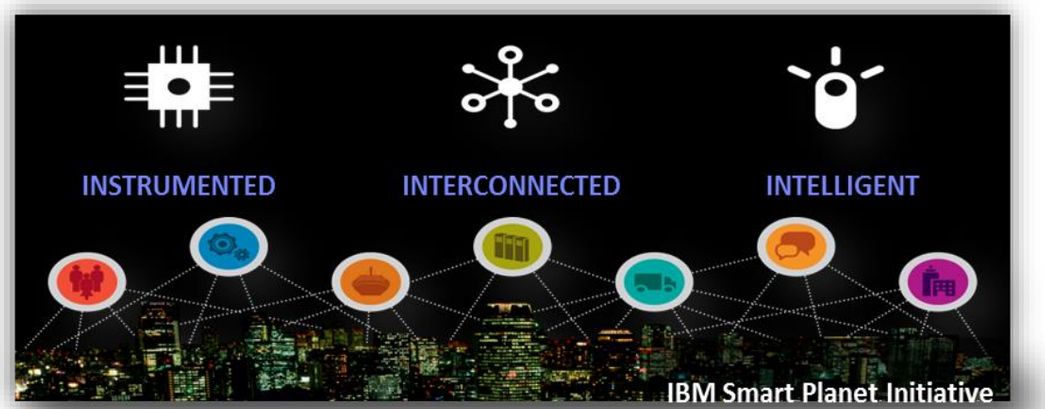
Embedded system

Physical system

CPSs Interconnect the World Around Us and Make It "Smarter"

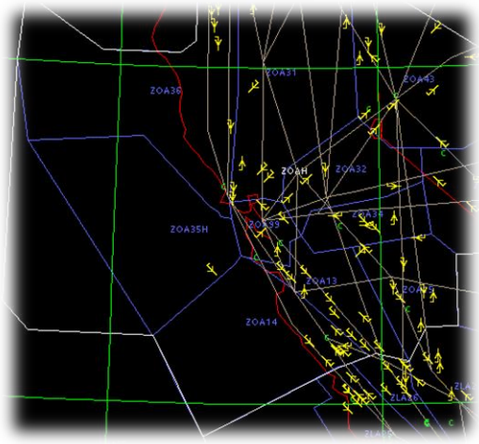
Autonomous Driving





Avionics

Transportation (Air traffic control)



Telecommunications

Health care

Buildings

Power generation and distribution

Factory automation

Resilient Cyber-Physical System Design: What Can Go Wrong?

BBC NEWS

Pilots of the crashed Ethiopian Airlines Boeing 737 Max were unable to prevent the plane repeatedly nosediving despite following procedures, an initial report has found.

The captain and first officer followed safety procedures recommended by Boeing. But **they couldn't stop the aircraft going into a fatal dive** shortly after take off from Addis Ababa on 10 March, the report by Ethiopian investigators said. All 157 people on board were killed.

Aviation authorities grounded the entire global fleet of 737 Max aircraft in March after two fatal crashes in five months.

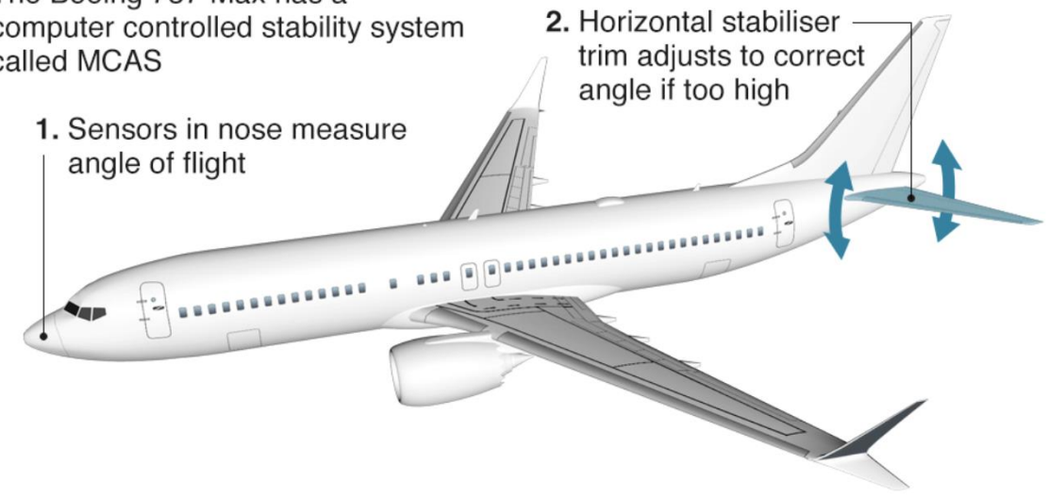
The Ethiopian Airlines crash followed a Lion Air crash in Indonesia in October, which left 189 dead.

How the MCAS system works

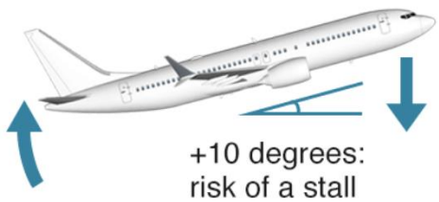
The Boeing 737 Max has a computer controlled stability system called MCAS

1. Sensors in nose measure angle of flight

2. Horizontal stabiliser trim adjusts to correct angle if too high



3. Nose pushed down to reduce risk of a stall



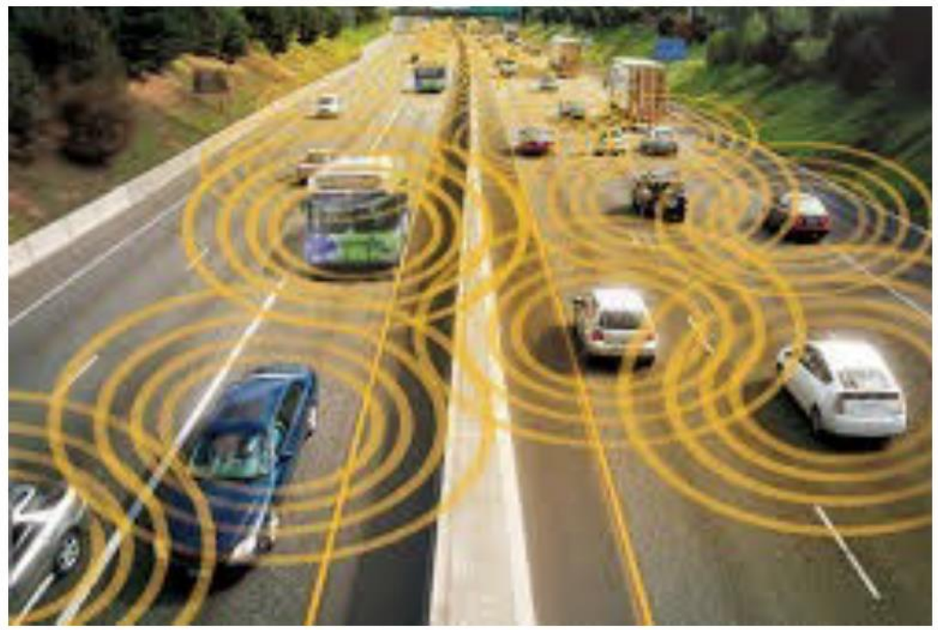
+10 degrees: risk of a stall

False reading

Reality

4. But if the sensor reading is wrong, MCAS may activate and push the nose down anyway

Resilient Cyber-Physical System Design: What Can Go Wrong?



