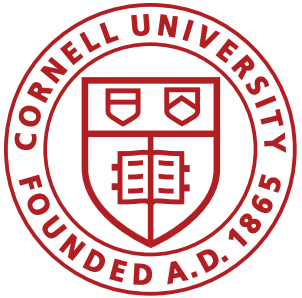


Stealthy Tracking of Autonomous Vehicles with Cache Side Channels

Mulong Luo, Andrew C. Myers, G. Edward Suh
Cornell University
29th USENIX Security Symposium 2020

Privacy in Cyber-Physical Systems (CPS)

- Autonomous vehicles and other CPS (e.g. UAV) are widely used
- Privacy in CPS involves in physical information (e.g., location)
- CPS as computer systems, are vulnerable to cache side-channel attack

Autonomous cars [1]

UAV [2]

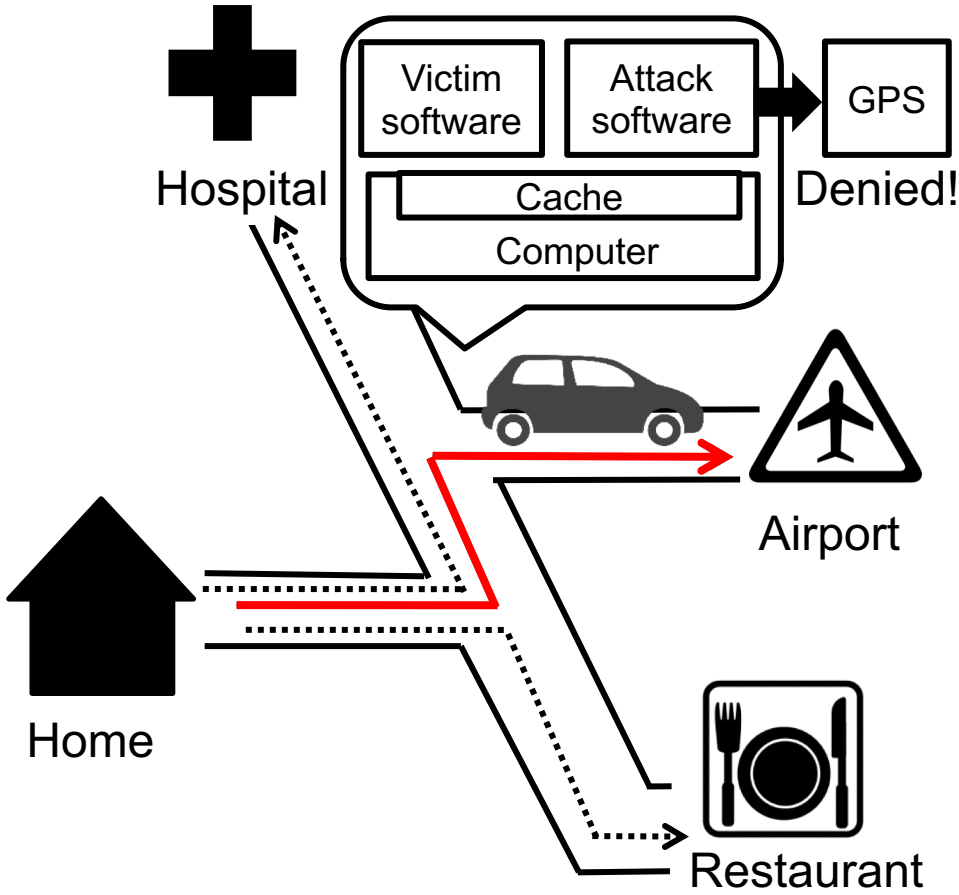
1. waymo.com
2. dji.com

Main Contributions

- We identified CPS as victims that can expose physical privacy information
  - Traditional victims of cache side-channel attack usually leaks non-physical information (e.g., crypto keys)
- We showed adaptive algorithms whose computation depends on inputs in CPS are vulnerable to cache side-channel attack
- We demonstrated a concrete example of cache side-channel attack on autonomous vehicles for route prediction

