Protecting Information

UF UNIVERSITY of FLORIDA

Duke UNIVERSITY

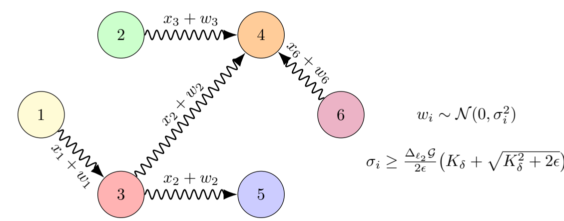
TEXAS The University of Texas at Austin

UC SANTA CRUZ

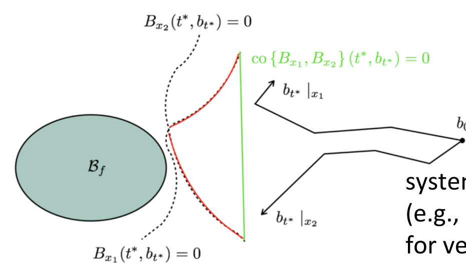
Protecting Information

- RT6 focuses on methods to protect against indirect vulnerabilities caused by adversarial observation of computation, communication, and mission execution

- Major initiatives this year:
  - Protecting information through examination of entropy and Markov decision processes
  - Differential privacy
  - Systems approaches to privacy, access control, and machine learning



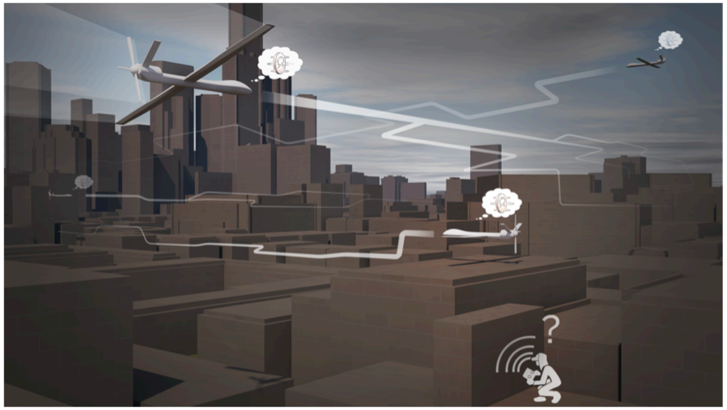
Noise-adding mechanisms to tradeoff individual privacy and aggregate performance



system-theoretic principles (e.g., barrier certificates) for verifying privacy

Mission-Critical Information

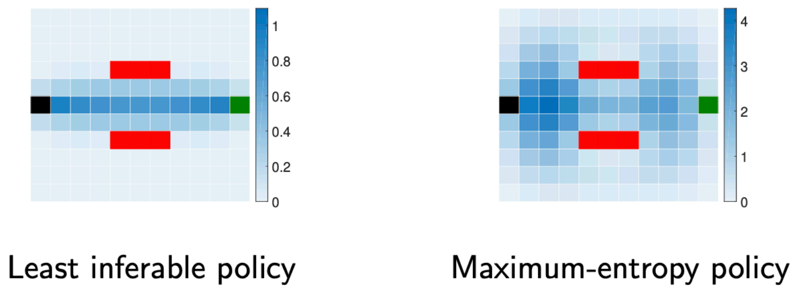
- Entropy maximization in Markov decision processes subject to temporal logic constraints
  - Synthesize constrained, entropy-maximizing strategies
  - The higher the entropy, the less predictable
  - Convex-optimization-based synthesis



- Entropy maximization in partially observable Markov decision processes
  - Extension to decision making with limited information at runtime

Policies and Information Leakage

- Least inferable policies in partially-observable Markov decision
  - Accounts for both the amount and informativeness of the adversary's observation
- Minimizing information leakage regarding high-level task specification

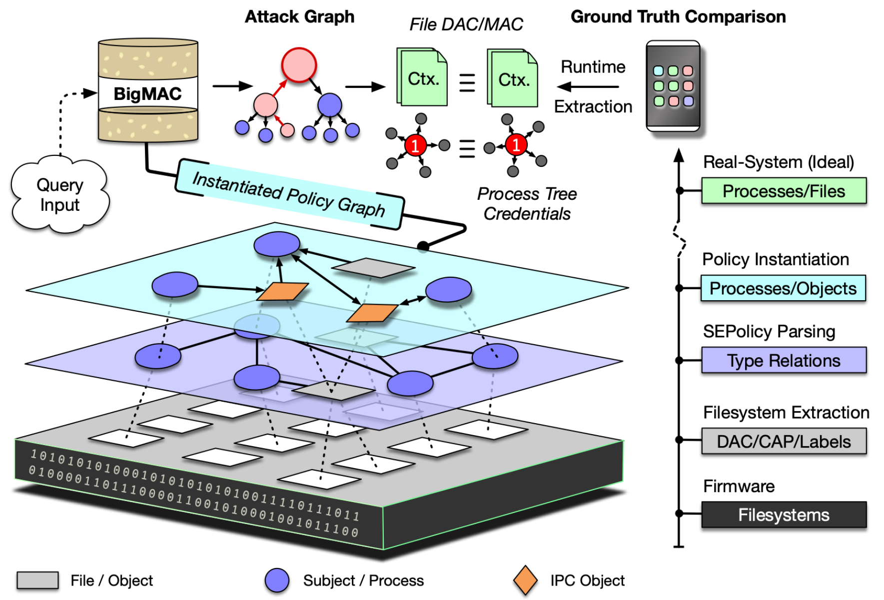


Differential Privacy

- Dirchlect mechanism for differential privacy on the user simplex
  - Provides differential privacy to Markov decision process properties
  - Ensures recipient of data cannot learn anything meaningful from privatized simplex data
- Error bounds and guidelines for privacy calibration in differentially-private Kalman Filtering
  - First control-theoretic guidelines for calibrating differential privacy
- Differentially private controller synthesis with metric temporal logic specifications
  - Multi-agent control policies with differential privacy

Systems and Access Control

- Privacy-preserving secure localization
  - Extension of privacy-preserving local optimization to embedded device processors
- Framework for secure machine learning
  - Interpretable security reference monitor design with applications to autonomous agents
- Access control policy
  - Framework for reasoning about disparate access policies in embedded devices



Access Control Policy Analysis for Embedded Devices

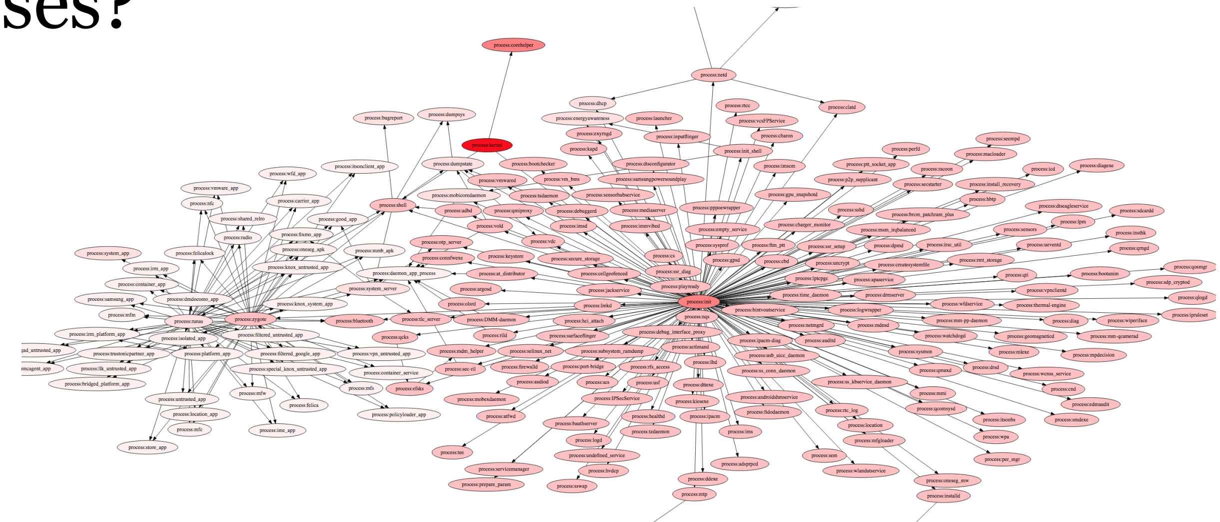
with Grant Hernandez, Dave Tian, Anurag Yadav, and Byron Williams