Highly-dynamical unknown environment *and the lack of prior information*

System and components are susceptible to ***faults****, both known and unknown*

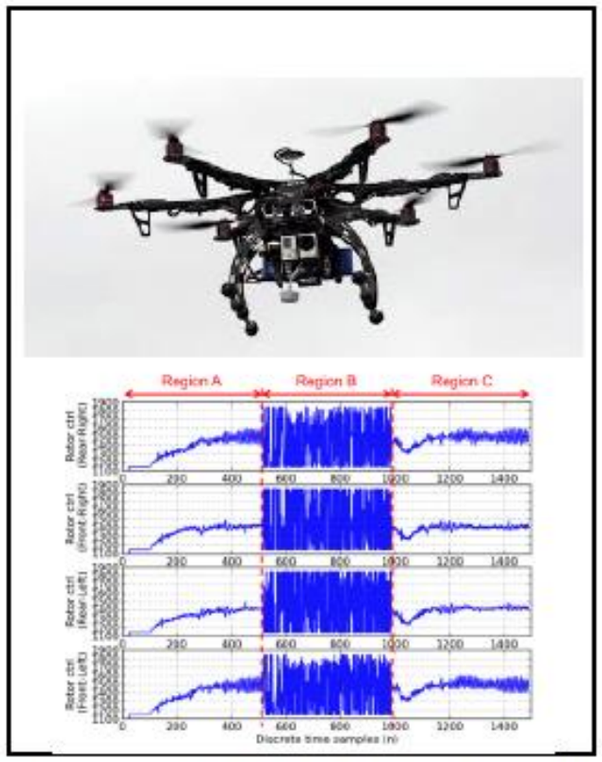
Malicious *agents can break design assumptions and trigger unexpected behaviors*

Control-theoretic approach: Design a system "robust" to faults and adversarial inputs

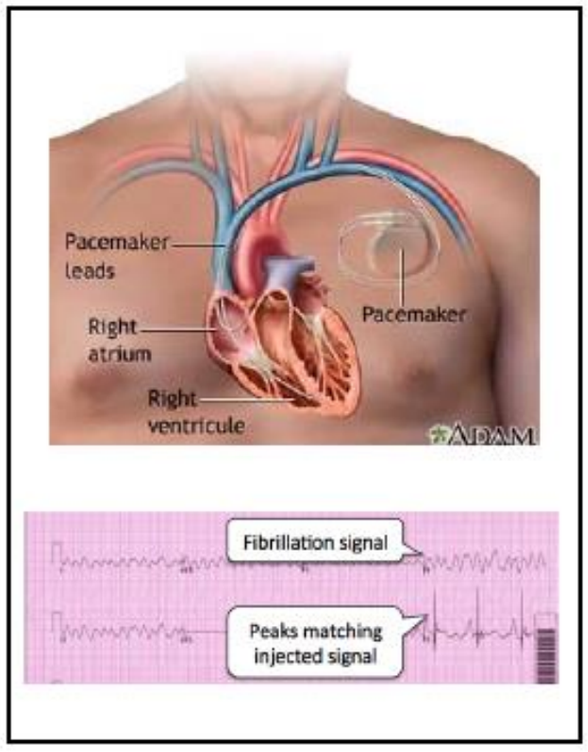
Fault-tolerance approach: Build redundancies into the system

Cryptographic approach: Authenticate agents and embed trust into components and platforms

Resilient Cyber-Physical System Design: Data Injection Attacks



Y. Son, et. al, "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," USENIX Security 2015.



D. Kune, et. al, "Ghost Talk: Mitigating EMI Signal Injection Attacks against Analog Sensors," IEEE S&P 2013.

Traditional information security is ineffective!

Need a **cross-layer approach:**

- Develop **algorithms that exploit dynamics and redundancy**
- Build **trust in HW and SW platforms**
- Co-design **algorithms with platforms**

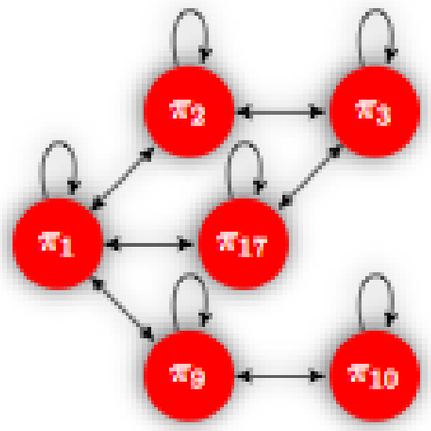
Outline

- Reasoning About Software and Dynamics: Satisfiability Modulo Convex Programming (SMC)

- Principled System-Level Design of Hardware Obfuscation: Obfuscation Design Space Exploration Engine (ODSEE)

- Conclusions

Reasoning About Software and Dynamics: Satisfiability Modulo Convex Programming (SMC)



Boolean Constraints

SAT Solvers

SMT Solvers

SAT + Convex

Mixed Integer Programming

Convex Optimization

Convex Constraints

$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$

$$y^{(t)} = Cx^{(t)}$$

"CalCS: SMT Solving for Non-Linear Convex Constraints," FMCAD 2010

"SMC: Satisfiabiity Modulo Convex Programming," Proc. IEEE 2018

Example: Secure State Estimation Against Data Injection Attacks

- A total of p **heterogenous** sensors monitor the state of the physical system:

$$y(t) = Cx(t) + \underbrace{\psi(t)}_{\text{noise}} + \underbrace{a(t)}_{\substack{\text{attack} \\ \text{vector}}}$$

- The attacker has access to s sensors, **unknown** but **fixed** over time. The value of s is also **unknown** although we assume the knowledge of an upper bound $\overline{s}$.
- At any instant of time, the attacker is free to corrupt **all/some/none** of the compromised sensors.
- The attack can be **arbitrary** (no boundedness assumption, no stochastic model).
- **Objective:** estimate (**online**) the state of the physical system $x(t) \in \mathbb{R}^n$.
- Physical system modeled as a discrete-time linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t) + \underbrace{\eta(t)}_{\substack{\text{process} \\ \text{noise}}} .$$

- Sensor noise $\psi(t)$ and process noise $\eta(t)$ (e.g., model uncertainty, nonlinearities) are bounded.

