Attack-Resilient Cyber-Physical Systems

Yasser Shoukry

Assistant Professor
Resilient Cyber-Physical Systems Lab
Department of Electrical and Computer Engineering
University of Maryland, College Park

UNIVERSITY OF MARYLAND 18 56

A. JAMES CLARK SCHOOL OF ENGINEERING

Cyber-Physical Systems

Fully autonomous container terminal, Altenwerder, Germany. Siemens promotional video.

What can go wrong?

DARPA's High-Assurance Cyber Military Systems (HACMS)
(Interview with 60 minutes)

What can go wrong?

DARPA's High-Assurance Cyber Military Systems (HACMS)
(Interview with 60 minutes)

What can go wrong?

What can go wrong?

Although some characteristics are similar, ICS also have characteristics that differ from traditional information processing systems. Many of these differences stem from the fact that logic executing in ICS has a direct affect on the physical world. Some of these characteristics include significant risk to the health and safety of human lives and serious damage to the environment, as well as serious financial issues such as production losses, negative impact to a nation's economy, and compromise of proprietary information. ICS have unique performance and reliability requirements and often use operating systems and applications that may be considered unconventional to typical IT personnel. Furthermore, the goals of safety and efficiency sometimes conflict with security in the design and operation of control systems.

Cyber security = steal credit card, leak personal information, …

CPS security = loss of control in nuclear reactors, affecting transportation networks, …

Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)

Recommendations of the National Institute of Standards and Technology

---

Keith Stouffer
Joe Falco
Karen Scarfone

Threat Models

GPS/Sensor Spoofing Attacks

Software Vulnerabilities

Denial of Service Attacks

Privacy Leaks

Threat Models

GPS/Sensor Spoofing Attacks

Software Vulnerabilities

Denial of Service Attacks

Privacy Leaks

Outline

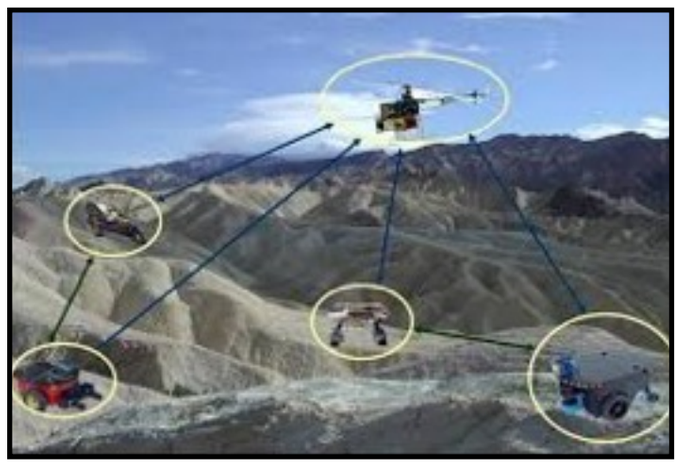
False Data Injection Attacks

TAC 2016, CDC 2017, ICCPS 2016
(Best paper award)

Sybil Attacks + False Data Injection

ICCPS 2018

Privacy-preserving Sensor Fusion + False Data Injection



CDC 2016, IPSN 2017
(Best demo award)

Outline

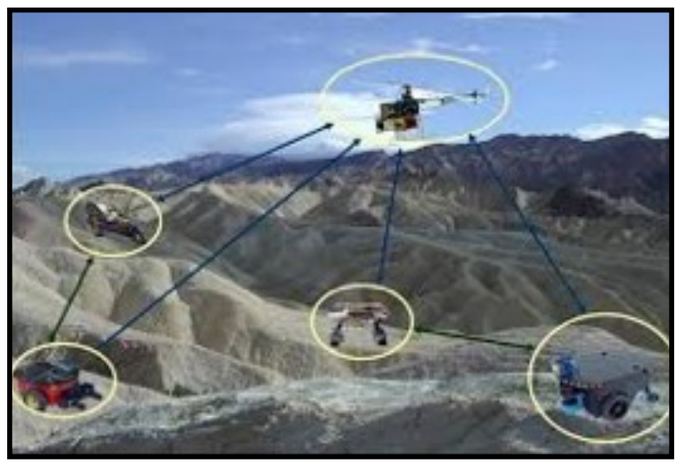
False Data Injection Attacks

TAC 2016, CDC 2017, ICCPS 2016
(Best paper award)

Sybil Attacks
+
False Data Injection

ICCPS 2018

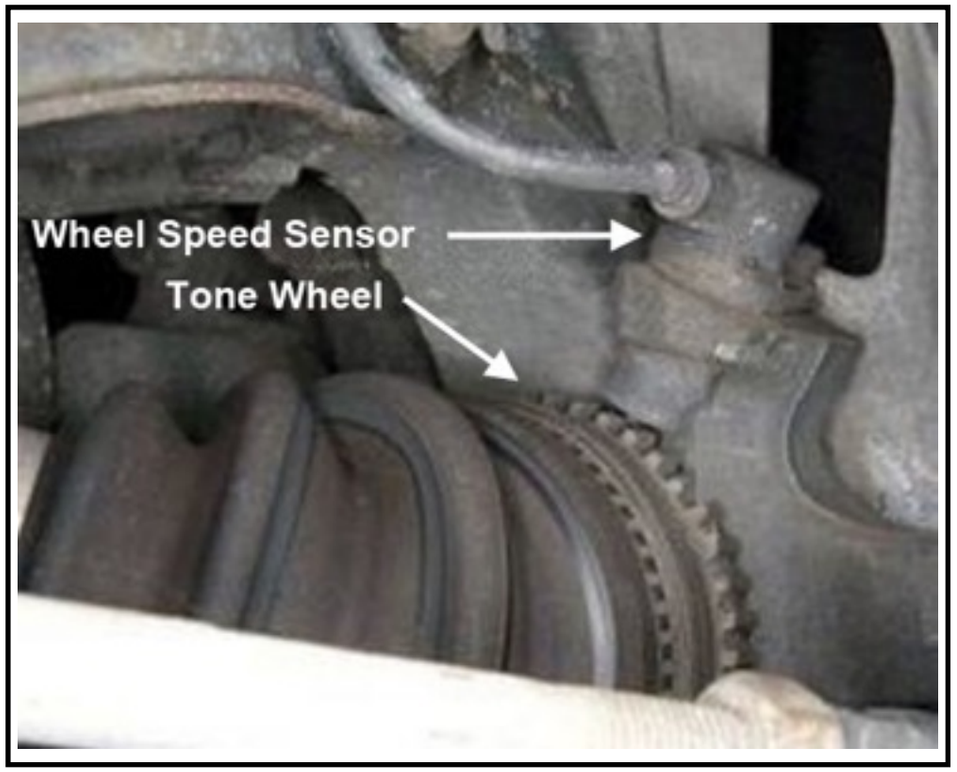
Privacy-preserving Sensor Fusion
+
False Data Injection