Embedded Devices

- Embedded agents and the operating systems that they run are subject to numerous methods for protecting on-device information
- Example: Android OS – the world's most popular operating system (mobile and embedded devices)
- Numerous ways of protecting access to on-device critical information
  - Discretionary access controls
  - Mandatory access controls
  - Linux capabilities
  - Middleware protections/SECCOMP
- Many attack surfaces

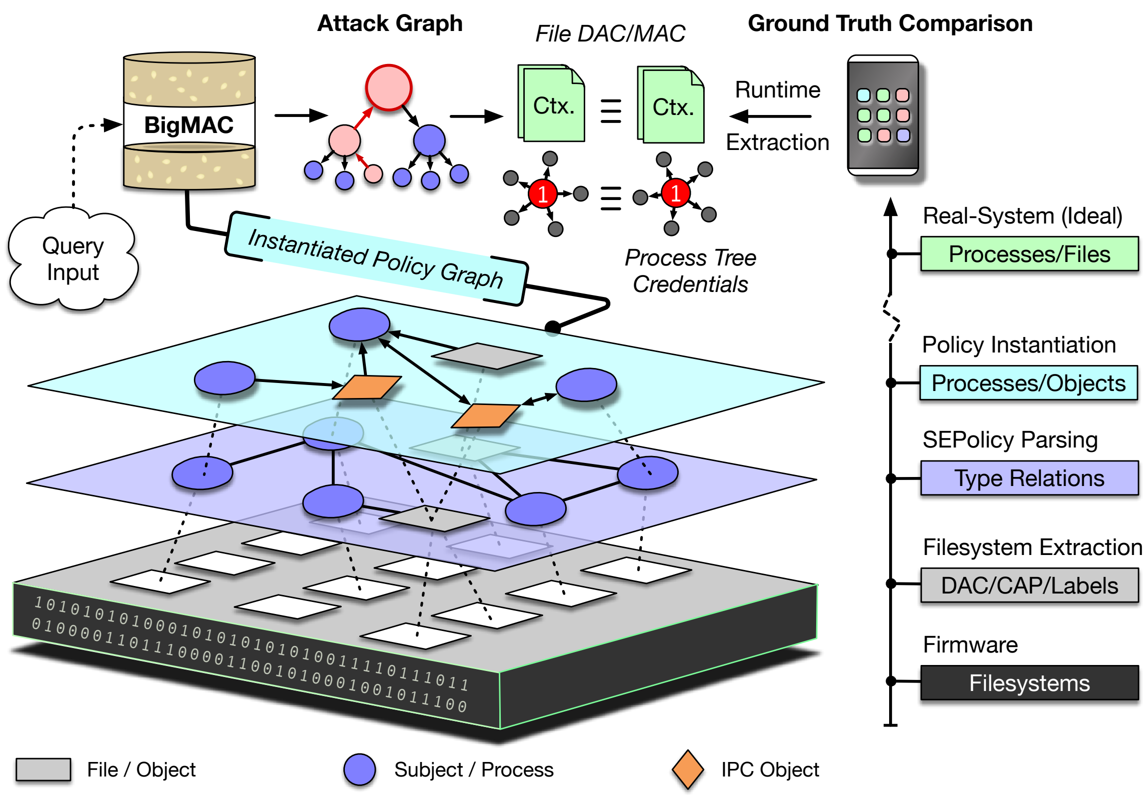
Reasoning about Access Control

- Challenge: there is no way to currently reason about the myriad access control mechanisms
- Consequence: access policies from different mechanisms not in concert with each other, may be mutually conflicting
- What objects and processes can be accessible to untrusted processes?



BigMAC Design

- Examine, recover, simulate, fully instantiate files, process, IPC from an extracted firmware image
- Simulate boot process to recover types instantiated at runtime
  - Process hierarchy and process metadata

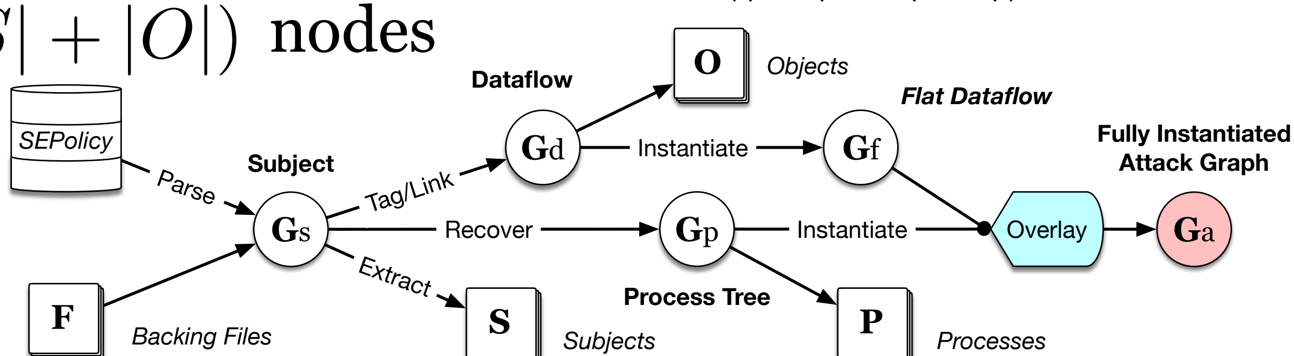


Policy Recovery

- Decompile binary SEPolicy into connected multi-edge directed graph using Access Vector rules
  - Subject graph includes all types and attributes used during SELinux MAC type enforcement
  - Instantiate on filesystem through type enforcement rules

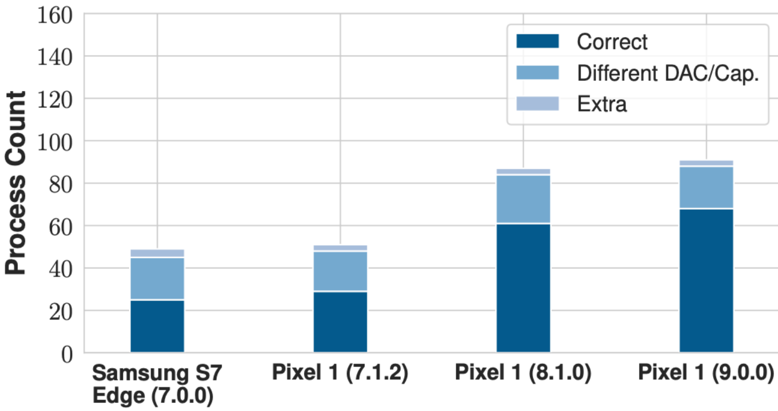
$$C_p : S_i \xrightarrow{(O_j, C_p)} S_j$$

  - Build dataflow graph from subject graph to examine paths and edges where privilege escalation may occur
  - Bipartite graph, worst case $\mathcal{O}(|S| * |O|)$ edges, $\mathcal{O}(|S| + |O|)$ nodes

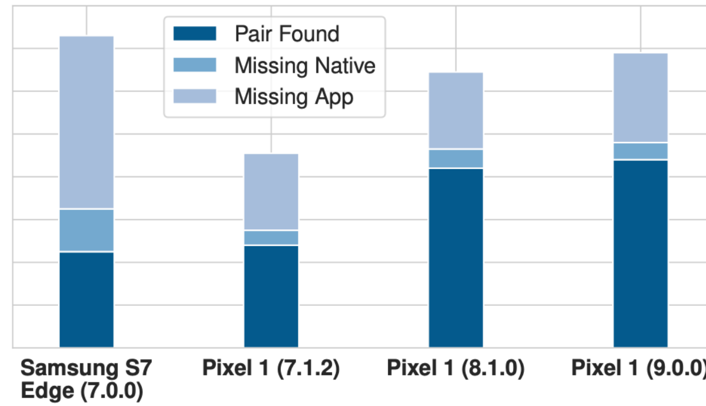


Evaluation/Case Study

- Over 98% of DAC and MAC data recovered
- Approx. 75% of processes (mostly missing app level)