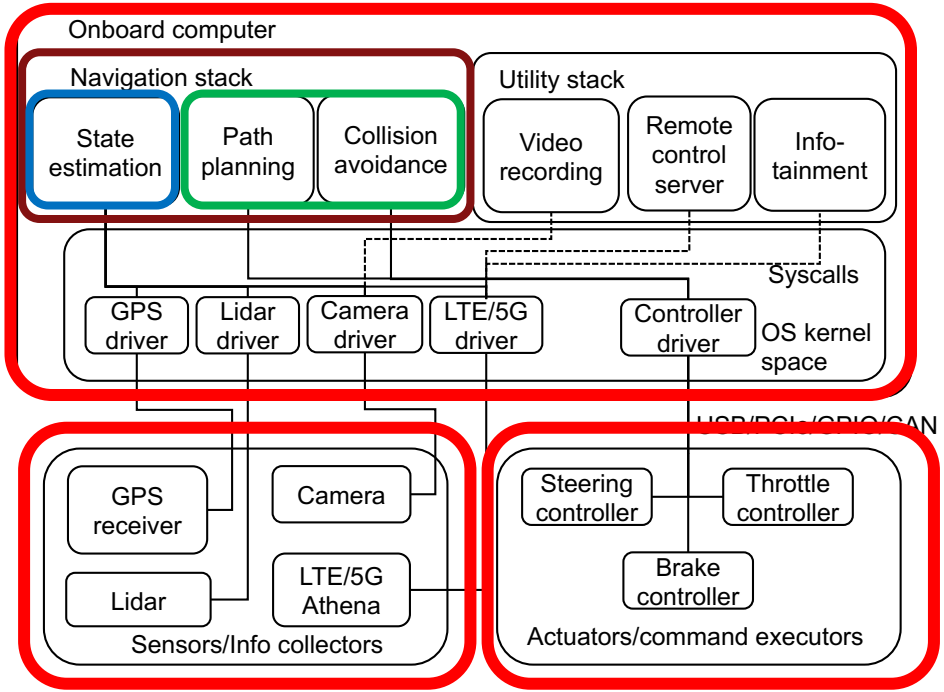
Threat Model

- Attack software shares computer with victim software on the vehicle under attack
- Attack software has no permission to access location service (e.g., GPS)
- Attack software uses cache side-channel attack to infer the route of the vehicle from a set of known routes



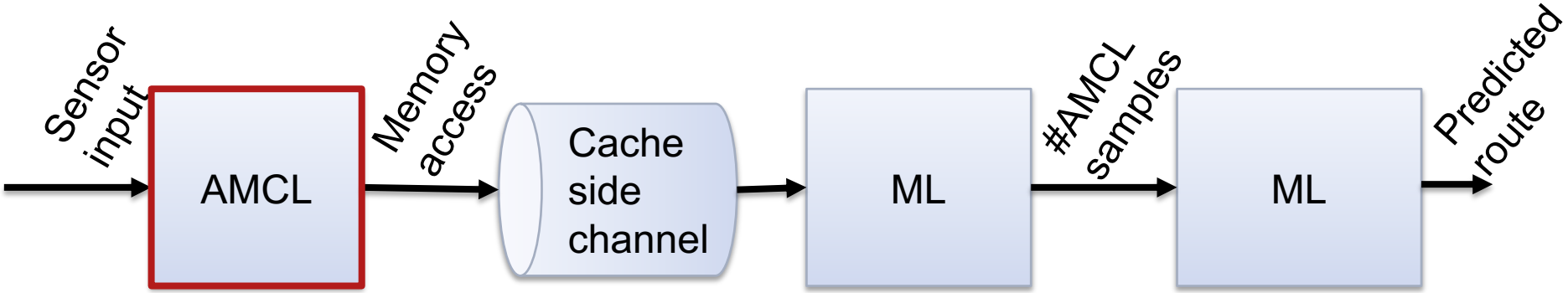
Autonomous Vehicle Systems

- Main components
  - Sensors
  - Actuators
  - Computer
- Major navigation tasks
  - Estimation/localization software
  - Control/decision software