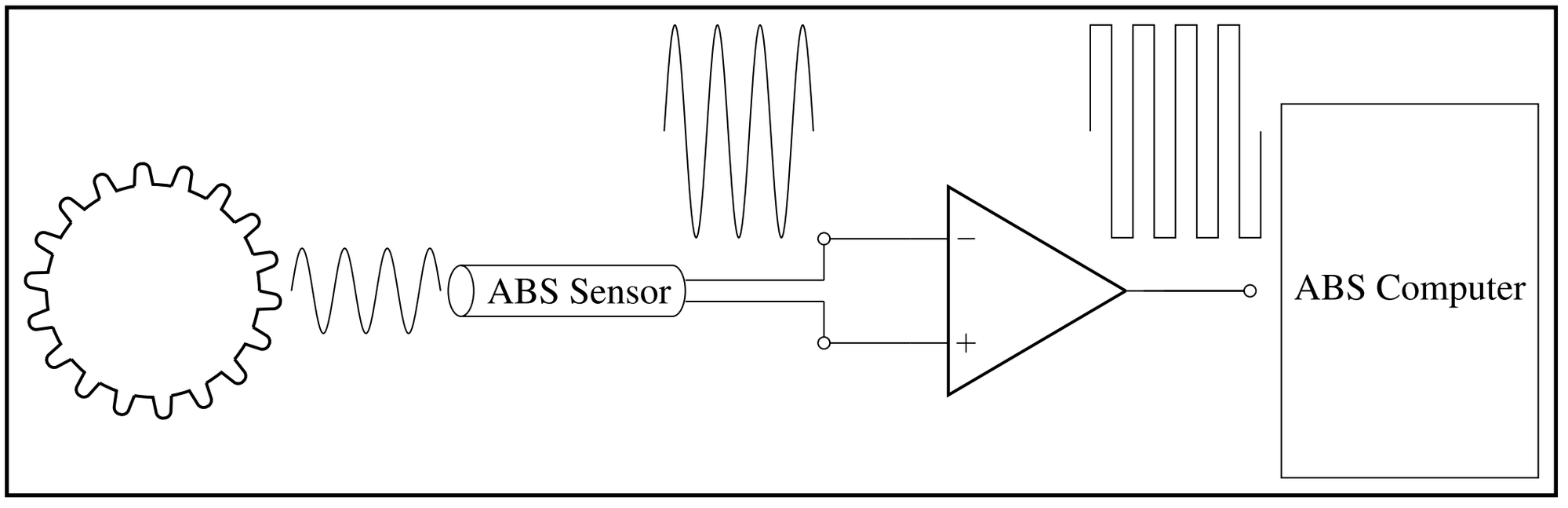


CDC 2016, IPSN 2017
(Best demo award)