Secure State Estimation: Problem Formulation

System Dynamics:

$$\Sigma_a \begin{cases} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + a(t) \end{cases}$$

Collect τ measurements:

$$Y_i = \begin{cases} \mathcal{O}_i x + E_i & \text{if sensor } i \text{ is under attack,} \\ \mathcal{O}_i x & \text{if sensor } i \text{ is attack-free} \end{cases}$$

- Define $b_i \in \mathbb{B}$ such that $b_i = 1$ when sensor i is under attack and $b_i = 0$ otherwise
- In the context of decision procedures on the reals, we resort to the notion of δ-completeness

Problem

(Secure State Estimation) *For the linear control system under attack Σ_a, construct an estimate $\eta = (x, b) \in \mathbb{R}^n \times \mathbb{B}^p$ such that $\eta \models \phi$, i.e., η satisfies ϕ, where ϕ is defined as:*

$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \| Y_i - \mathcal{O}_i x \|_2^2 \leq \delta \right) \quad \bigwedge \quad \left(\sum_{i=1}^{p} b_i \leq \overline{s} \right).$$

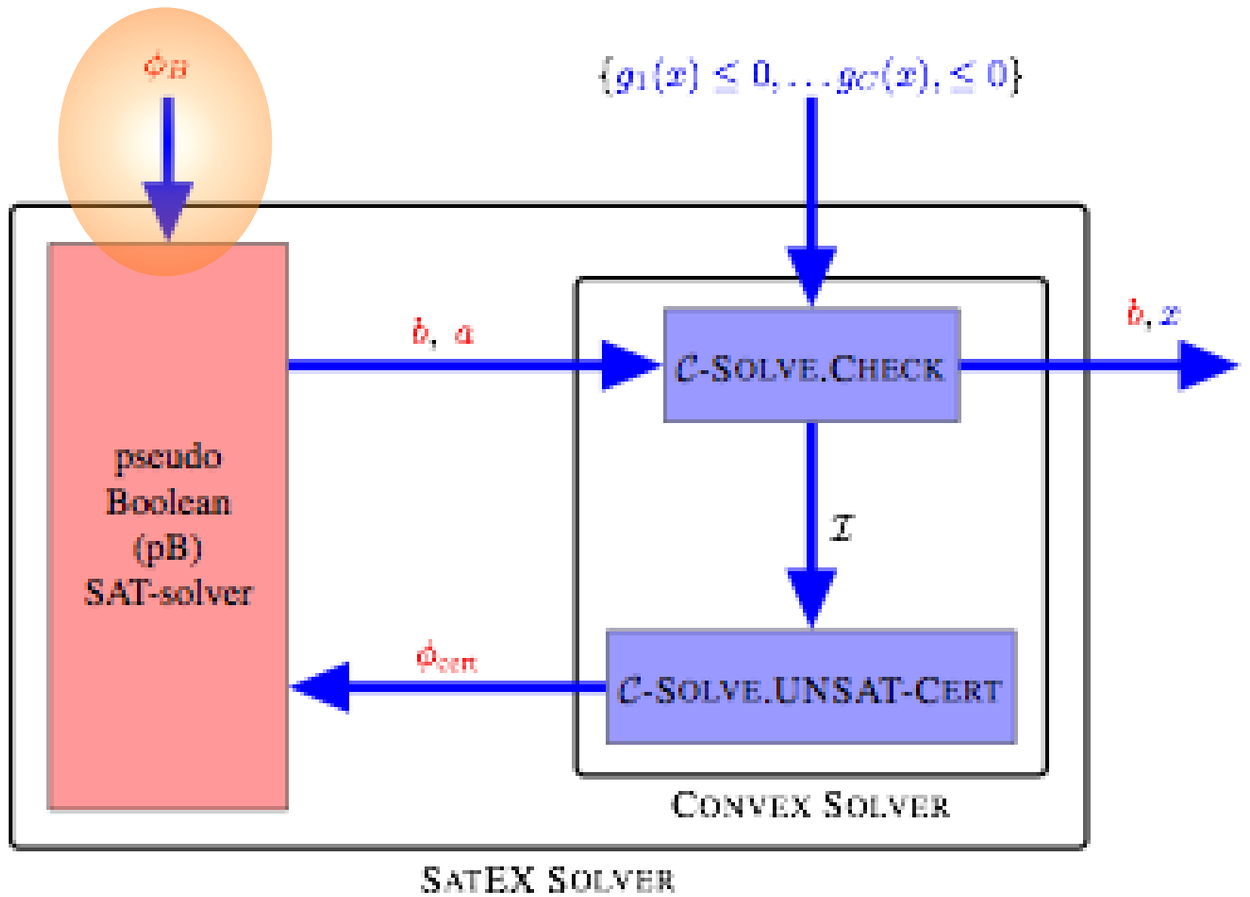
$\overline{s}$ is the maximum number of sensors under attack.

“Lazy” Coordination of SAT and Convex Programming for Monotone SMC

- **Step 1**: Solve the Boolean abstraction of the formula
- **Step II**: Extract involved convex constraints and check their feasibility

$$\min_{x \in \mathbb{R}^n} \quad 1$$
$$\text{s.t.} \quad g_j(x) \leq 0$$
$$j \in \{1, \ldots, |C|, a_j = 1\}$$

$$\phi'(a, b, x) = \phi_B \wedge \bigwedge_{i=1}^{|C|} (a_i \implies (g_i(x) \leq 0))$$



- **Step IV**: Generate UNSAT certificate: $\phi_{\text{trivial-ce}} = \bigvee_{j \in \text{supp}(a)} \neg a_j$

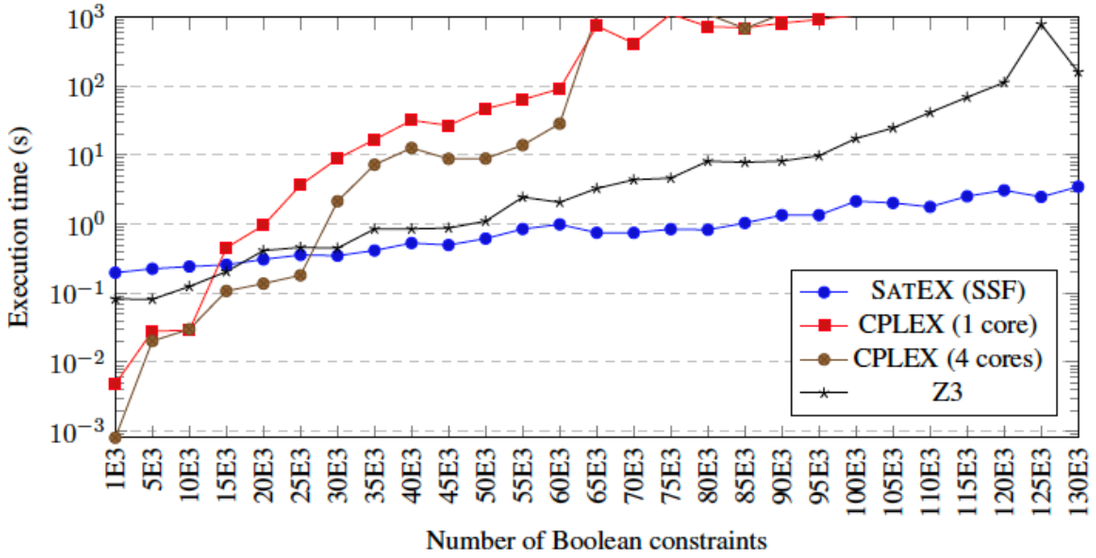
Generating Compact UNSAT Certificates

$$\text{Complexity} = \#\text{Iterations} \times \left(\underbrace{\text{Time}_{\phi(b)}}_{\text{“small”}} + \underbrace{\text{Time}_{g(x)\leq 0}}_{\text{polynomial}}\right)$$