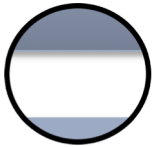
Attack Outline

- Adaptive Monte-Carlo Localization (AMCL): memory access pattern is affected by the physical sensor inputs
- Cache side-channel attack is used to extract the memory access patterns
- Machine learning (ML) models are used to infer the number of AMCL samples and the route of the vehicle

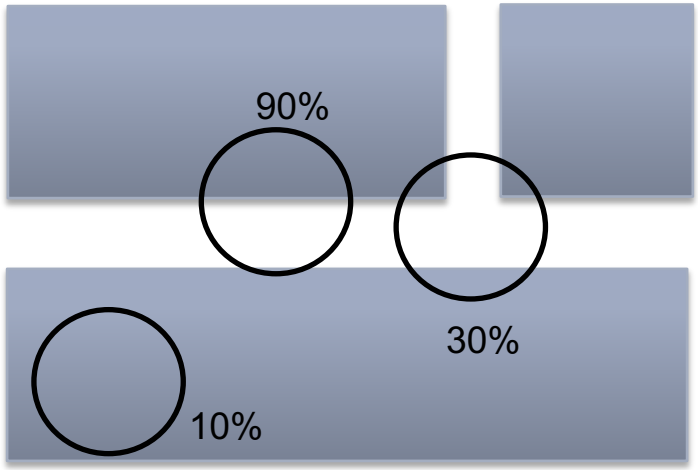


Adaptive Monte-Carlo Localization (AMCL)

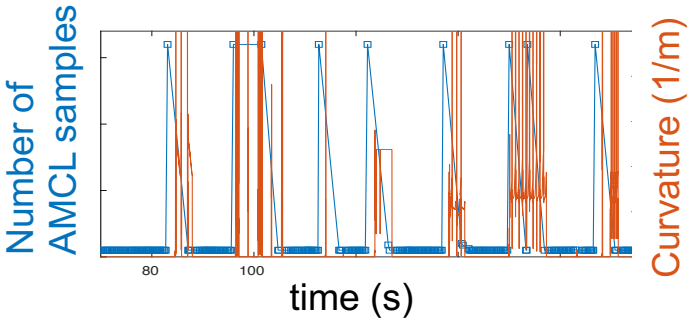
- Input: sensor image, and a map
- Output: location on the map
- Process:
  - Randomly sample locations on the map, calculate the similarity with the sensor image
  - Use k-means to find cluster center
- Adaptivity: the number of samples varies adaptively depending on uncertainty[1]
  - High curvature (uncertainty) → more AMCL samples → more memory access



Sensor image



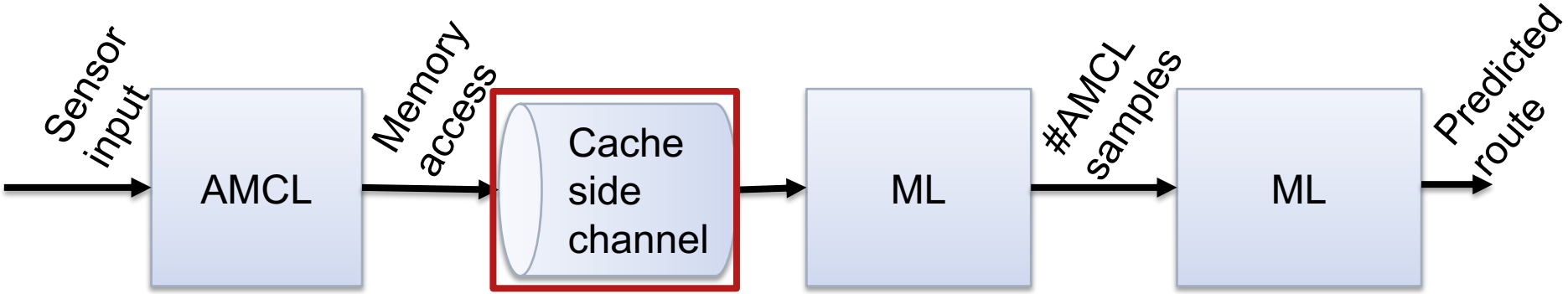
map



1.D. Fox. Adapting the sample size in particle filters through KLD-sampling. IJRR, 22(12):985–1003, 2003.