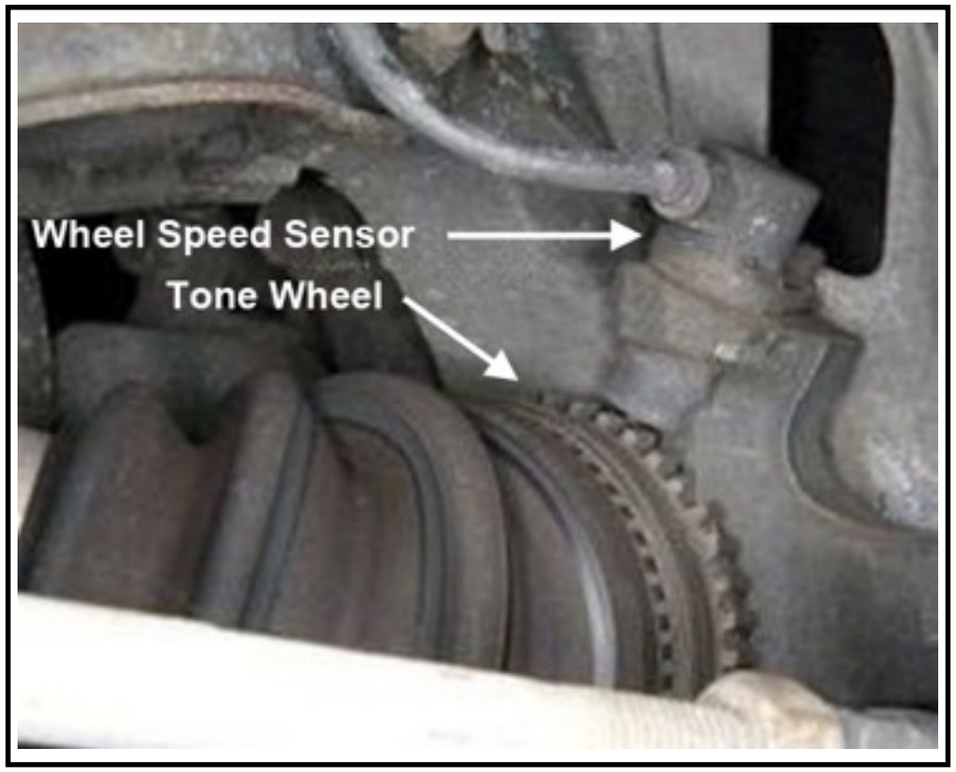
Noninvasive Sensor Spoofing Attacks

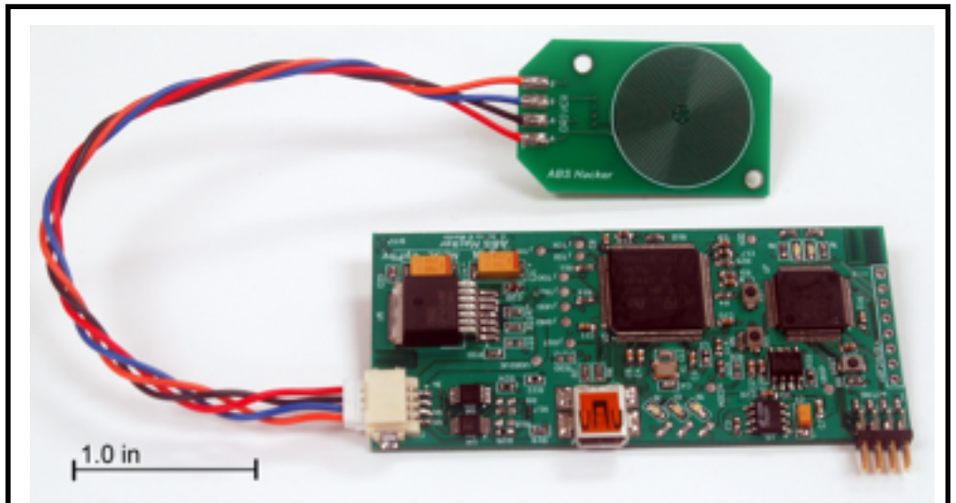
Noninvasive Sensor Spoofing Attacks

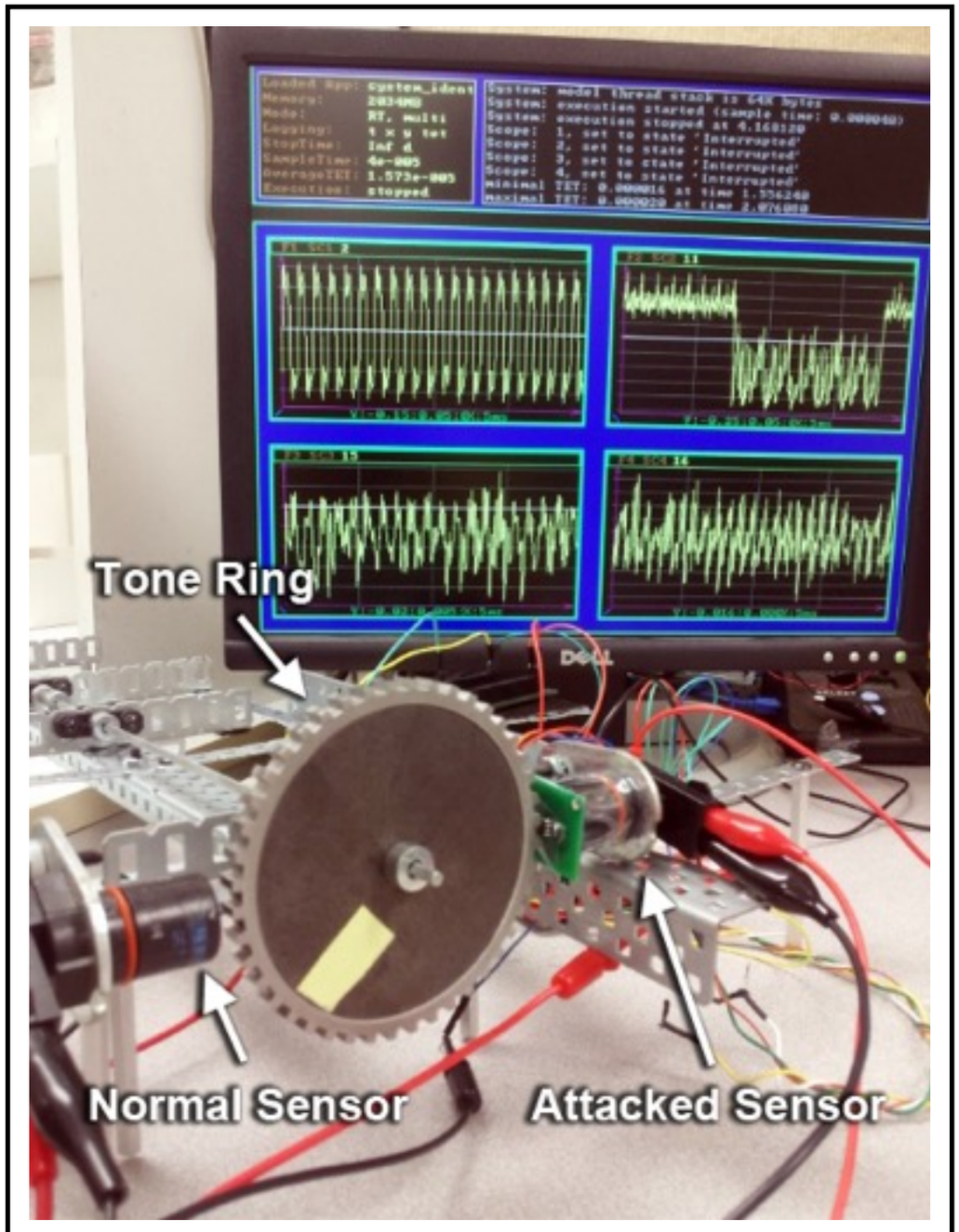




Noninvasive Sensor Spoofing Attacks

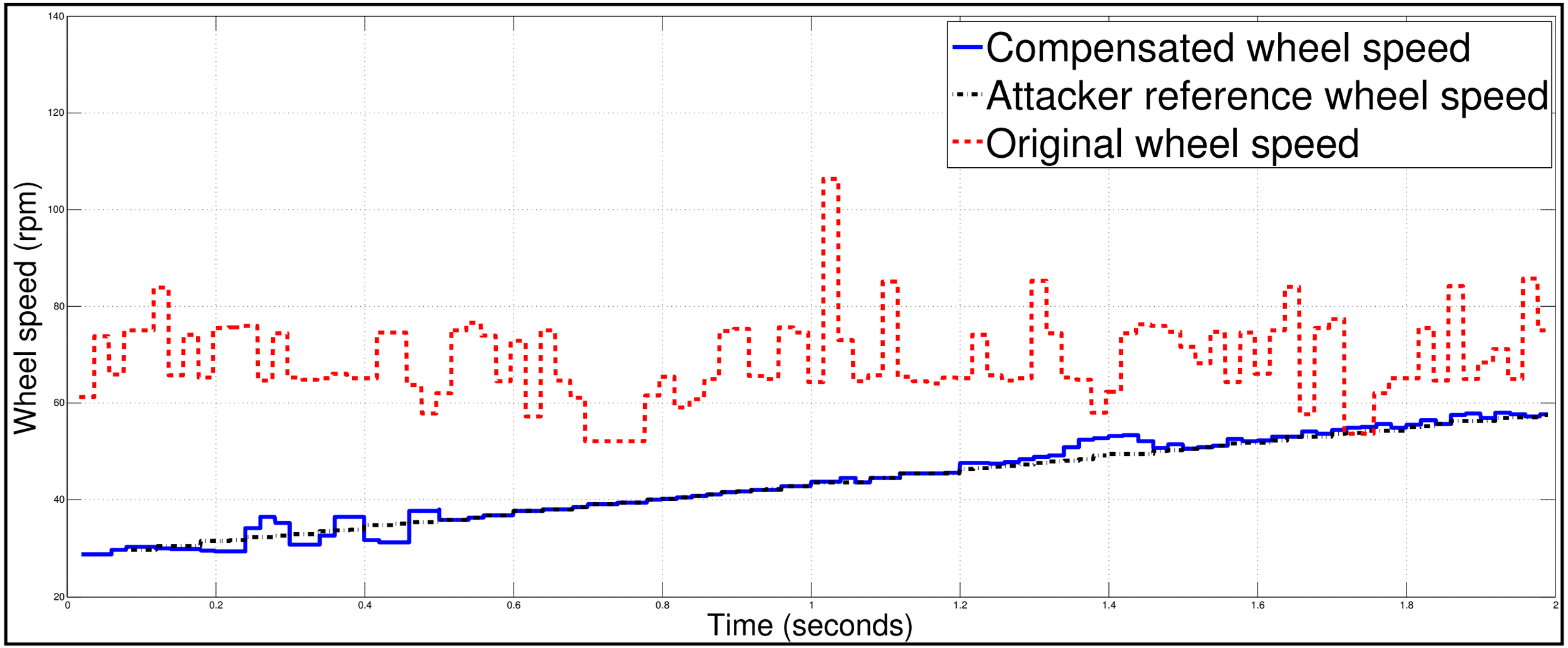






Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Noninvasive Spoofing Attacks for Anti-Lock Braking Systems," CHES 2013

Noninvasive Sensor Spoofing Attacks

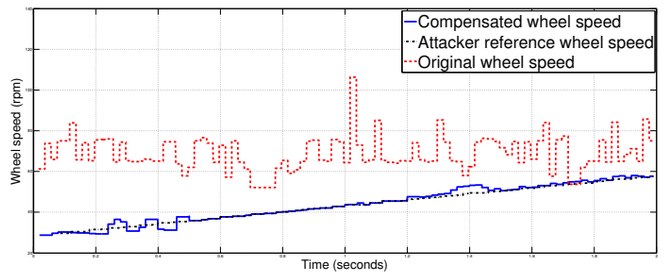


Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Noninvasive Spoofing Attacks for Anti-Lock Braking Systems," CHES 2013