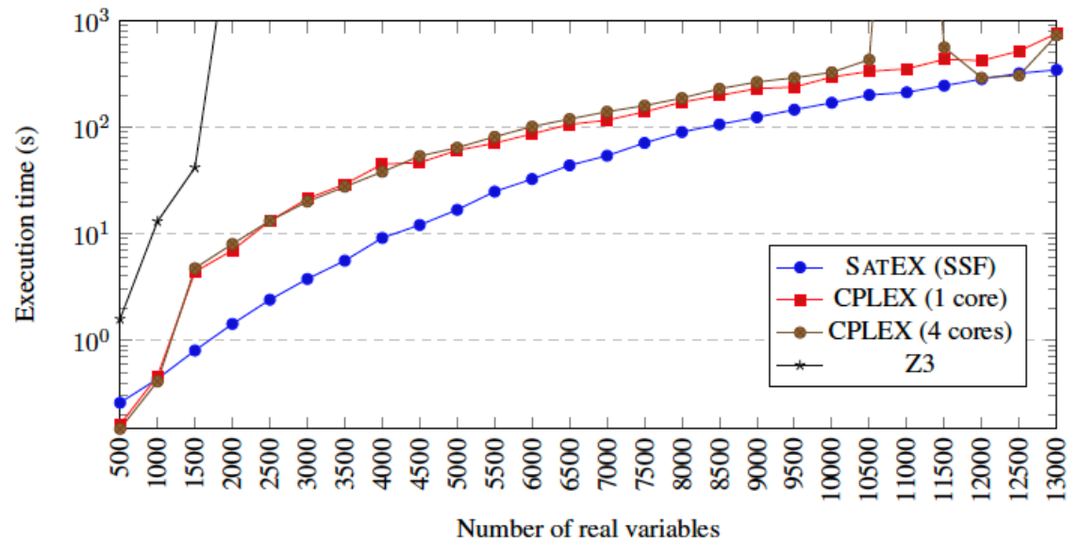
UNSAT Certificate	Minimality	Complexity (number of convex problems)
Trivial	No	Constant
Minimum Irreducible Inconsistent Set (IIS)	Yes	Exponential
Minimal IIS	Yes*	Linear/Logarithmic
Sum of Slacks	Yes*	Linear/Logarithmic
Minimum Prefix	Yes*	Constant