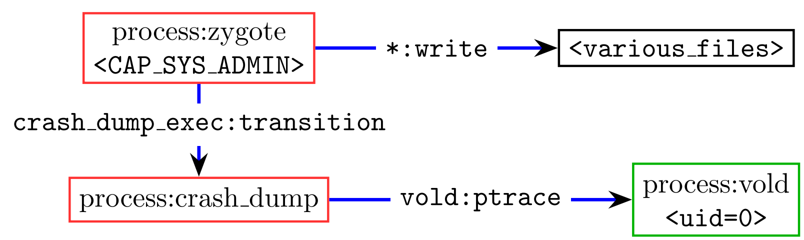


Recovered by BigMAC



Ground truth, running device

- Prolog-based query engine filter MAC,DAC, CAP, ext. surfaces
- Found additional attack paths for existing CVE

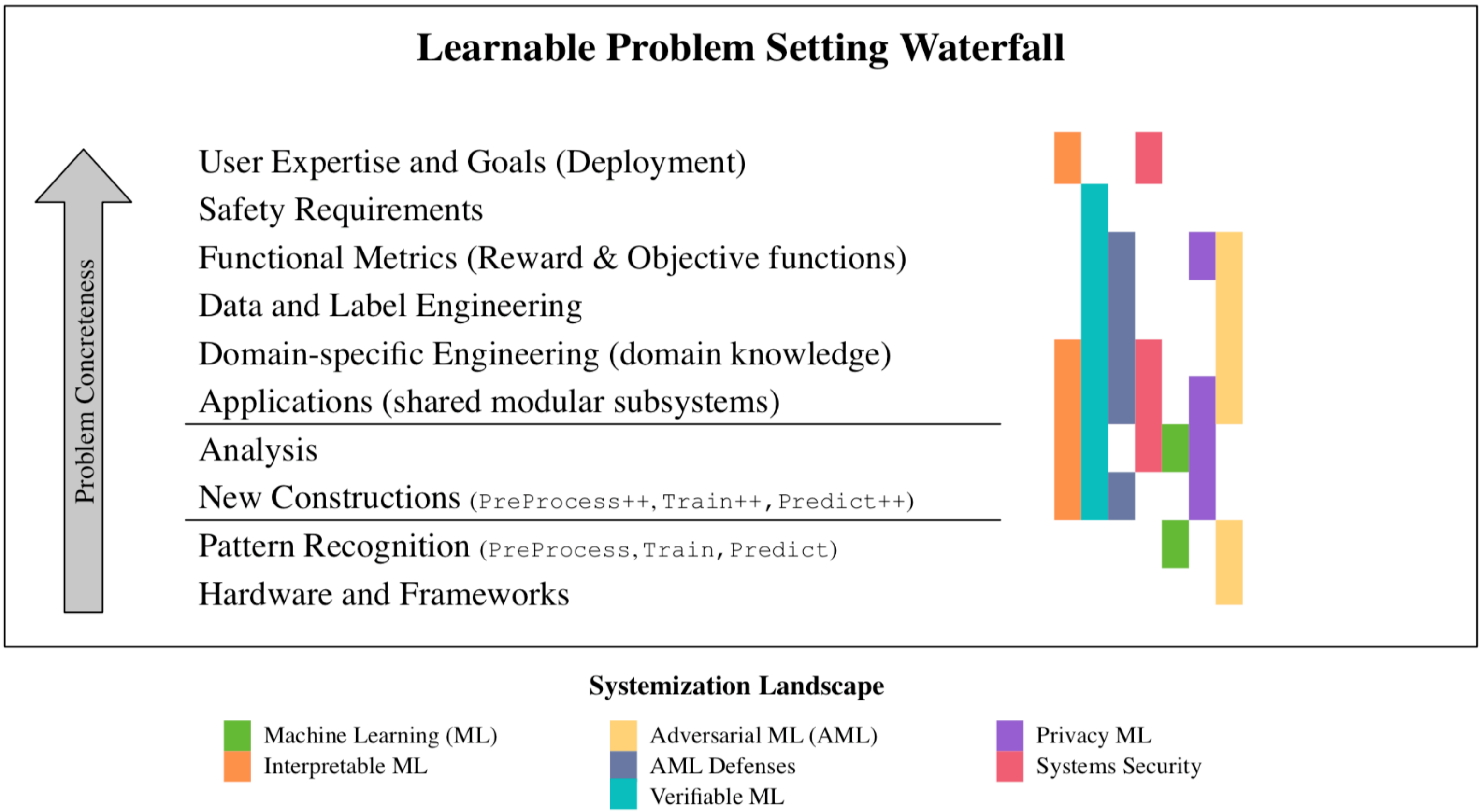


UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

A Learning Mechanism for Learning Systems

with Washington Garcia, Scott Clouse (AFRL/ACT3)

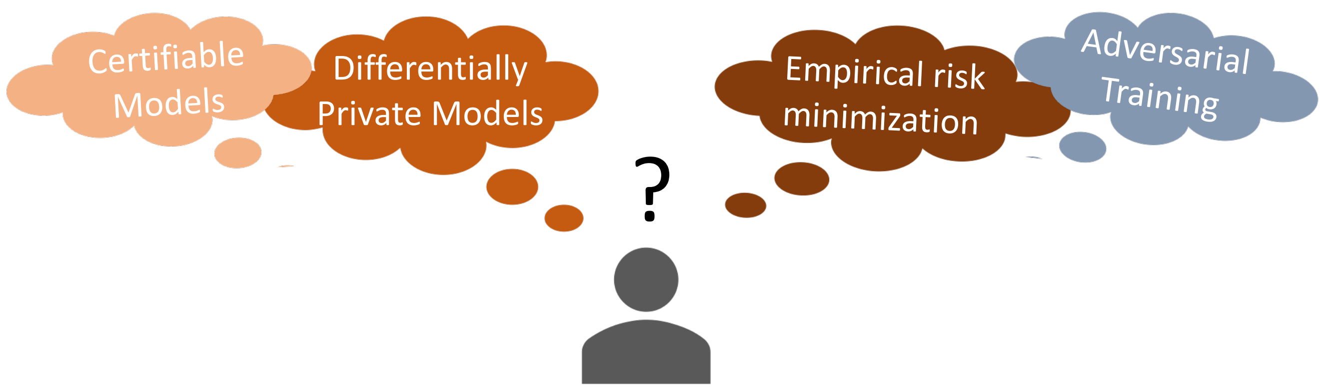
- Secure solutions to learning problems requires cooperation between many research disciplines.



Motivation

- Secure solutions to learning problems requires cooperation between many research disciplines.

- However, learning systems research tends to be conducted in an ad-hoc manner:
  - Threat modeling has only recently received attention from ML researchers.
  - If the threat model is known, choosing among seemingly competing frameworks can be difficult.



Motivation

- Secure solutions to learning problems requires cooperation between many research disciplines.