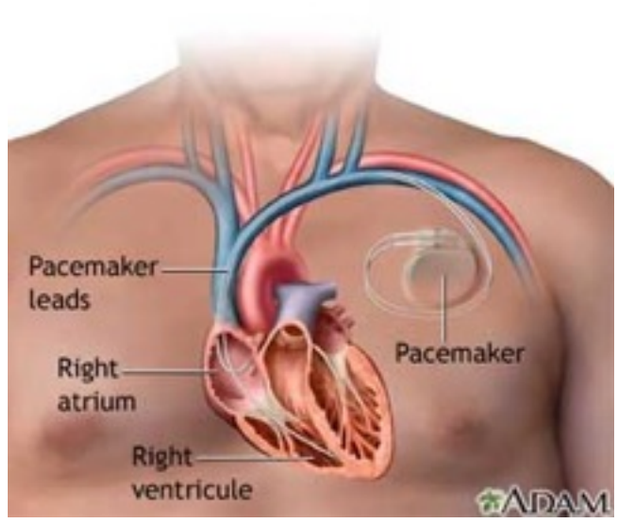
Noninvasive Sensor Spoofing Attacks

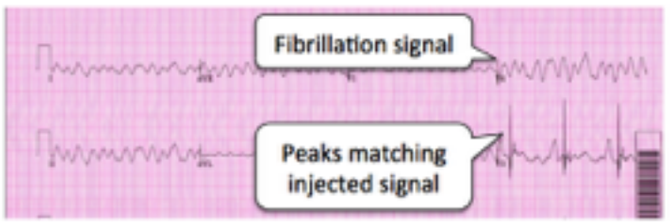
Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Noninvasive Spoofing Attacks for Anti-Lock Braking Systems," CHES 2013

False Data Injection Attacks

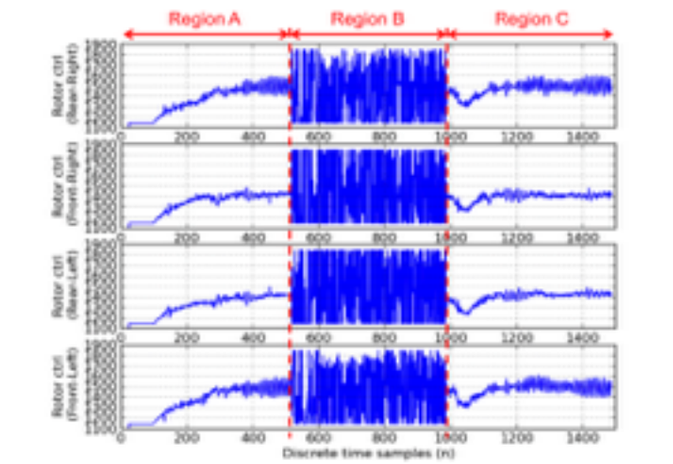


Y. Shoukry, et. al, "Noninvasive Spoofing Attacks for Anti-Lock Braking Systems," CHES 2013.





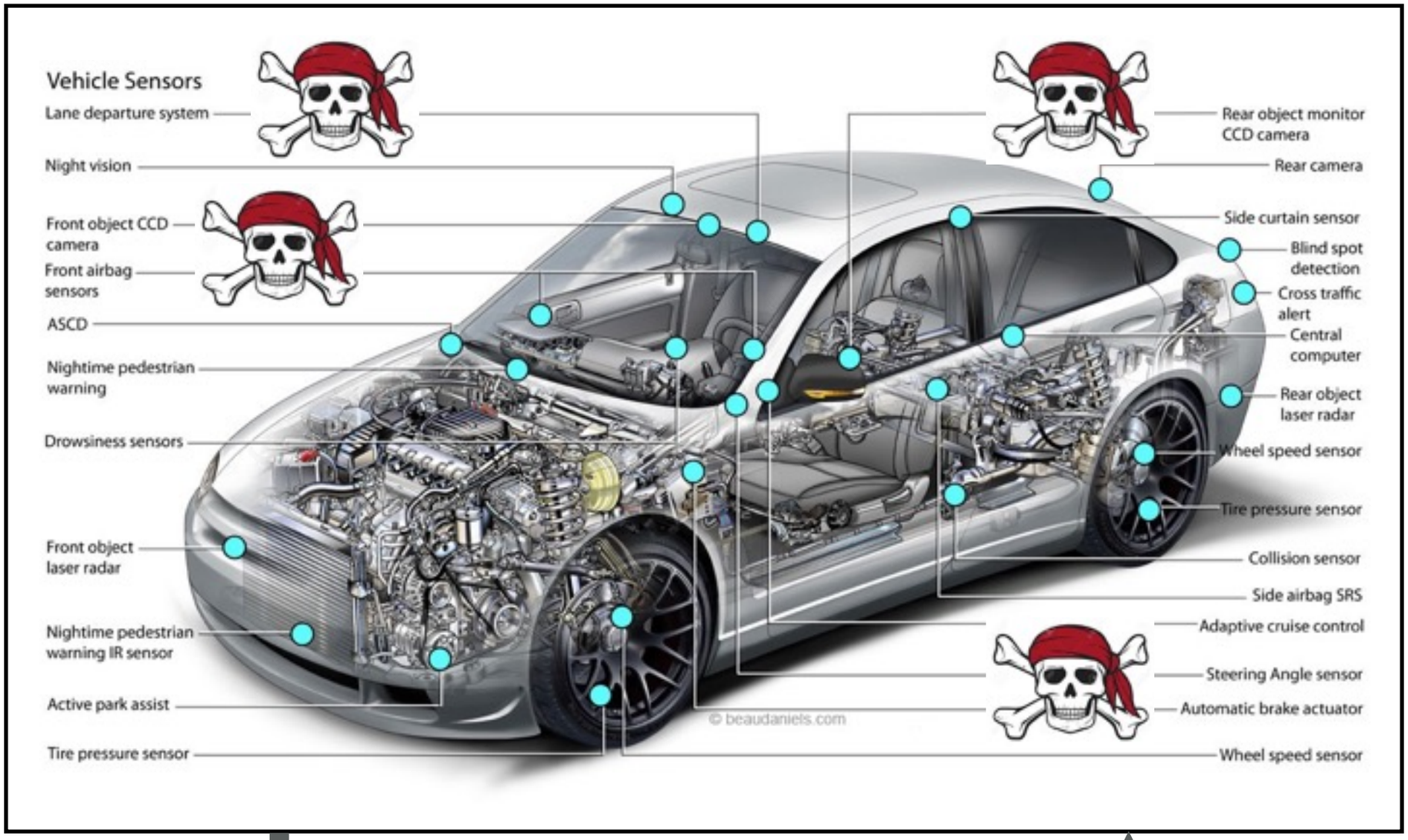
D. Kune, et. al, "Ghost Talk: Mitigating EMI Signal Injection Attacks against Analog Sensors," IEEE S&P 2013.



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

Traditional information-security offers no defense against these attacks!