* under additional assumptions

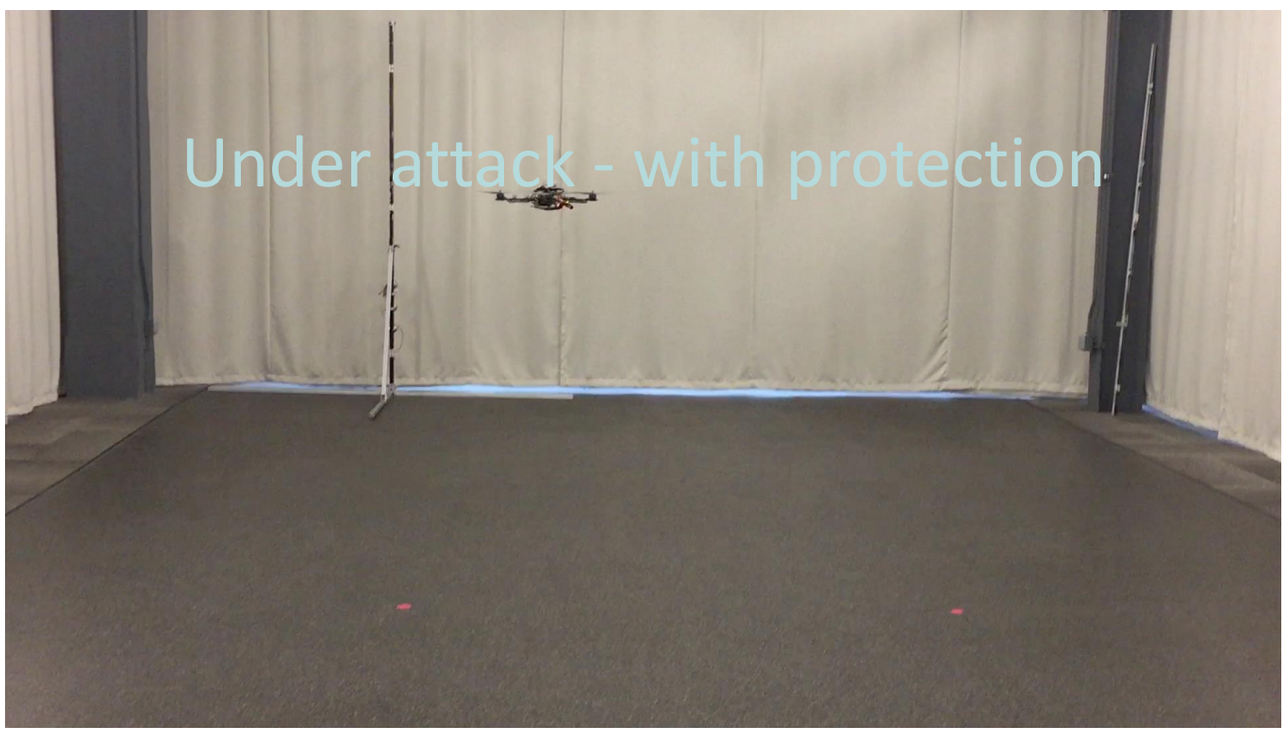
Secure State Estimation: Scalability



#Boolean variables = 4800
#Real variables = 100



#Boolean variables = 4800
#Boolean constraints = 7000



Outline

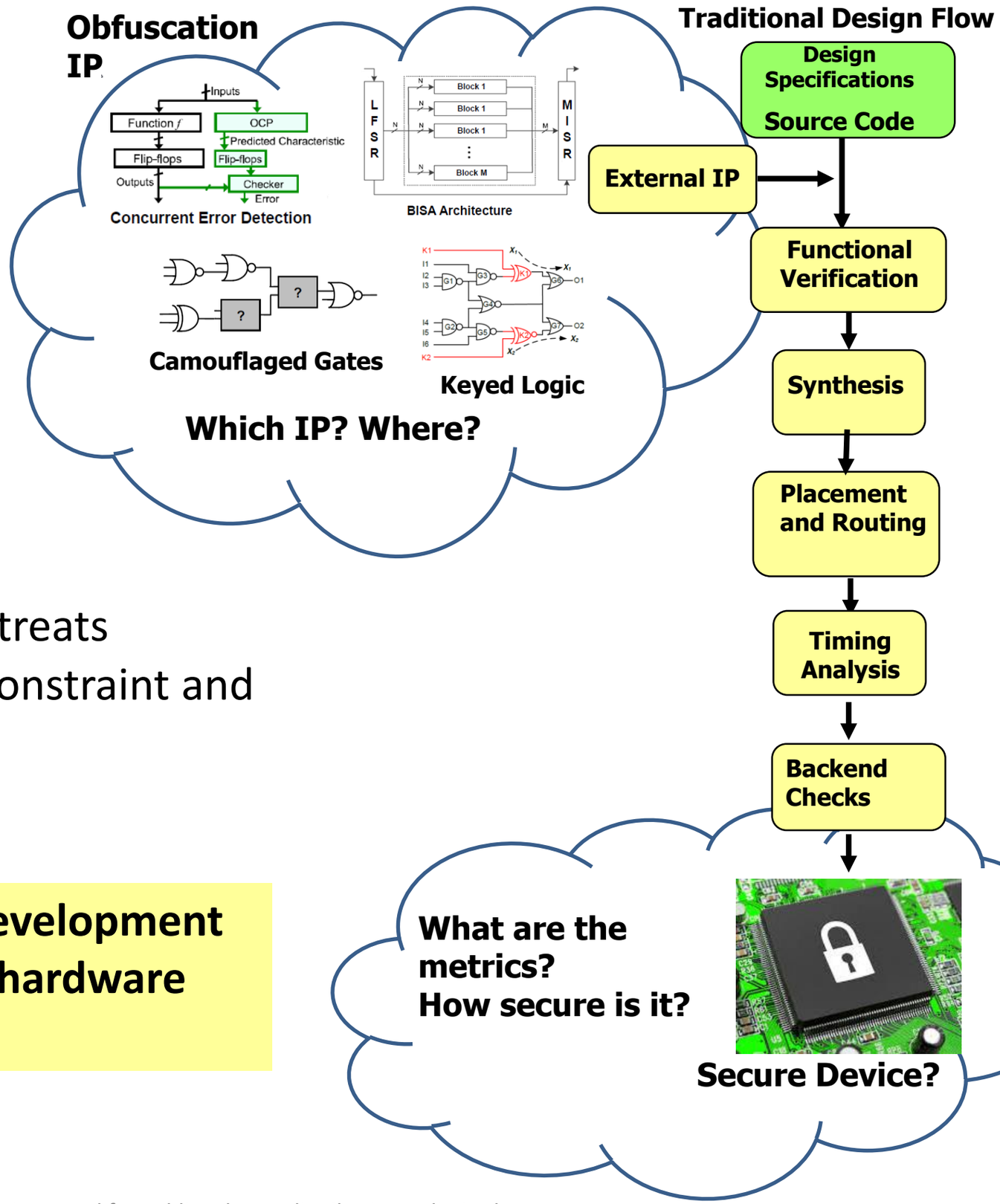
- Reasoning About Software and Dynamics: Satisfiability Modulo Convex Programming (SMC)

- Principled System-Level Design of Hardware Obfuscation: Obfuscation Design Space Exploration Engine (ODSEE)

- Conclusions

Trusted Platform Via IC Obfuscation

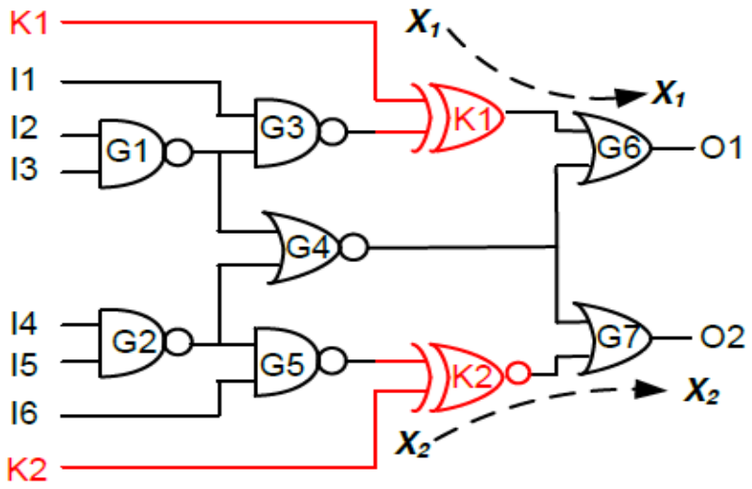
- Circuit obfuscation is a potentially viable Trust solution, however
  - No common metrics exist to evaluate techniques
  - No design tools exist to guide and validate implementation.



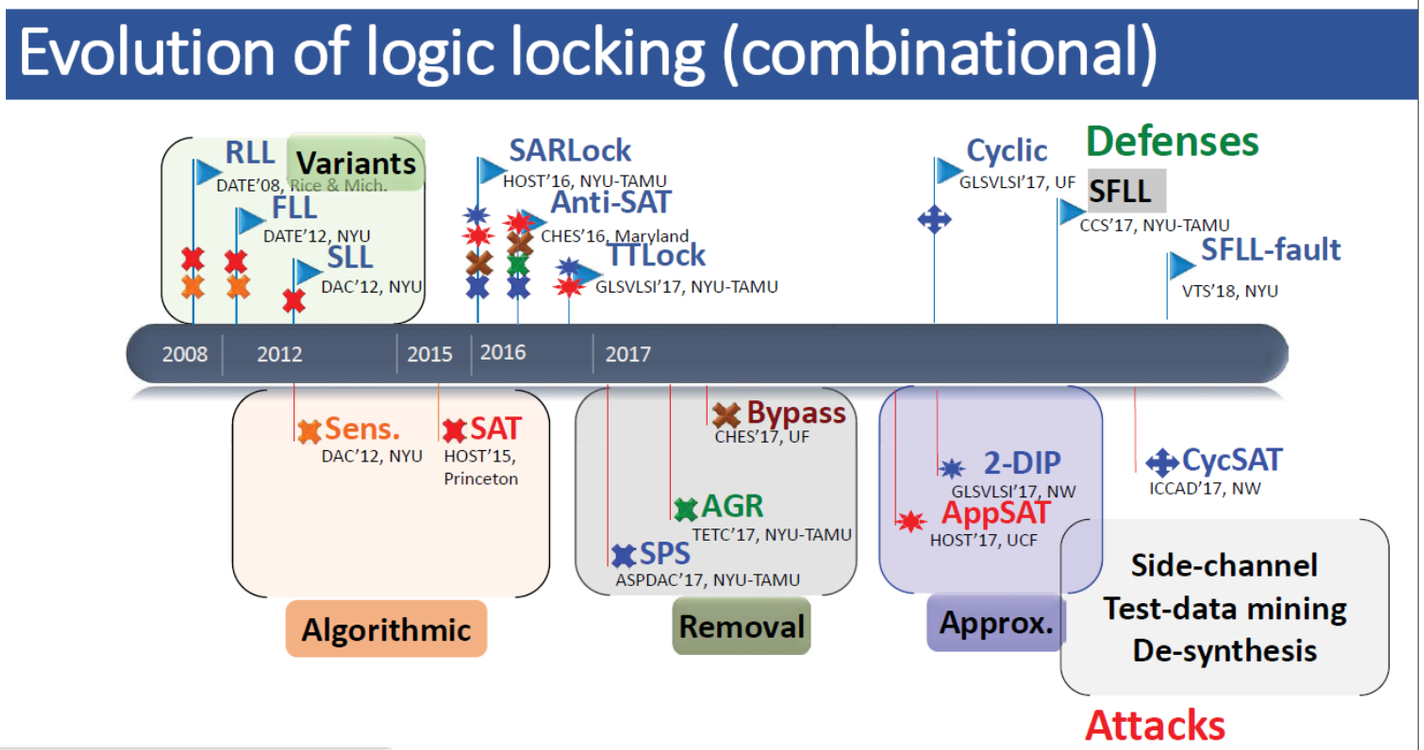
- **Mirage Project:** A tool set which treats obfuscation as a first class design constraint and relate it to system-level concerns