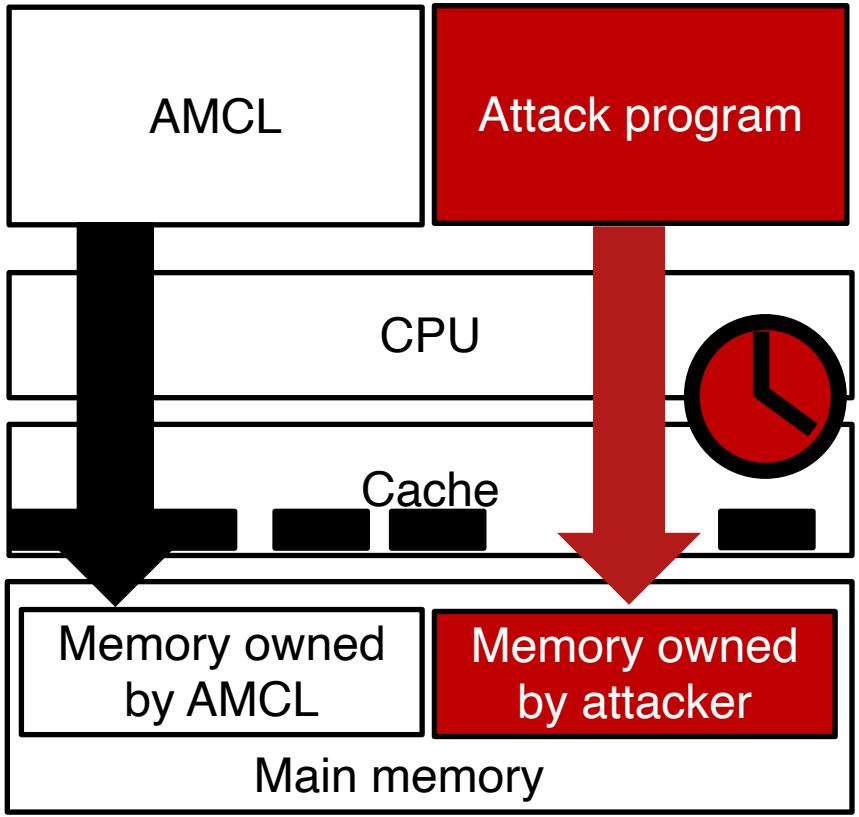
Attack Outline

- Adaptive Monte-Carlo Localization (AMCL): memory access pattern is affected by the physical sensor inputs
- Cache side-channel attack is used to extract the memory access patterns
- Machine learning (ML) models are used to infer the number of AMCL samples and the route of the vehicle