Secure State Estimation



Secure State Estimator

Feedback Control Algorithm

Secure State Estimation

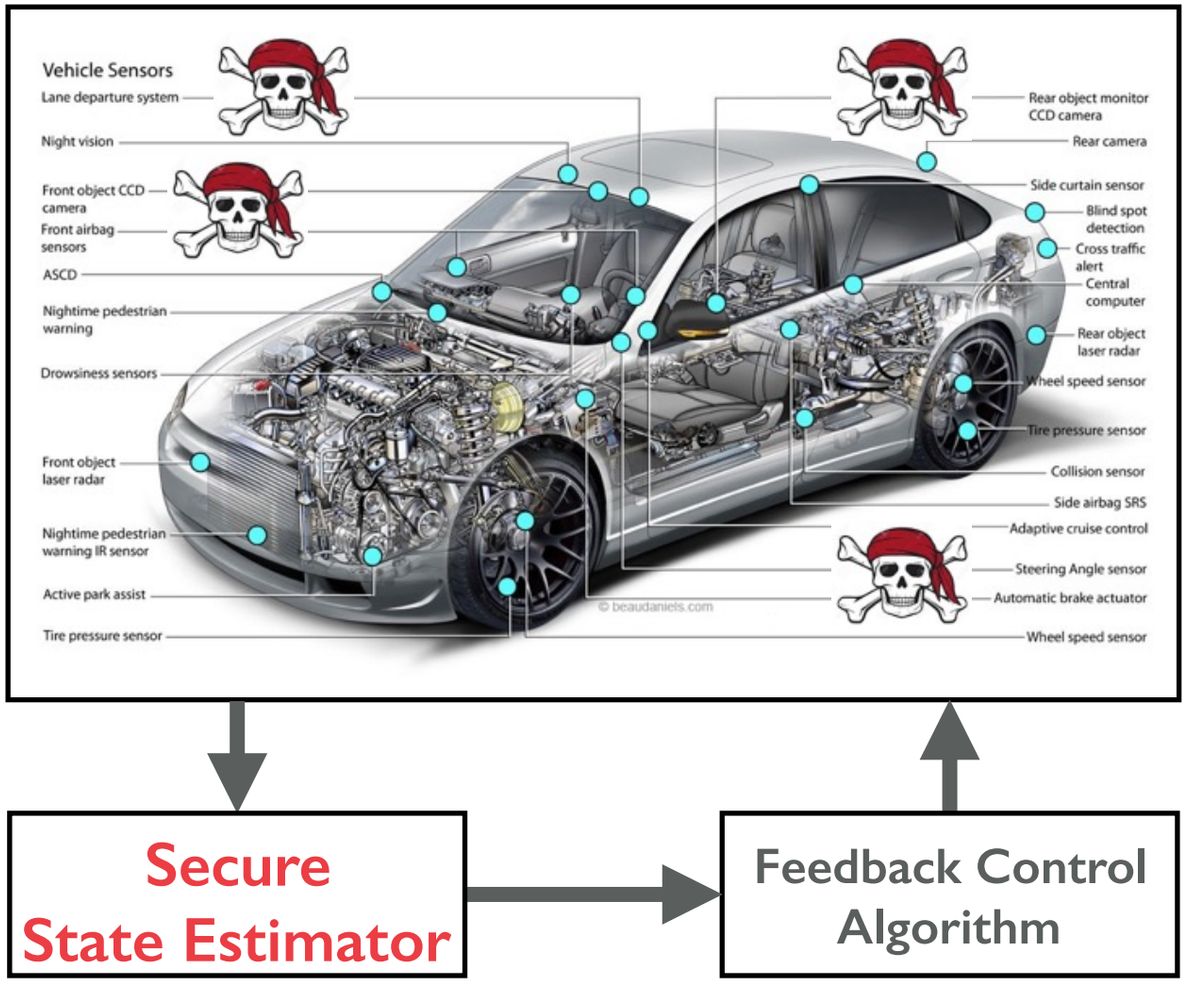
Threat Model:

- The attacker has compromised **"s" (out of "p")** sensors.

- **Sensor attacks:** any attack mechanism, e.g., sensor spoofing, communication channel, software virus, ….

- The attacker is free to corrupt **all/some/none** of the compromised sensors.

- The attack can be **arbitrary** (no boundedness assumption, no stochastic model, …).



Secure State Estimation

- Redundancy

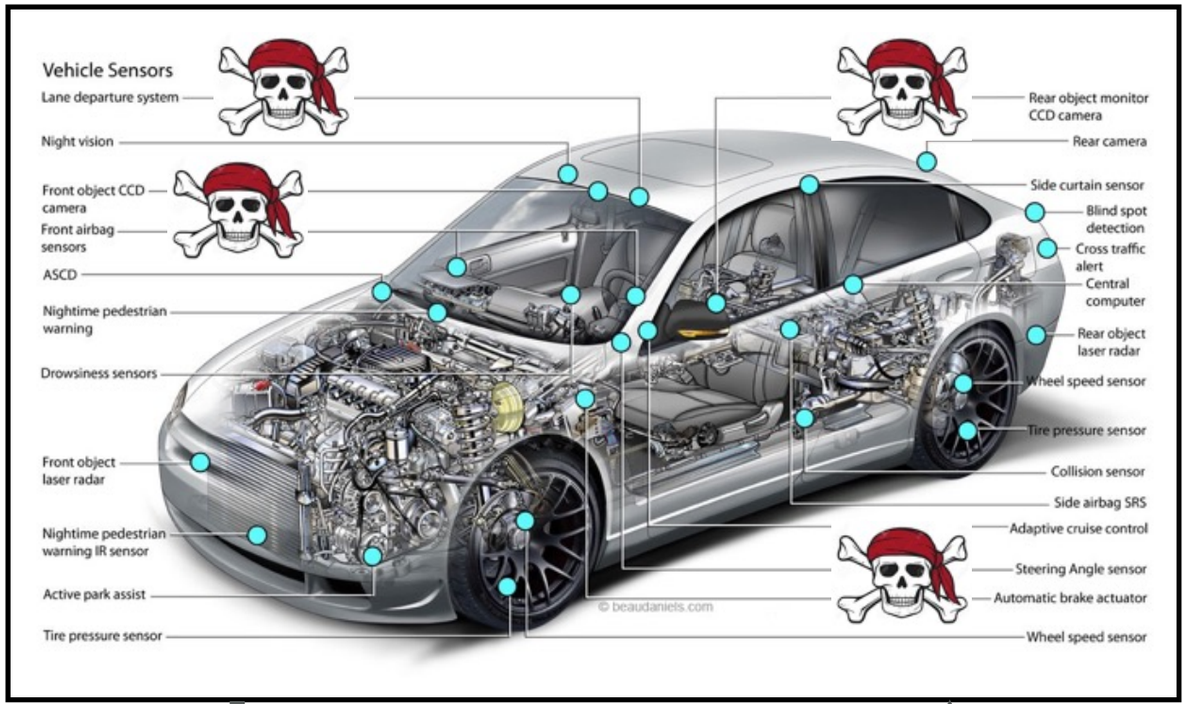
p = total number of sensors
s = number of attacked sensors

- Homogeneous sensing

majority voting (**p > 2 s**)

- Heterogeneous sensing ?