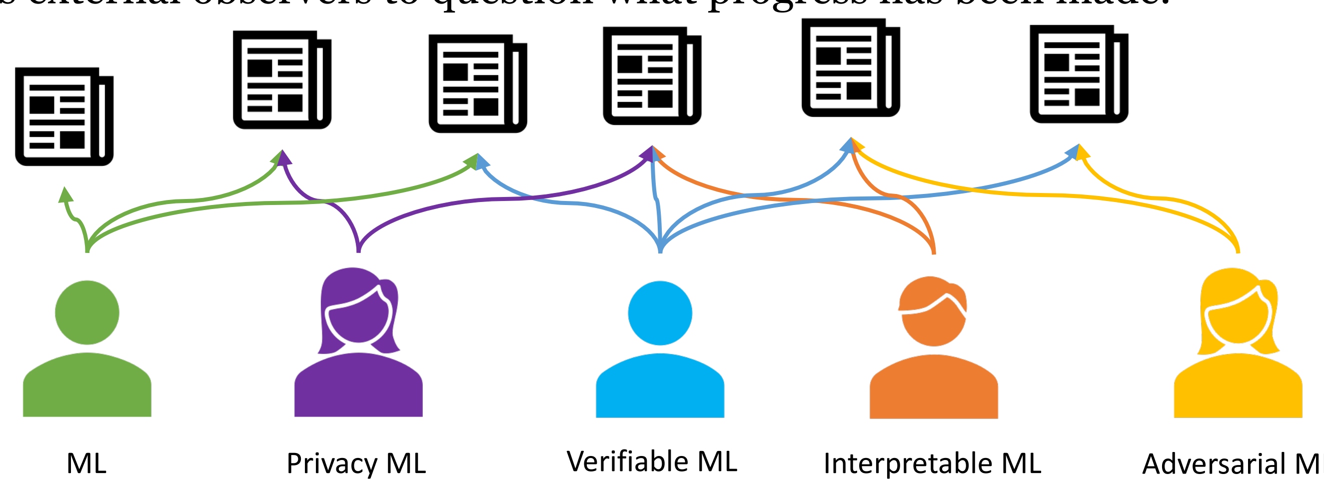
- However, learning systems research tends to be conducted in an ad-hoc manner:
  - Threat modeling has only recently received attention from ML researchers.
  - If the threat model is known, choosing among seemingly competing frameworks can be difficult.
  - Lack of clarity mitigates cross-pollination between each research discipline, causes external observers to question what progress has been made.

Motivation

- Secure solutions to learning problems requires cooperation between many research disciplines.

- However, learning systems research tends to be conducted in an ad-hoc manner:
  - Threat modeling has only recently received attention from ML researchers.
  - If the threat model is known, choosing among seemingly competing frameworks can be difficult.
  - Lack of clarity mitigates cross-pollination between each research discipline, causes external observers to question what progress has been made.

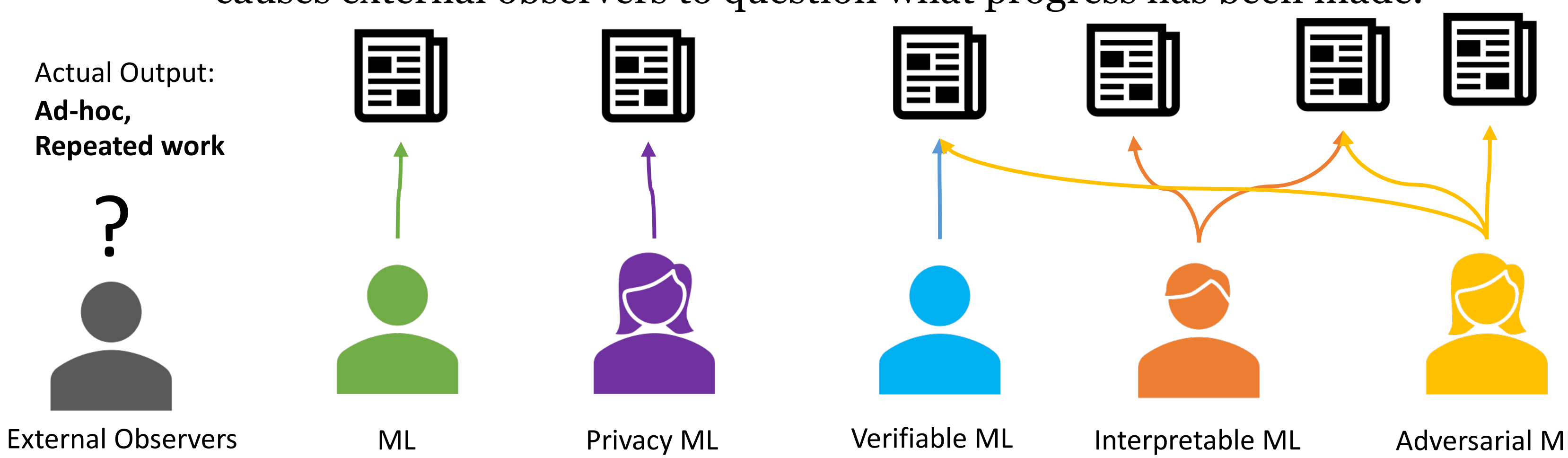
Potential Output:
Unified proposals



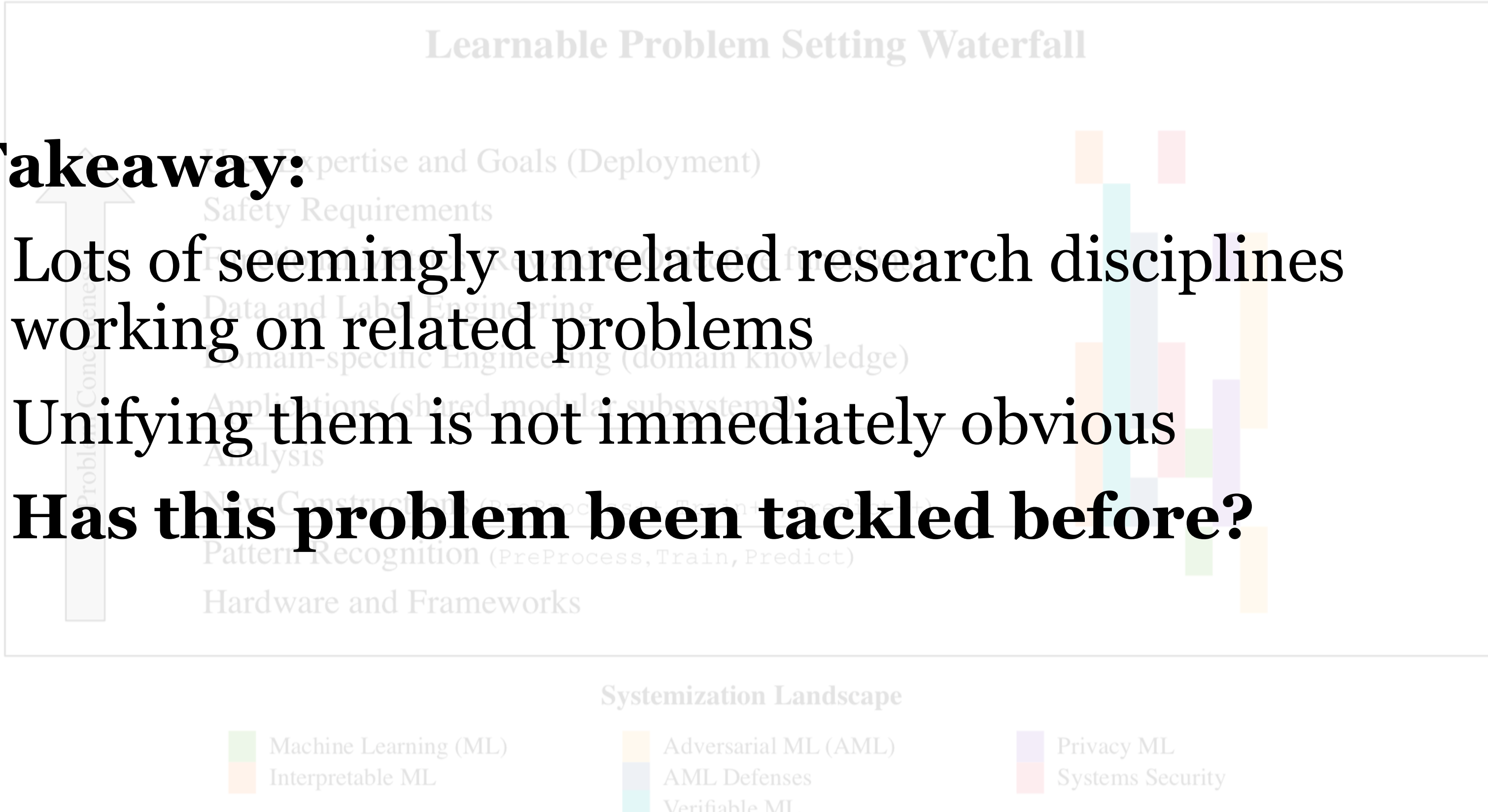
Motivation

- Secure solutions to learning problems requires cooperation between many research disciplines.