A scientifically based, systematic development and verification environment for hardware obfuscation security

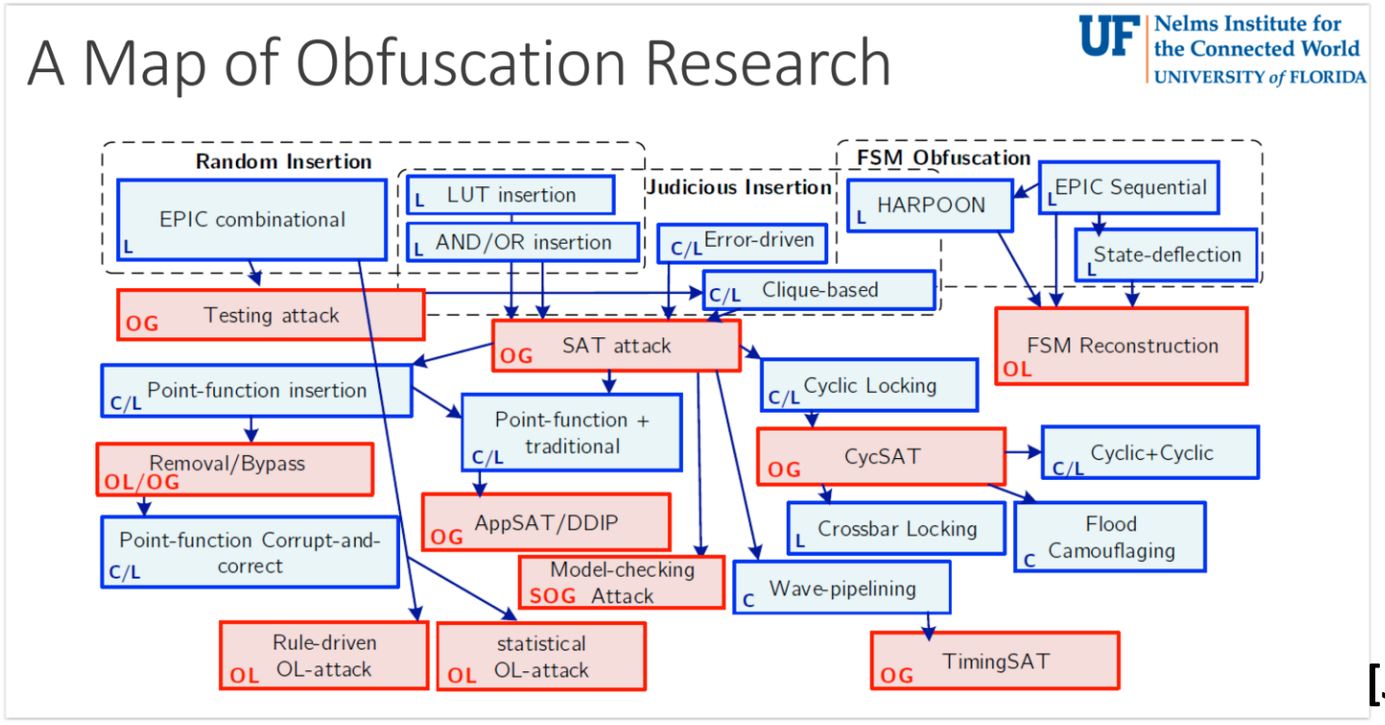
Example: Logic Locking (Encryption)



Sample Locked Circuit
[Yasin TCAD 2015]

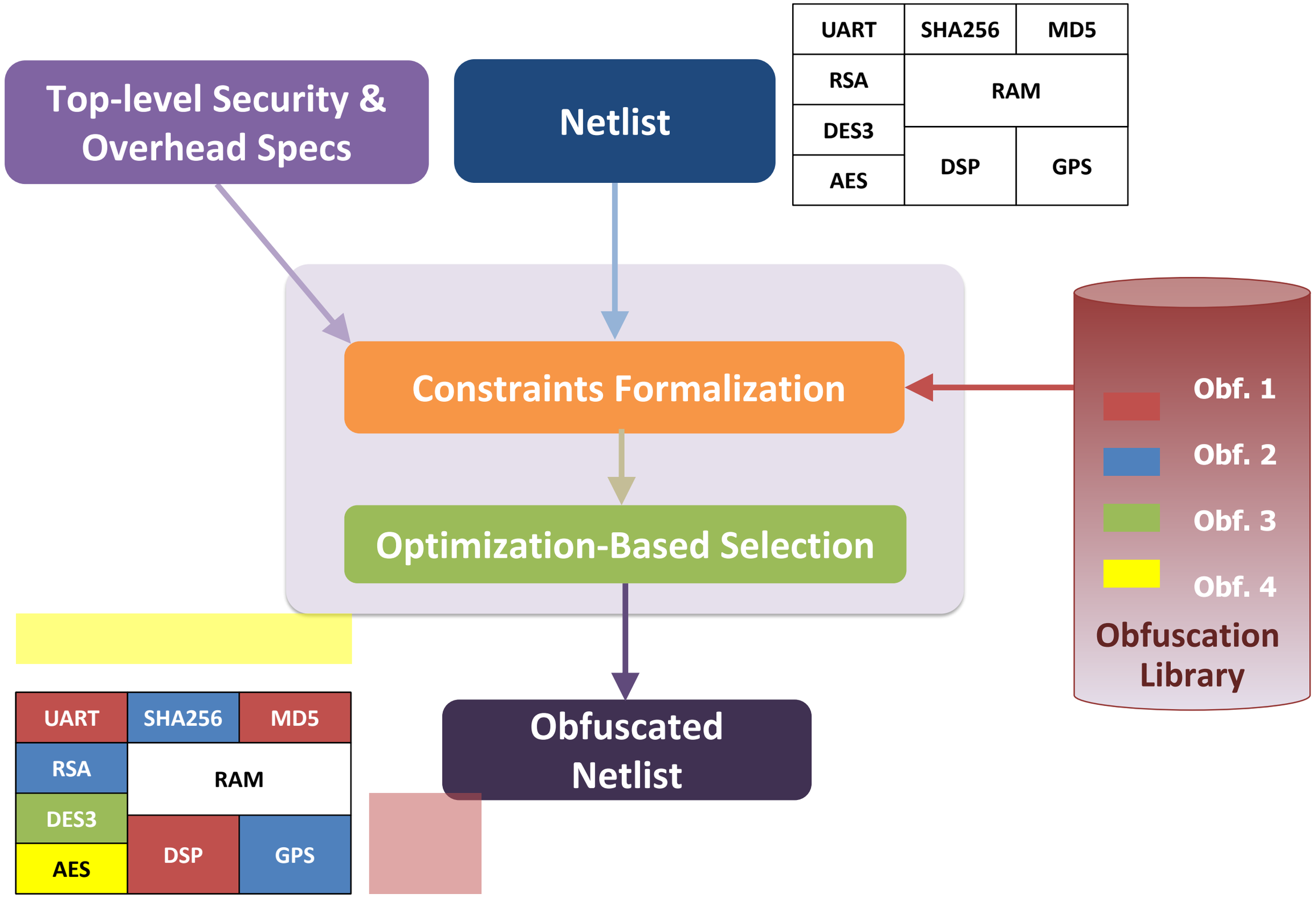


Attack progression timeline
[Rajendran, ECLIPSE, 2018]