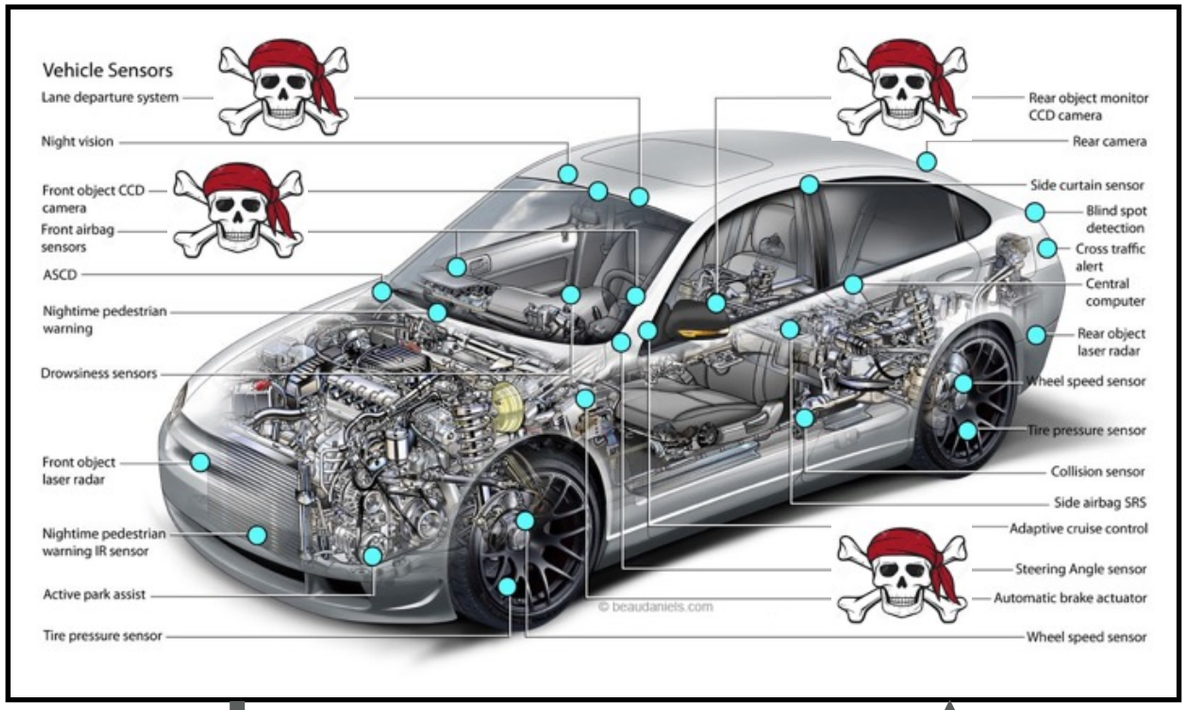
- Dynamics ?

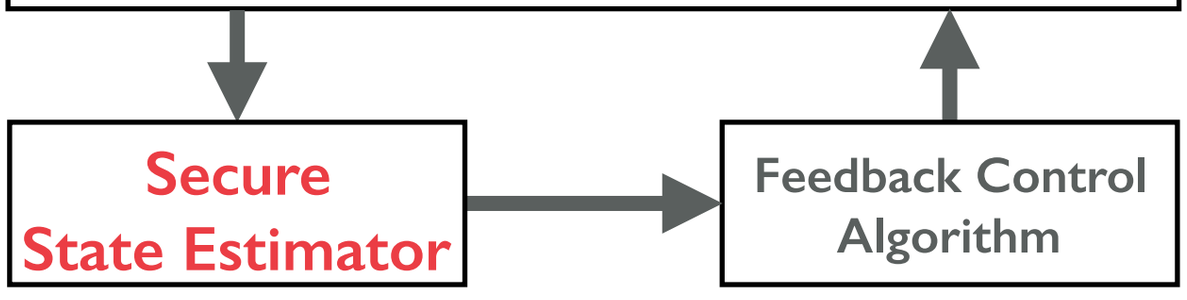


Secure State Estimator → **Feedback Control Algorithm**

Secure State Estimation

Key idea:
exploit physics and
dynamics to increase
redundancy





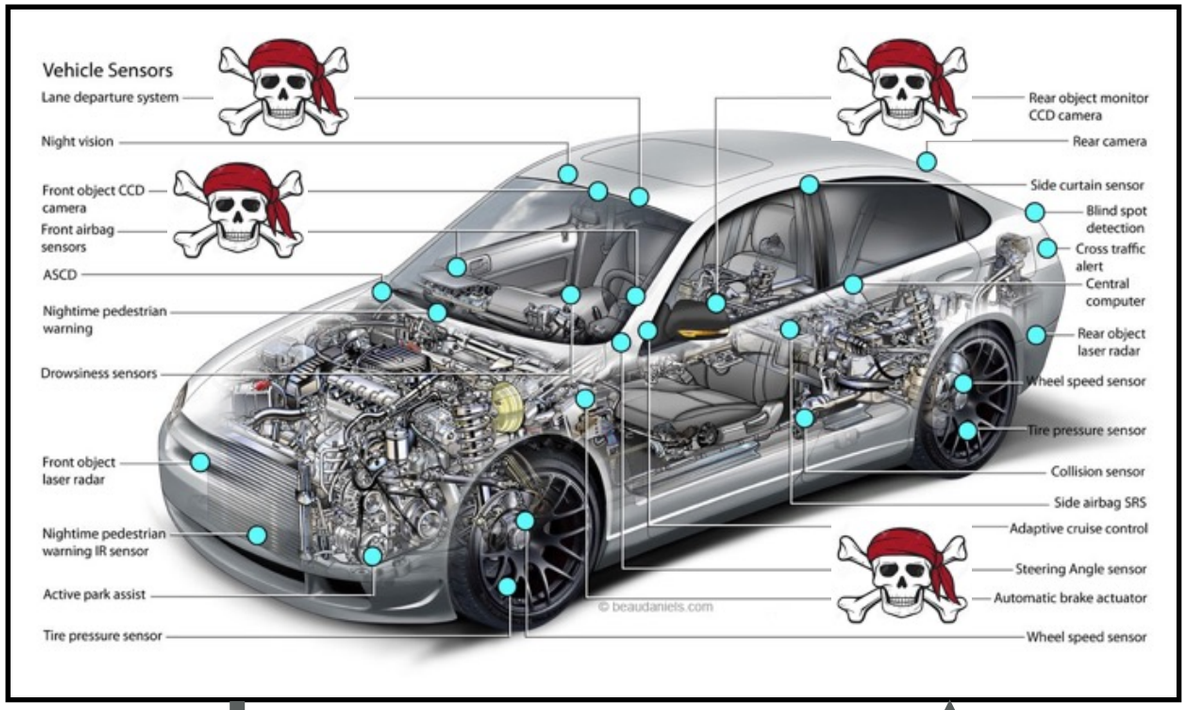
Secure State Estimation

Key idea:
exploit physics and dynamics to increase redundancy

Scalability?

Real-time?

NP-hard?



Secure State Estimation

Observability:

The ability to construct the state from the outputs $Y = \mathcal{O}x$

Definition (s-sparse observable):

A dynamical system is said to be s-sparse observable if it is observable from any p - s sensors.

Theorem:

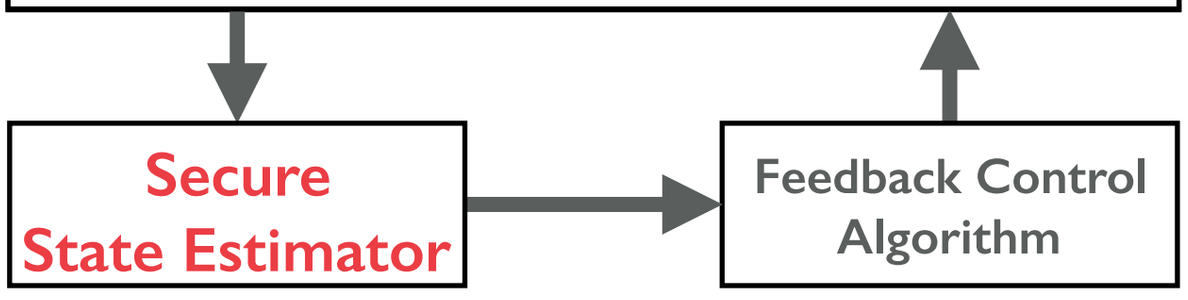
The secure state estimation admits a unique solution **if and only if** the dynamical system is **2s**-sparse observable system.