- However, learning systems research tends to be conducted in an ad-hoc manner:
  - Threat modeling has only recently received attention from ML researchers.
  - If the threat model is known, choosing among seemingly competing frameworks can be difficult.
  - Lack of clarity mitigates cross-pollination between each research discipline, causes external observers to question what progress has been made.



UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ



Takeaway:

- Lots of seemingly unrelated research disciplines working on related problems
- Unifying them is not immediately obvious
- **Has this problem been tackled before?**

Anderson research model

Has this problem been tackled before?

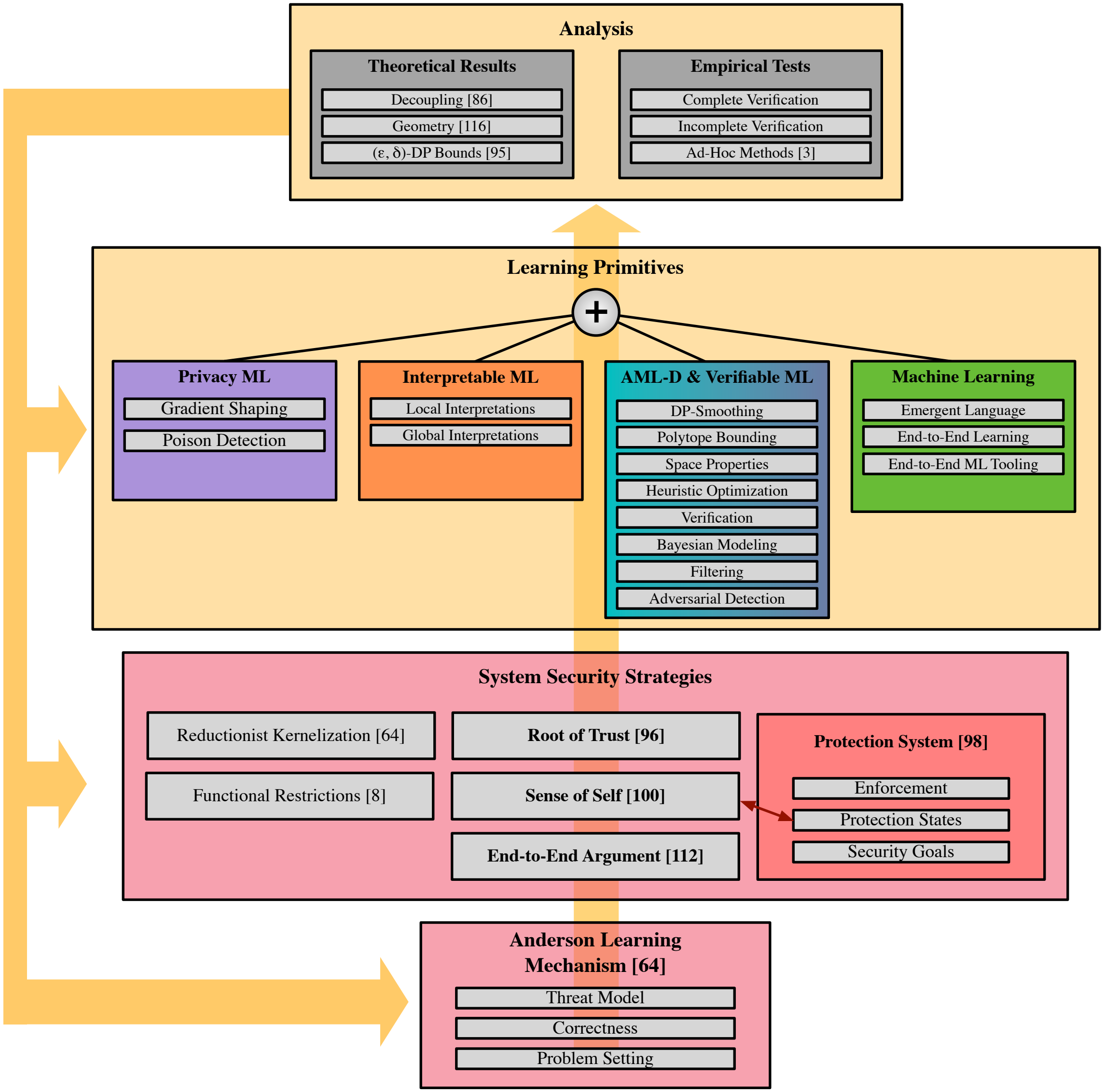
In 1993, Anderson described cryptographic systems which faced similar issues:

- Unclear threat models:
  - what *could* happen vs. what was *likely* to happen.
- Little communication due to centralized certification authority
  - crypto systems mainly cared about passing certification

Unifying research in ML with systems security:

- Adopt Anderson's research model to design secure learning systems
- Organize around system security strategies which manifest within ML research

Proposed Research Model



Instantiation of model

Example using:

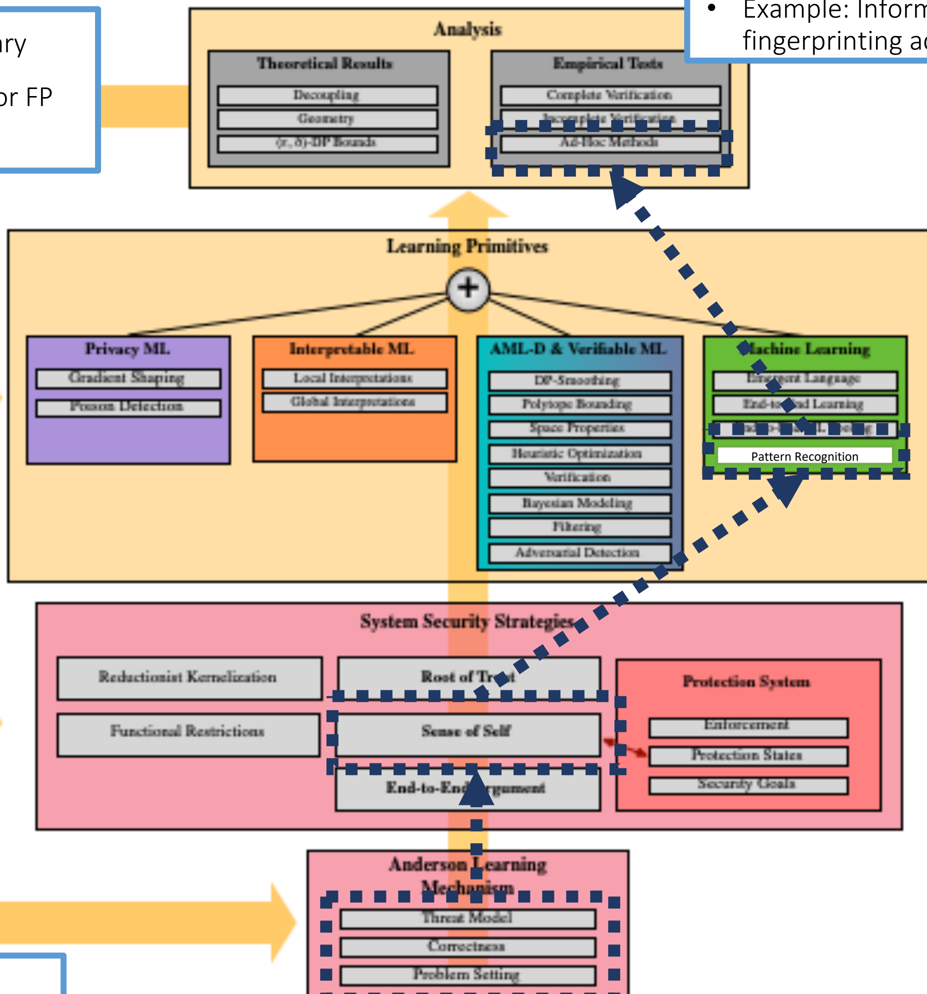
"Brittle Features of Device Authentication"

Analysis:

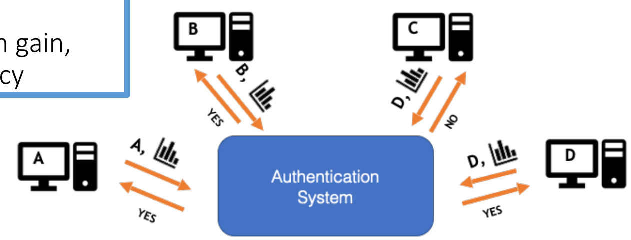
- Ad-hoc empirical test.
- Example: Information gain, fingerprinting accuracy

Modify adversary

(originally none for FP systems)



Re-analyze



The system uses sense of self to:

- Protect the integrity of network resources, by offering physical authentication channel.
- Example: Encode the sense of self using machine learning models.