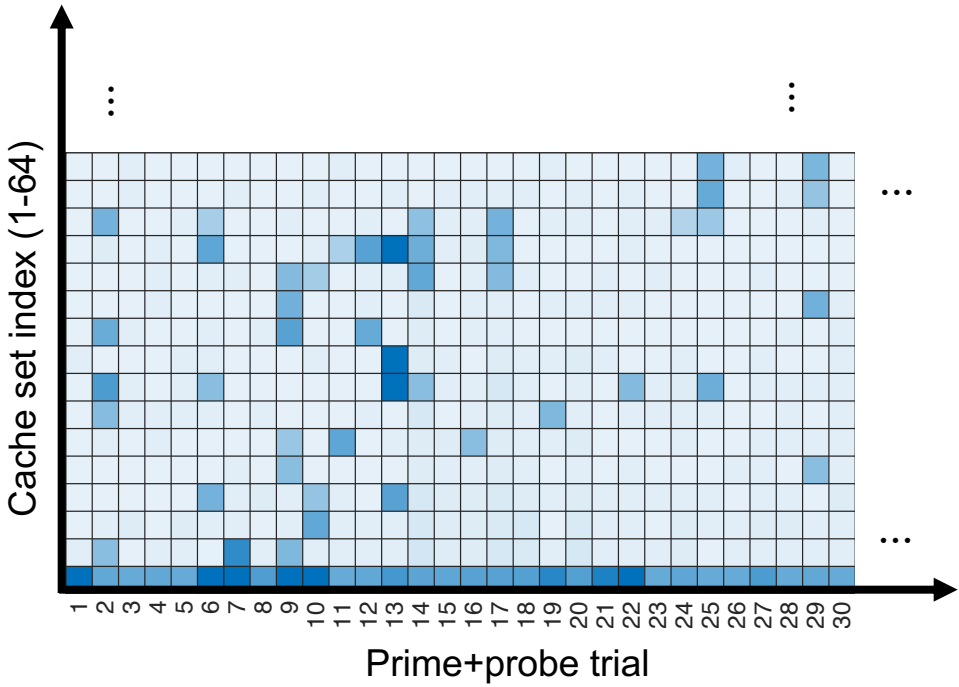
Cache Side-Channel Attack

- Sharing of cache
- Memory accesses patterns from AMCL are left in the cache
- Attack program access the cache and infer AMCL accesses through timing
  - Prime+probe



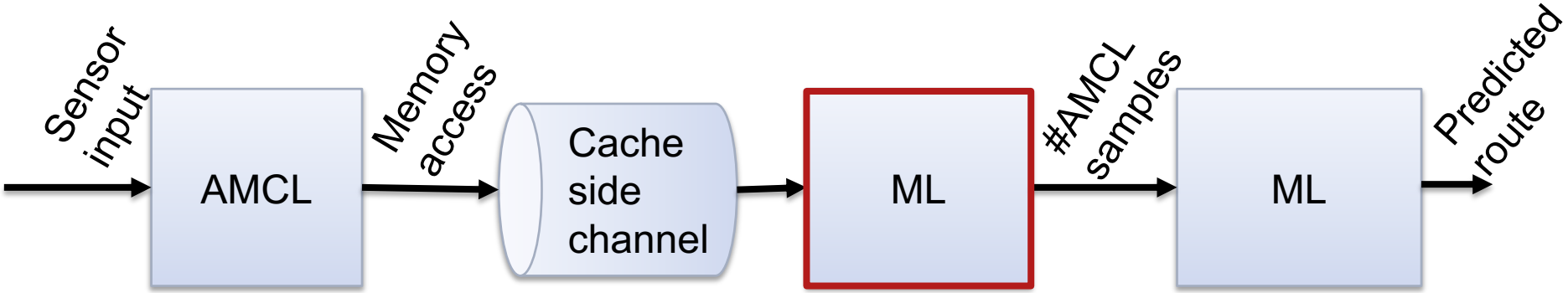
Cache Prime+Probe

- For each prime+probe trial
  - Probe each cache set and record the probing time
  - High probing time (dark cell) → cache set accessed by a victim
  - #dark cells → #AMCL samples
- ML model to predict number of samples