[Jin, Feb 2019]

ODSEE's Architecture



Security Specifications: Disentangling Functional and Structural Properties of Circuits

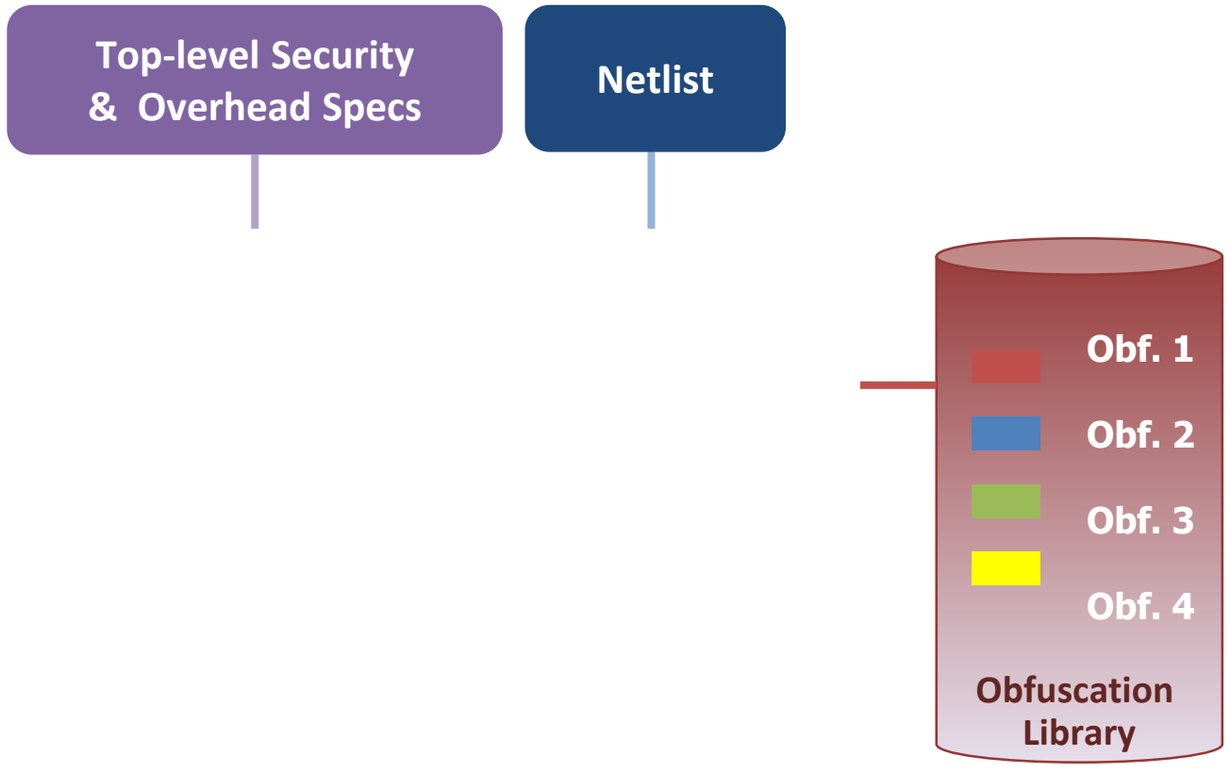
ODSEE rethinks the taxonomy and metrics for capturing security requirements:

- What would we like to protect?
  - Logic/functional properties
    - Output/functional corruptibility
    - SAT-attack resiliency
    - Structural properties
    - …
- What is the attack model?
  - Targets logic properties: e.g., SAT attack, Approximate SAT-based attacks, …
  - Targets structural properties: e.g., removal attack

Obfuscation Library: Disentangling Functional and Structural Properties of Obfuscation Schemes

ODSEE rethinks the taxonomy and metrics for modeling obfuscation schemes:

- Targeting high error rates
  - XOR/XNOR based: e.g., Fault-based analysis Logic Locking (FLL), Random Logic Locking (RLL), Strong Logic Locking, …
  - LUT based
  - …



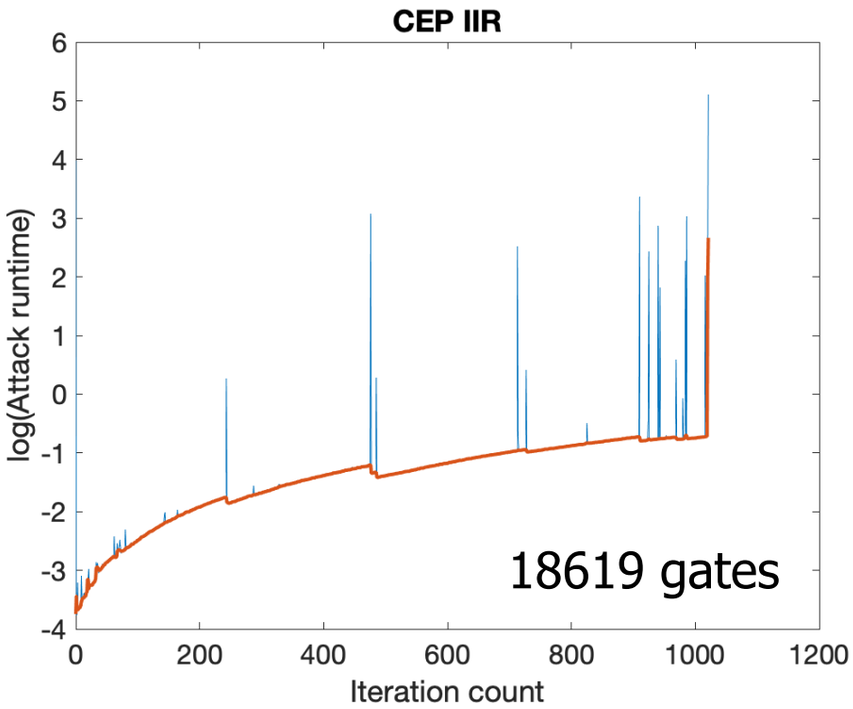
- Targeting SAT resilience
  - SARLock
  - Anti-SAT
  - …

- Targeting structural attacks

- Hybrid schemes targeting a mixture of metrics

Obfuscation Library: Accurately Representing Implementation Aspects of Obfuscation Schemes