Recall: homogeneous sensing (**p > 2s**)

extended for nonlinear systems, bounded noise, Gaussian noise

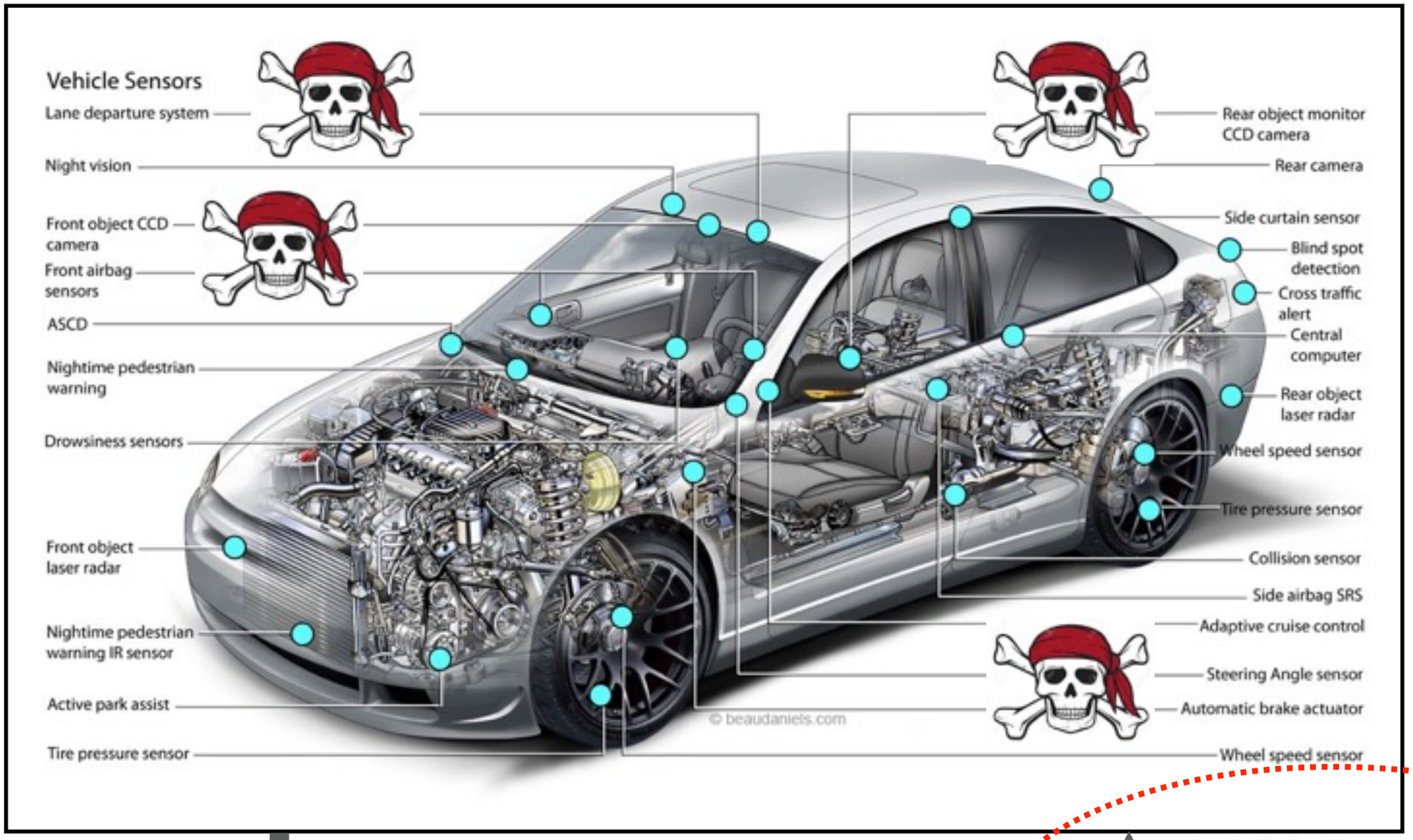
$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$

$$y^{(t)} = Cx^{(t)} + \underbrace{a^{(t)}}_{s-sparse}$$



Y. Shoukry and P. Tabuada, "Event-Triggered State Observers for Sparse Sensor Noise/Attacks," TAC 2016

Secure State Estimation



Secure State Estimator

Feedback Control Algorithm

Actuator Attack

Secure State Estimation

Threat Model:

- The attacker has compromised **"s" (out of "p")** sensors.
- The attacker has compromised **"r" (out of "m")** actuators.

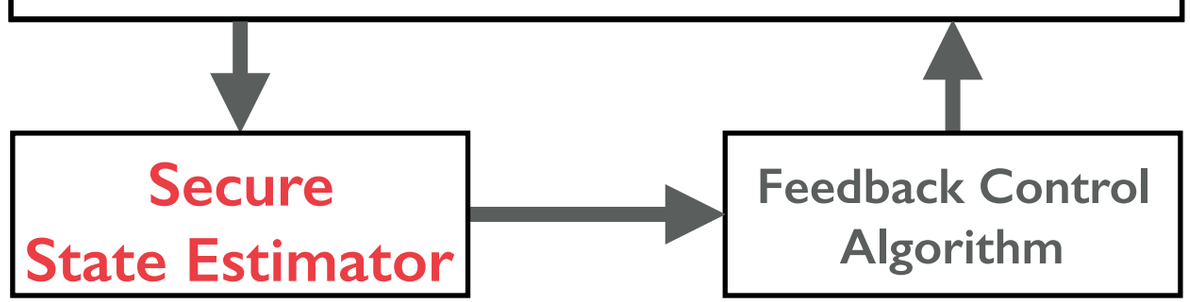
Definition ((r,s)-sparse strongly observable):
A dynamical system is said to be (r, s)-sparse strongly observable if it is strongly observable from any r actuators and p - s sensors.