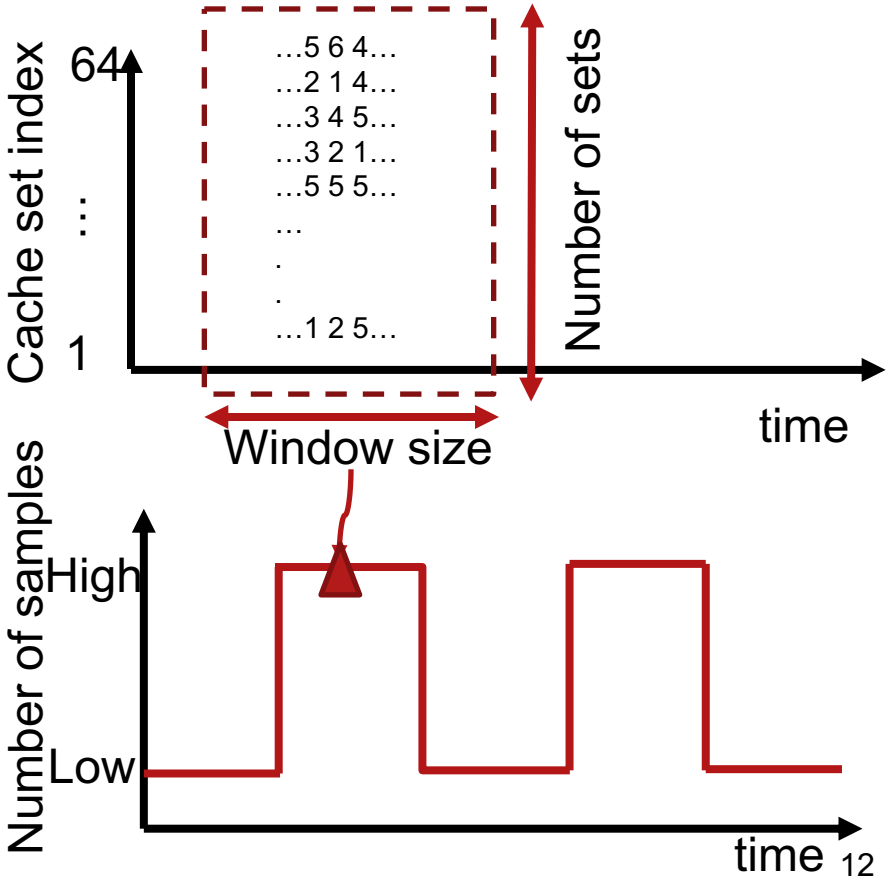
Attack Outline

- Adaptive Monte-Carlo Localization (AMCL): memory access pattern is affected by the physical sensor inputs
- Cache side-channel attack is used to extract the memory access patterns
- Machine learning (ML) models are used to infer the number of AMCL samples and the route of the vehicle



ML Model: #AMCL Samples Predictor

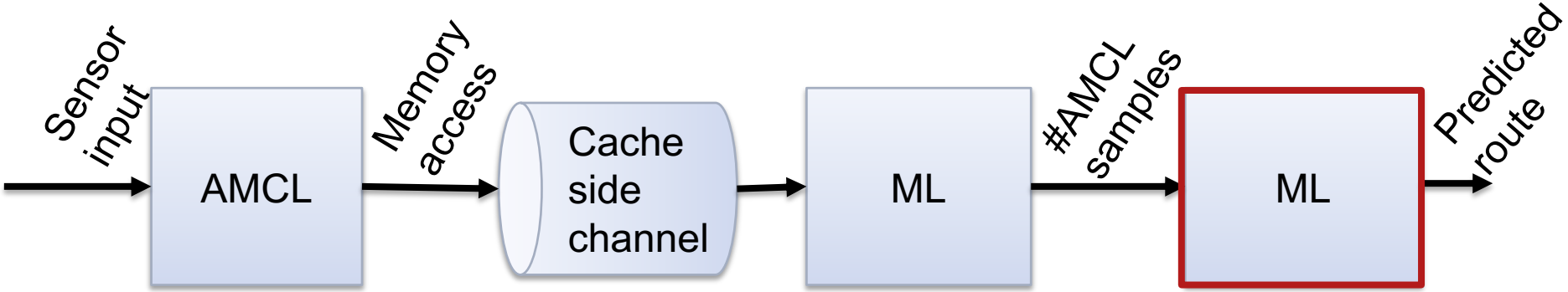
- Formulated as a binary classification problem
  - Input: a window of cache probing time matrix
  - Output: two classes {High, Low} representing the number of samples
  - RUSBoost[1] binary predictor is used



1.C. Seiffert, et al., "RUSBoost: Improving classification performance when training data is skewed," *2008 ICPR.*