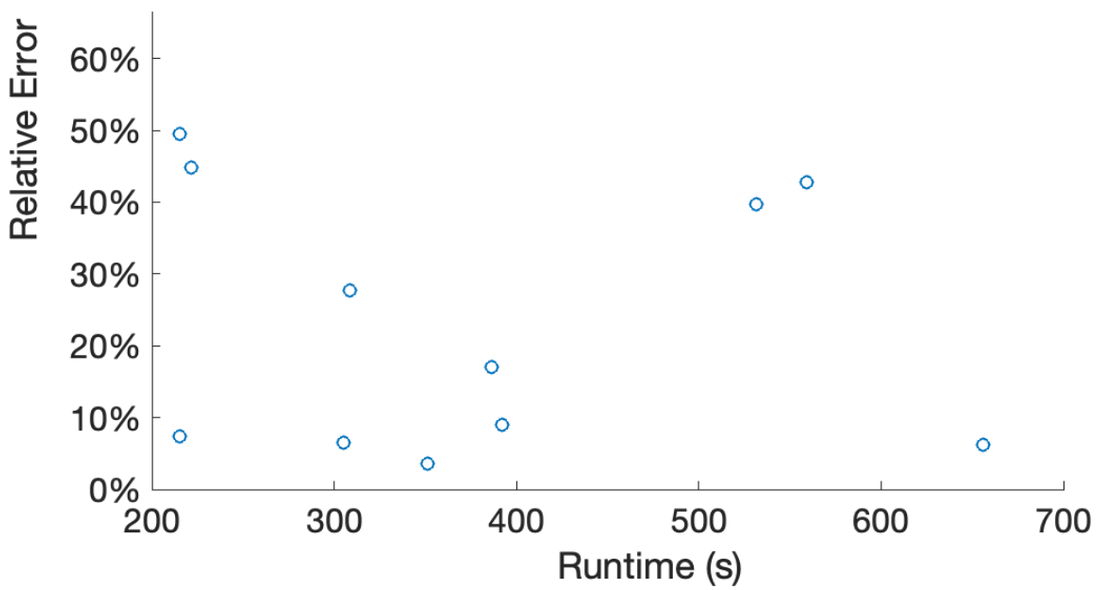
ODSEE incorporates accurate circuit-aware compact models of obfuscation techniques, their effectiveness, and their cost



$$t_{SARLock} \approx \beta G \cdot 2^{2K} + 2^{\gamma G}$$

K is the number of key bits
G is the gate count

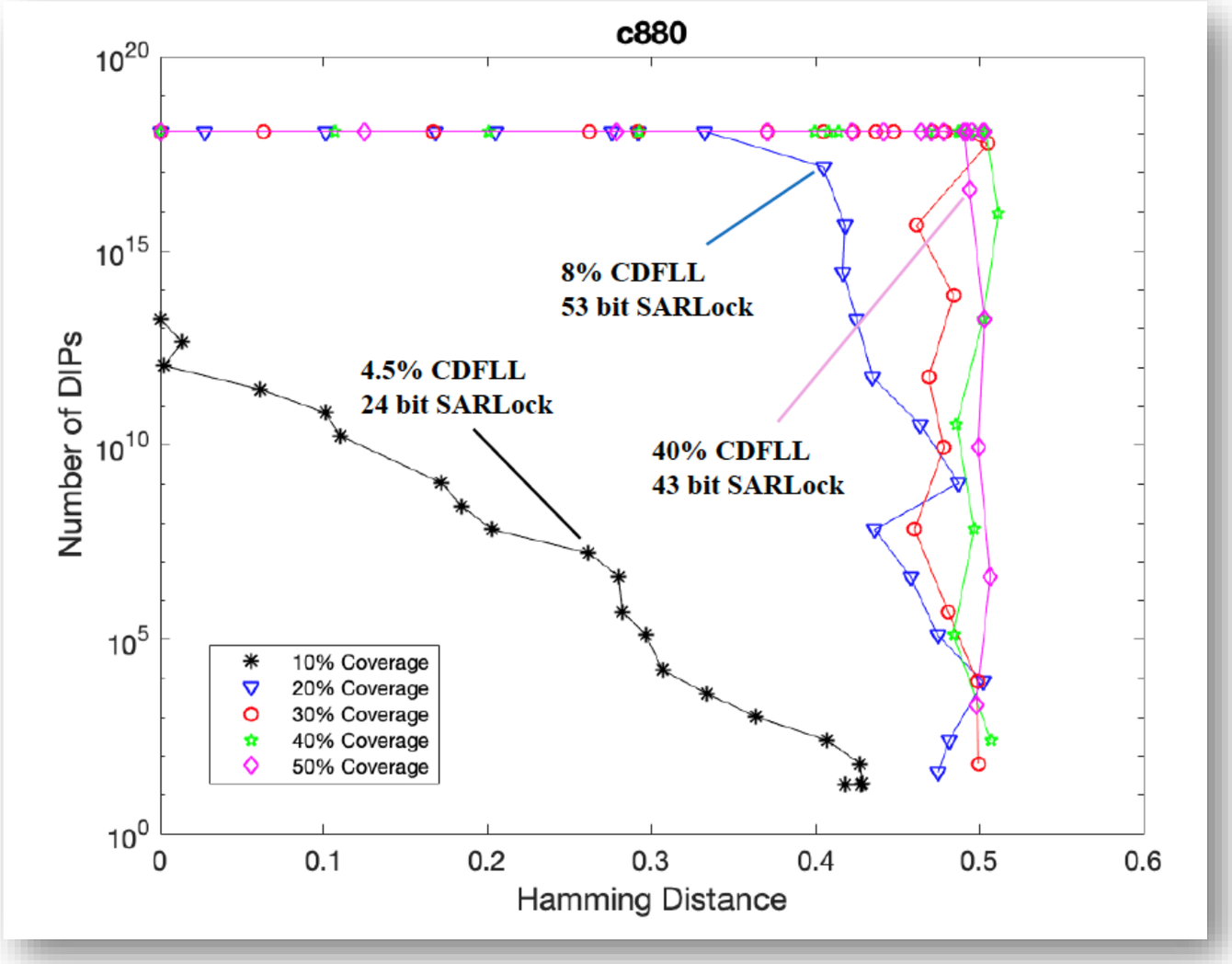
Relative error is below 30% for most cases



Mapping Specifications to Implementations: Constraint-Driven Logic Locking (CDLL)

ODSEE captures constraints from different concerns and obfuscation schemes using a uniform language

- Constraints from fault analysis
- Conditions on controllability and observability
- Conditions involving fan-in/fan-out cones
- Can protect specific input patterns
- Can identify and select specific locations in the netlist
- Enables hybrid obfuscation



Current ODSEE implementation is based on mixed integer linear constraints and leverages mathematical programming to select Pareto optimal obfuscation schemes

Conclusions

- Orchestrating billions of devices around our body, transportation systems, critical infrastructures, and the planet presents unprecedented design challenges

- High-assurance cyber-physical system design will require cross-disciplinary, cross-layer approaches

- SMC and ODSEE are formal frameworks that enable reasoning across the algorithms/HW/physical boundaries

USC Viterbi
School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

Thank you.