The system captures:

- Forrest et al. sense of self: encode a probabilistic definition of normal and abnormal.

Establish likely problem setting (device fingerprints)

Conditions for correctness (true positives/negatives)

Plausible threat model (hard label adversary)

Instantiation of model

Example using:

"Brittle Features of Device Authentication"

Zeroth Order Optimization framework

Modify adversary

~~(originally none for FP systems)~~ ZOO

$$C(x) := \arg\max_{c \in [m]} F_c(x).$$

$$S_{x^\star}(x') := \begin{cases} \max_{c \neq c^\star} F_c(x') - F_{c^\star}(x') & \text{(Untargeted)} \\ F_{c^\dagger}(x') - \max_{c \neq c^\dagger} F_c(x') & \text{(Targeted)} \end{cases}$$

$$\phi_{x^\star}(x') := \operatorname{sign}\left(S_{x^\star}(x')\right) = \begin{cases} 1 & \text{if } S_{x^\star}(x') > 0, \\ -1 & \text{otherwise.} \end{cases}$$

$$\min_{x'} d(x', x^\star) \quad \text{such that} \quad \phi_{x^\star}(x') = 1,$$

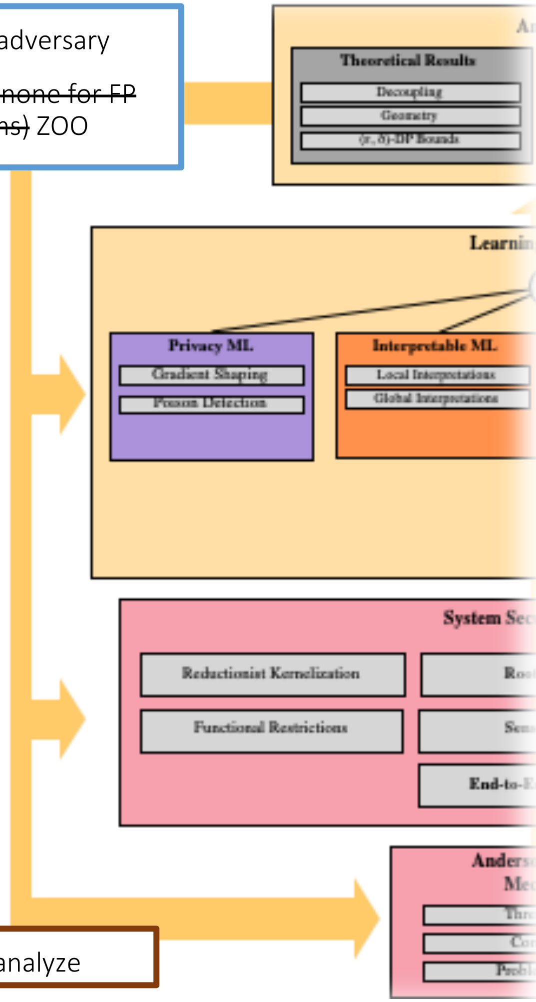
Hard label updates based on gradient approximation

$$\tilde{x}_t := x_t + \xi_t v_t(x_t, \delta_t), \text{ such that}$$

$$v_t(x_t, \delta_t) = \begin{cases} \widehat{\nabla S}(x_t, \delta_t) / \|\widehat{\nabla S}(x_t, \delta_t)\|_2, & \text{if } p = 2, \\ \operatorname{sign}(\widehat{\nabla S}(x_t, \delta_t)), & \text{if } p = \infty, \end{cases}$$

$$\widehat{\nabla S}(x_t, \delta) := \frac{1}{B-1} \sum_{b=1}^{B} (\phi_{x^\star}(x_t + \delta u_b) - \overline{\phi_{x^\star}}) u_b.$$

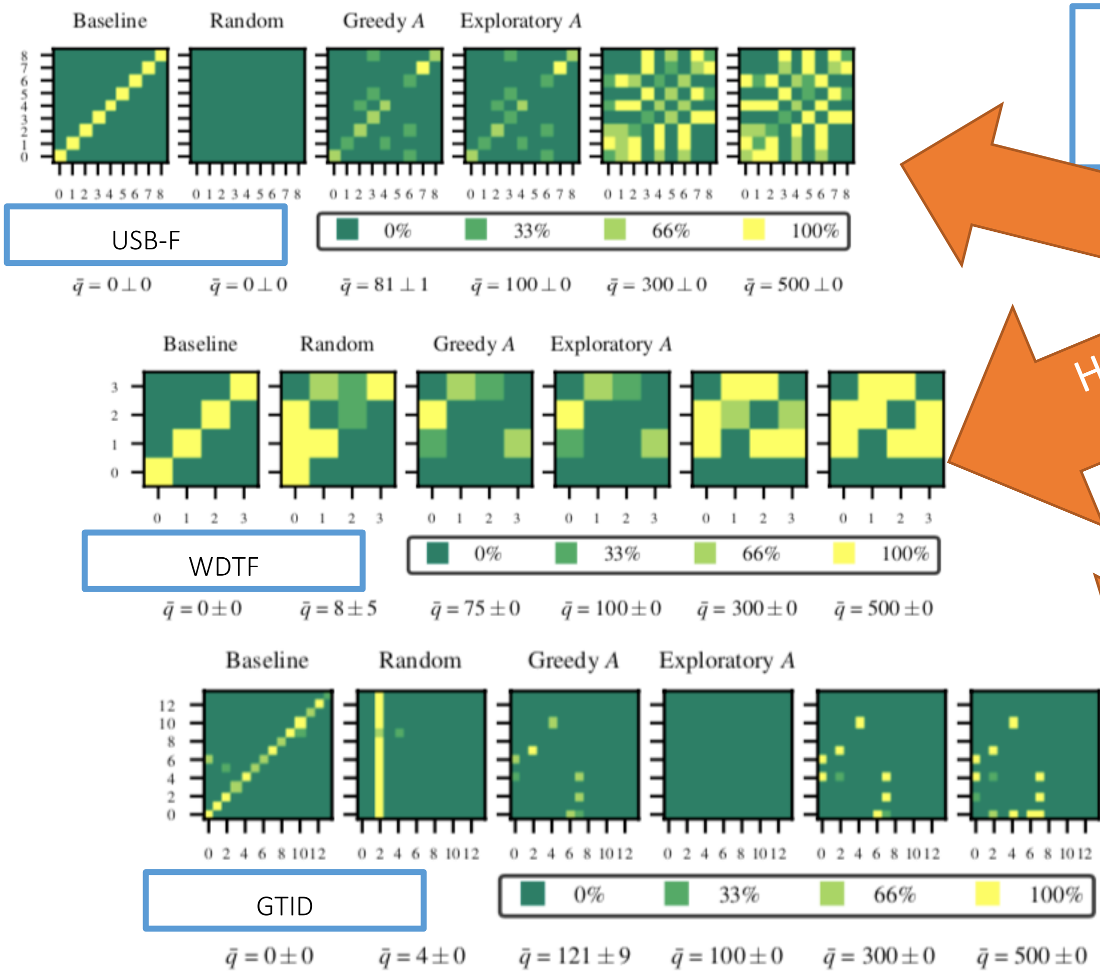
Re-analyze



Instantiation of model

Example using:

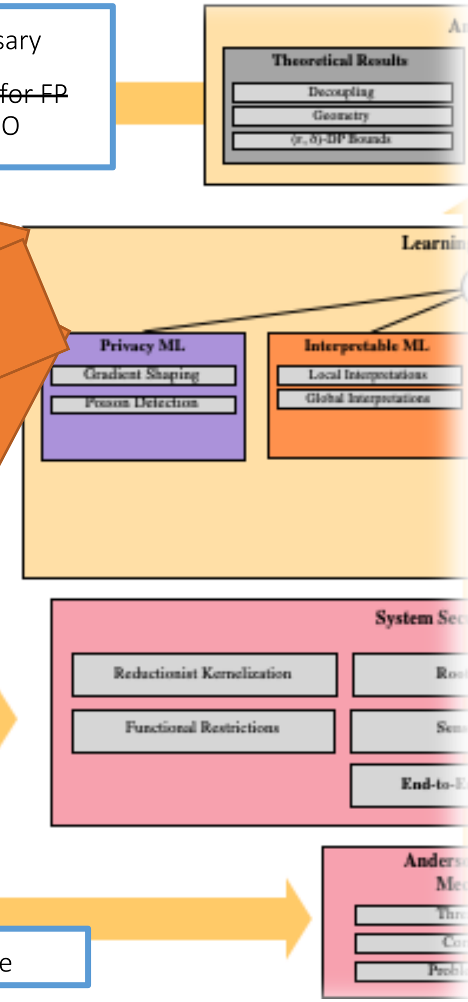
“Brittle Features of Device Authentication”



Modify adversary

~~(originally none for FP systems)~~ ZOO

High False Positives

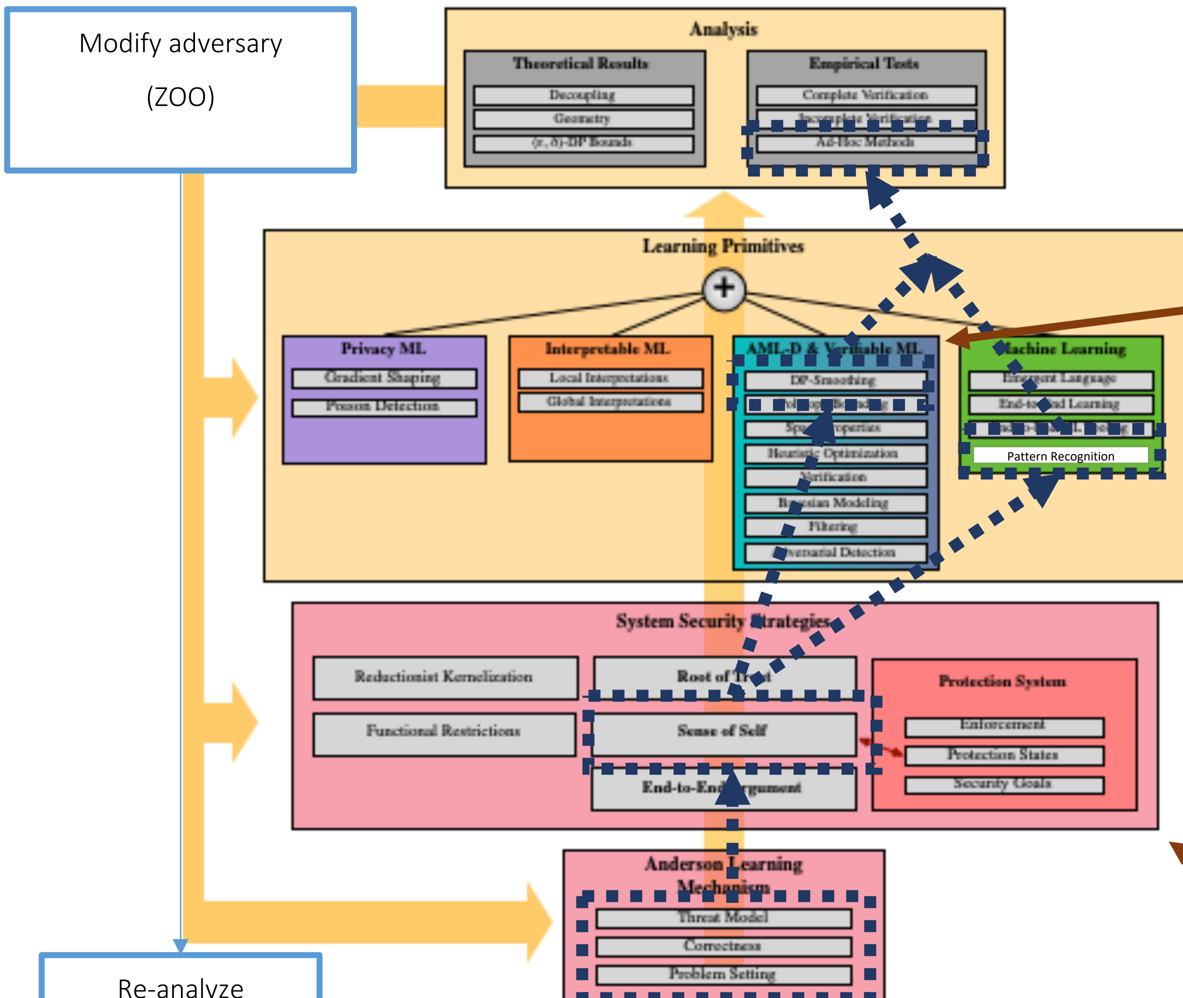


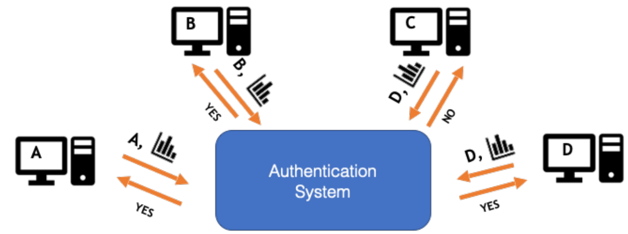
Re-analyze

Instantiation of model

Example using:

"Brittle Features of Device Authentication"





DP-based Randomized Smoothing Defense (Cohen et al.)

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\varepsilon) = c)$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$

Modifies the system's sense of self

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Instantiation of model

Example using:

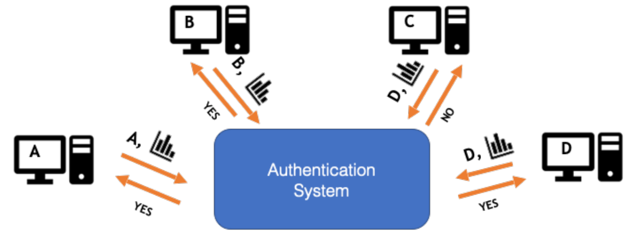
"Brittle Features of Device Authentication"

Modify adversary

(ZOO)

GTID

Re-analyze



DP-based Randomized Smoothing Defense (Cohen et al.)