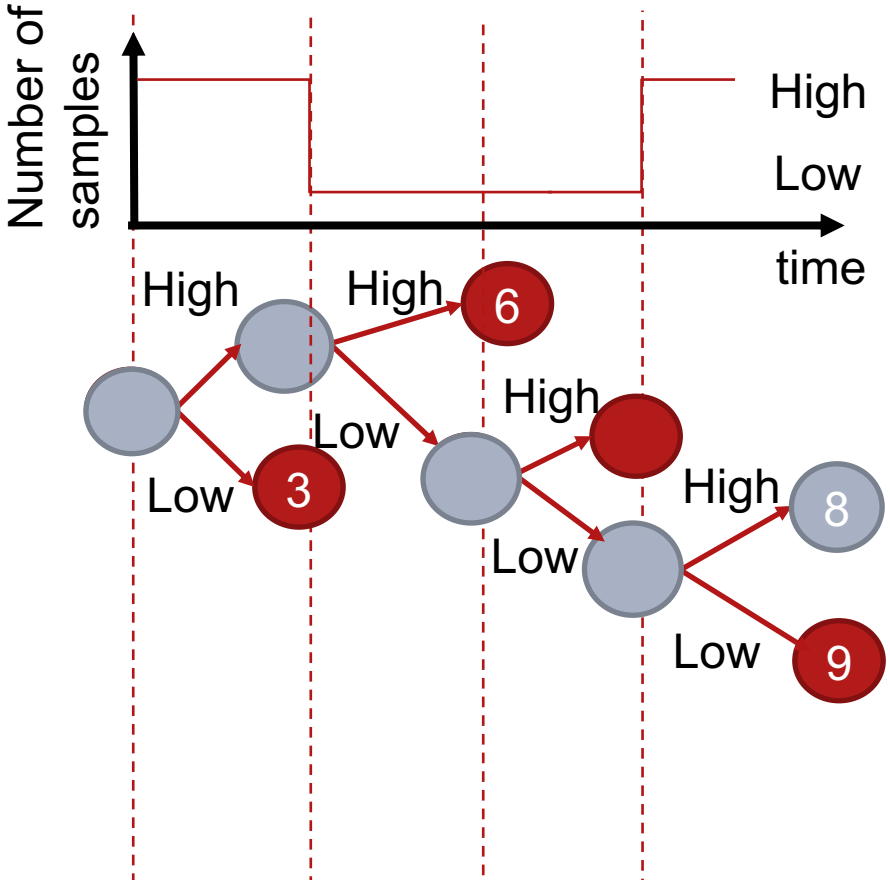
Attack Outline

- Adaptive Monte-Carlo Localization (AMCL): memory access pattern is affected by the physical sensor inputs
- Cache side-channel attack is used to extract the memory access patterns
- Machine learning (ML) models are used to infer the number of AMCL samples and the route of the vehicle



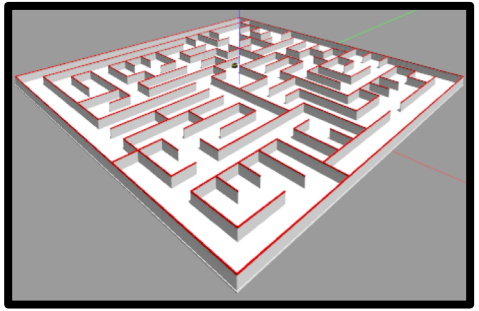
ML Model: Route Predictor

- Formulated as a classification problem
  - Input: a series of {High, Low} which indicates the number of samples
  - Output: label of the route
  - Decision tree/forest is used

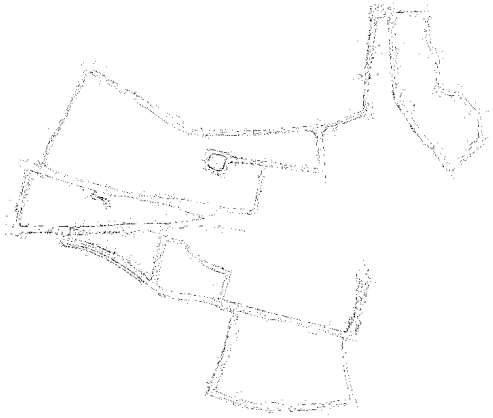


Evaluation: Datasets

- Two datasets used
  - Gazebo/ROS[1]
    - Simulated robot in a maze
  - Oxford[2]
    - Data collected on a car driving around Oxford, UK
- Processors for evaluation
  - Intel i5-3317u, and Intel Xeon E5-1270 (recommended by Apollo[3])