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\varepsilon) = c)$$

$$\text{where } \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Modifies the system's sense of self

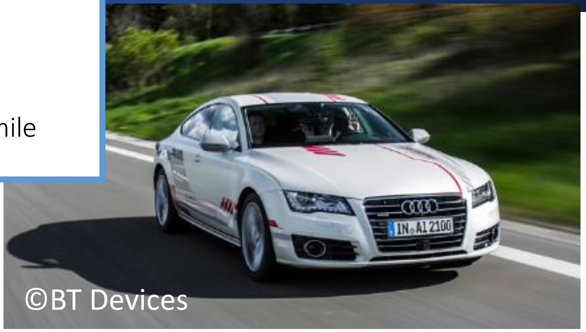
Instantiation of model

Example using:

”Jack” Audi automated driving system

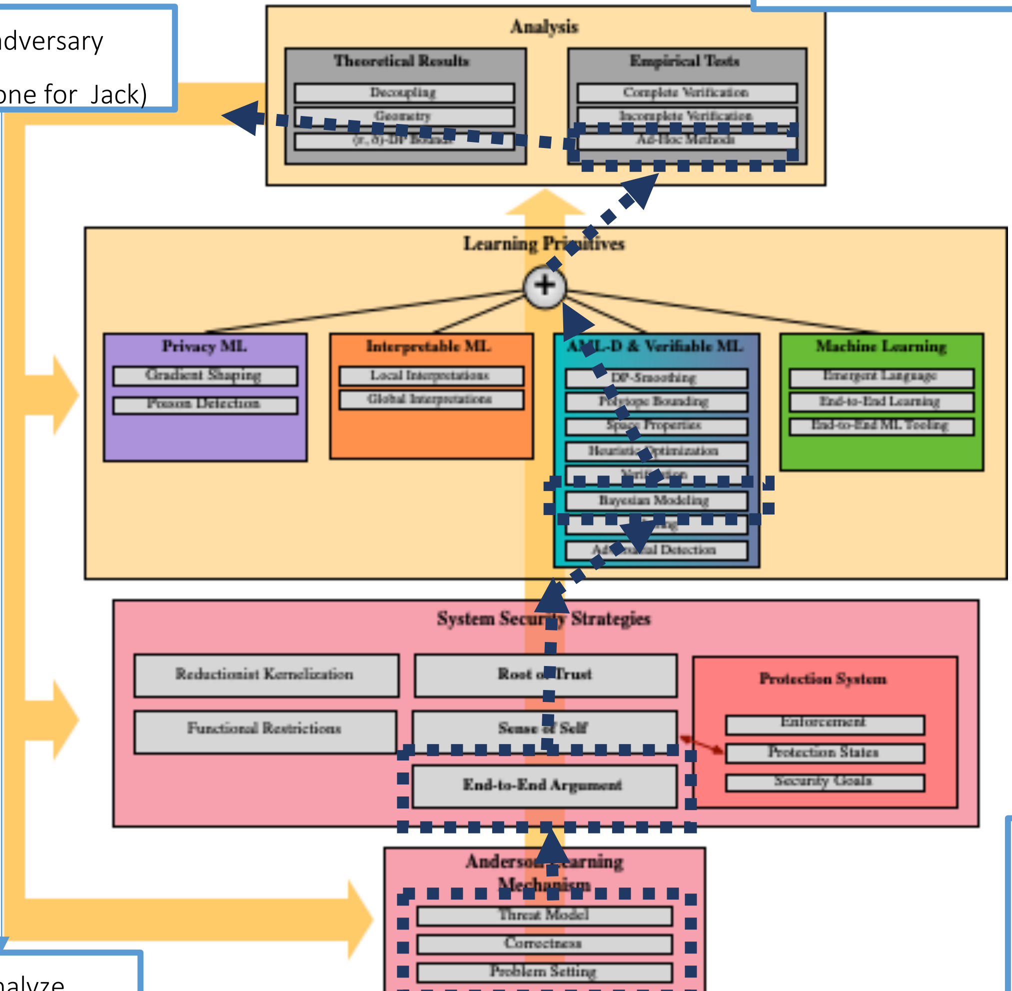
Analysis:

- Ad-hoc empirical test.
- Example: Drive a 550-mile automated test drive.



Modify adversary

(originally none for Jack)



Re-analyze

The system uses end-to-end argument to:

- Offer robustness, through Bayesian modeling: probabilistic measure of uncertainty between inputs and outputs.
- Example: Jack can perform probabilistic reasoning from perception sensor uncertainties

The system captures:

- Saltzer et al. end-to-end agreements (i.e., measures of agreement between inputs and outputs)

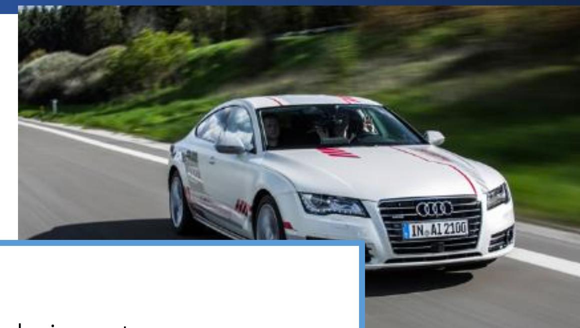
Establish likely problem setting (sensor inputs)

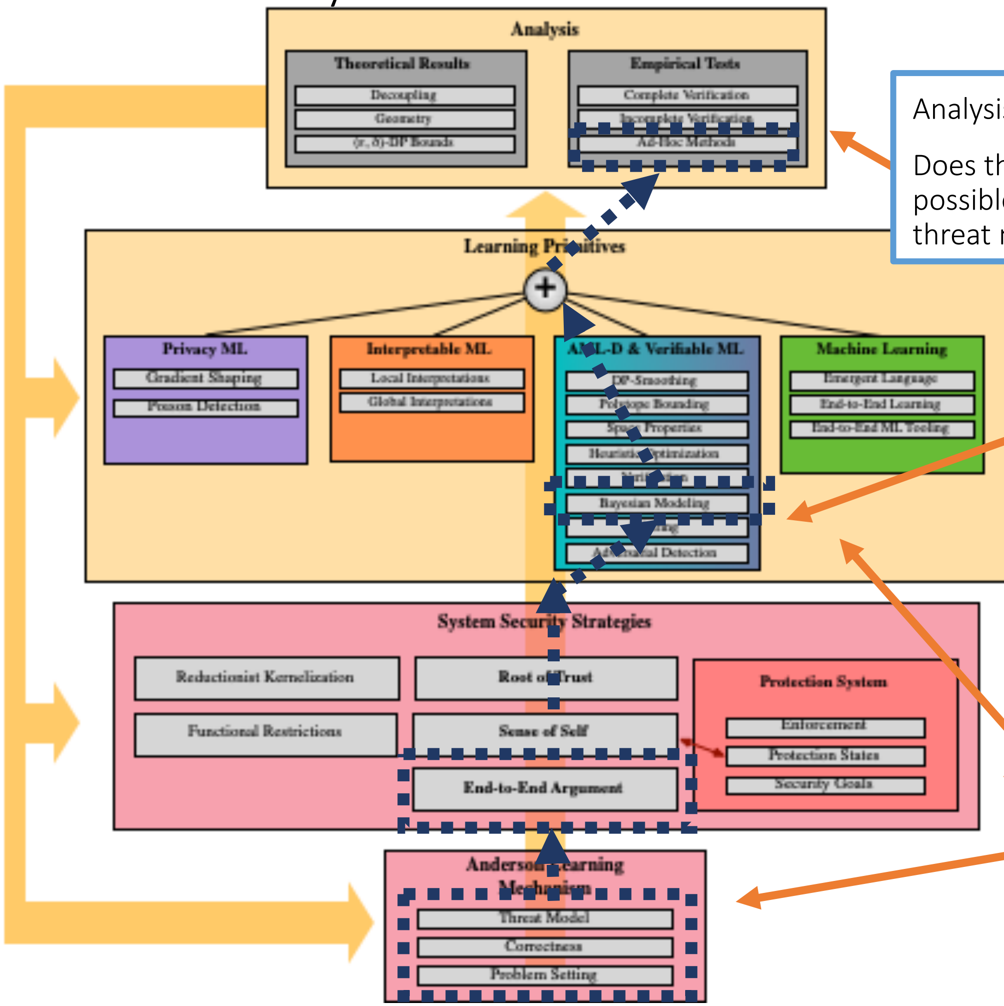
Conditions for correctness (crash avoidance)

Plausible threat model (in this case, innocent)

Instantiation of model

High-level feedback loop helps visualize gaps in constructions or analysis.





Analysis:

Does the analysis capture every possible failure mode in the current threat model?

Constructions:

Does the existing model of uncertainty completely capture each end of the system?

Structure informs of dependencies

If we change our threat model, we **must** re-visit our security strategies, primitives, and analysis.

Example: Analysis of Jack in the innocent setting says little about the adversarial setting

Next Steps

- Consider framework in the context of human-machine cooperation systems
- Verification of reinforcement learning agent policies
- Apply and implement on autonomous agents