Gazebo/ROS simulation maze

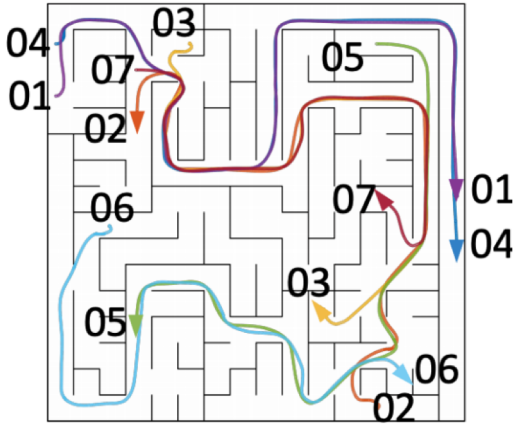


Oxford

1. http://gazebosim.org/
2. https://robotcar-dataset.robots.ox.ac.uk/
3. https://apollo.auto

Evaluation: Route Prediction - Gazebo

- 7 randomly-selected routes with similar lengths
- Perform route prediction using L1D or LLC probing data
  - Accuracy: L1D: 81.4%, LLC 75%



Maze and 7 routes

Predicted \ GroundTruth	01	02	03	04	05	06	07
01	18	0	0	0	0	1	0
02	0	17	1	0	6	0	0
03	0	0	19	0	1	0	0
04	0	0	0	18	0	2	2
05	0	3	0	0	13	0	0
06	1	0	0	1	0	15	4
07	1	0	0	1	0	2	14

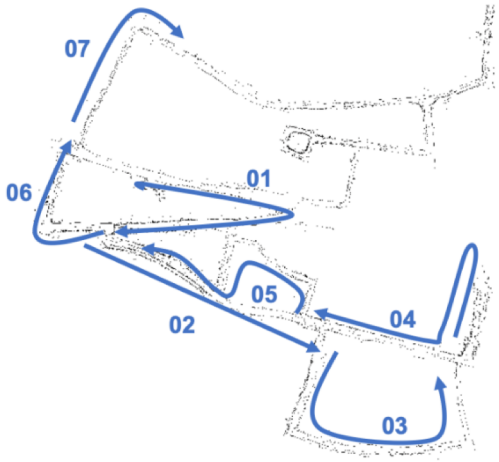
L1D prediction results

Predicted \ GroundTruth	01	02	03	04	05	06	07
01	16	3	0	6	0	1	0
02	0	15	2	1	1	2	0
03	0	0	16	0	3	1	0
04	4	1	0	13	0	0	0
05	0	0	2	0	14	1	3
06	0	1	0	0	1	15	1
07	0	0	0	0	1	0	16

LLC prediction results

Evaluation: Route Prediction - Oxford

- 7 routes with similar lengths from the Oxford dataset
- Perform route prediction using L1D or LLC probing data
  - Accuracy: L1D: 74.6%, LLC 73.0%



Oxford dataset and seven routes

Predicted \ GroundTruth	01	02	03	04	05	06	07
01	9	0	0	2	3	0	1
02	0	16	7	0	1	1	0
03	0	0	9	0	0	0	0
04	6	0	0	16	0	0	0
05	4	1	0	0	13	0	0
06	0	2	2	0	1	14	1
07	0	0	0	0	0	2	15

L1D prediction results

Predicted \ GroundTruth	01	02	03	04	05	06	07
01	13	0	0	2	1	0	0
02	0	16	6	0	0	1	0
03	1	0	9	1	0	2	0
04	3	0	0	15	1	0	0
05	2	1	0	0	15	0	0
06	0	2	3	0	0	11	2
07	0	0	0	0	1	3	15

LLC prediction results

More In the Paper…

- Evaluation: Host-client testbed architecture (Section 5.1.1)
- Location prediction (Section 5.3.2)
- L1D & LLC comparison (Section 5.3.3)
- Discussion (Section 6)
  - Processor architecture (Section 6.1)
  - Generality of the vulnerability (Section 6.2)
  - Limitations of the attack model (Section 6.3)
  - Difficult-to-predict routes (Section 6.4)

Conclusion

- We identified CPS as victim that can expose physical privacy information
- We showed adaptive algorithms in CPS are vulnerable to cache side-channel attack
- We demonstrated a concrete example of cache side-channel attack on autonomous vehicles for route prediction
- Cache side-channel protection necessary for strong CPS physical privacy

Thanks for listening!

ml2558@cornell.edu