Theorem:

The secure state estimation admits a unique solution **if and only if** the dynamical system is **(2r,2s)**-sparse **strongly** observable system.

$$x^{(t+1)} = Ax^{(t)} + B\left(u^{(t)} + \underbrace{a_u}_{r-sparse}{}^{(t)}\right)$$

$$y^{(t)} = Cx^{(t)} + \underbrace{a_y}_{s-sparse}{}^{(t)}$$



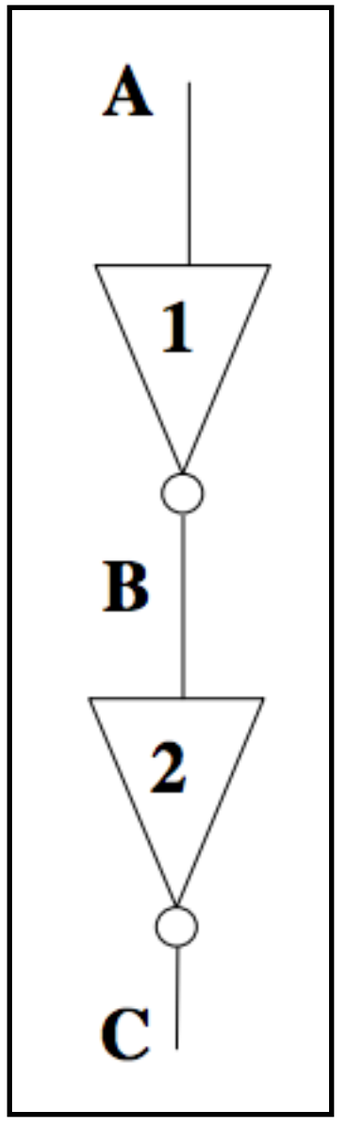
M. Showkatbakhsh, **Y. Shoukry**, R. H. Chen, S. Diggavi, and P. Tabuada, "An SMT-based Approach to Secure State Estimation Under Sensor and Actuator Attacks," CDC 2017.

Algorithms for Secure State Estimation

Algorithms for Secure State Estimation

- **Step 1:** For each gate, define a binary indicator variable (0 = gate is good, 1 = gate is bad)
- **Step 2:** Build a model:

$$\Sigma = \begin{cases} \neg b_1 & \Rightarrow & A \Leftrightarrow \neg B \\ \neg b_2 & \Rightarrow & B \Leftrightarrow \neg C \\ b_1 & \Rightarrow & (A \Leftrightarrow B) \vee \neg B \\ b_2 & \Rightarrow & (B \Leftrightarrow C) \vee \neg C \\ \sum_i b_i \leq 1 \end{cases}$$



- Collect inputs and outputs … append them to model.
- Satisfiability problem … use SAT solver.
- Scales to millions of gates!

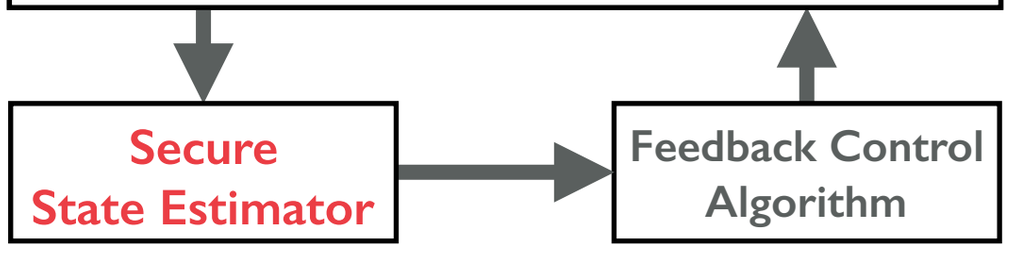
Algorithms for Secure State Estimation

- **Step 1:** For each sensor, define a binary indicator variable (0 = sensor is good, 1 = sensor is attacked)
- **Step 2:** Build a model:

$$Y_i = \mathcal{O}_i x + \underbrace{\Psi_i}_{\substack{\text{model} \\ \text{mismatch}}} \qquad \text{attack free}$$

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

$$y_i^{(t)} = C_i x^{(t)} + \underbrace{w^{(t)}}_{\substack{\text{sensor} \\ \text{noise}}}$$

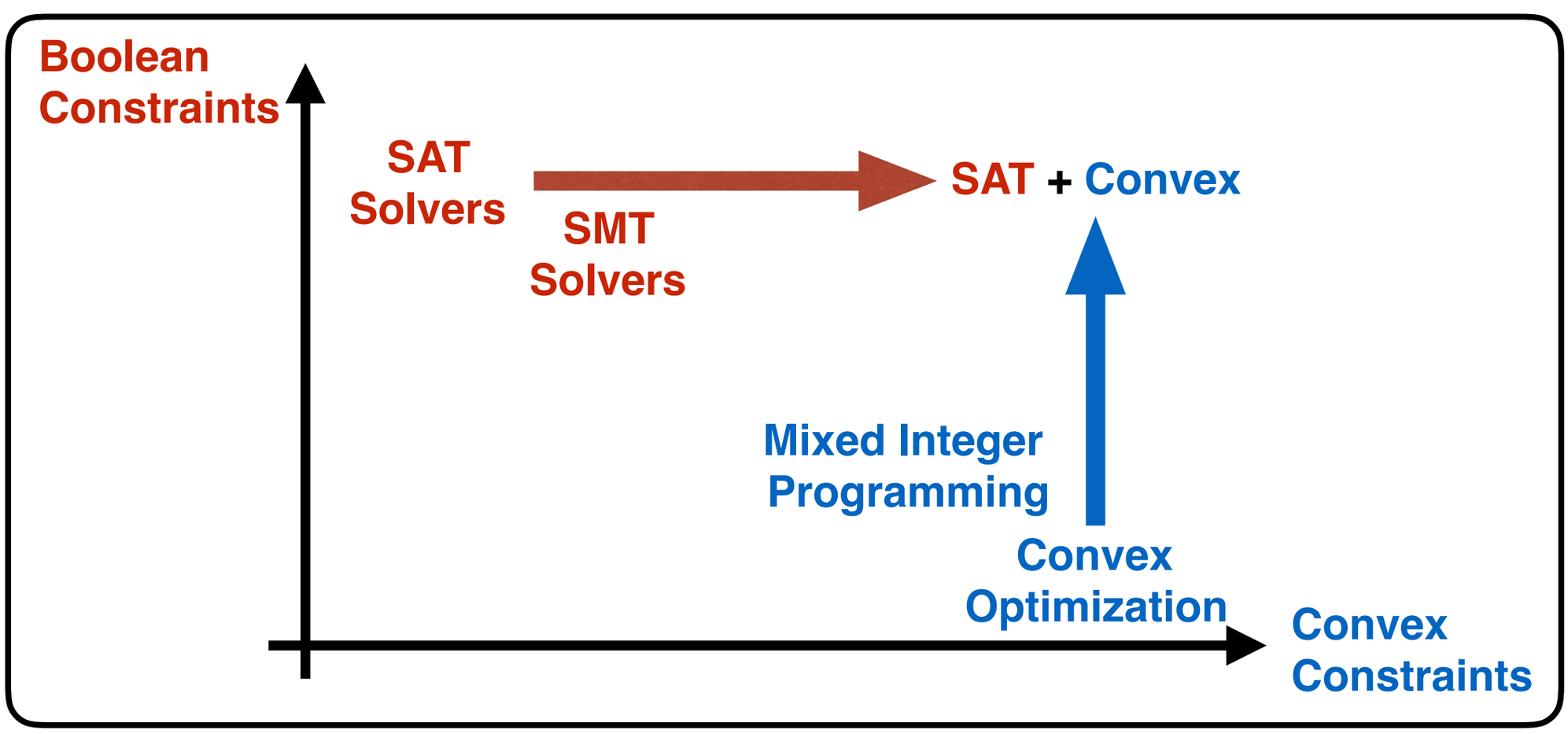


- Satisfiability problem … can not use SAT solvers

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

Satisfiability Modulo Convex Programming

- SAT Solvers: one of the centric tools in computer science to reason about cyber-systems.
- Convex Optimization: one of the centric tools in electrical engineering to reason about physical systems.
- Cyber-Physical Systems?

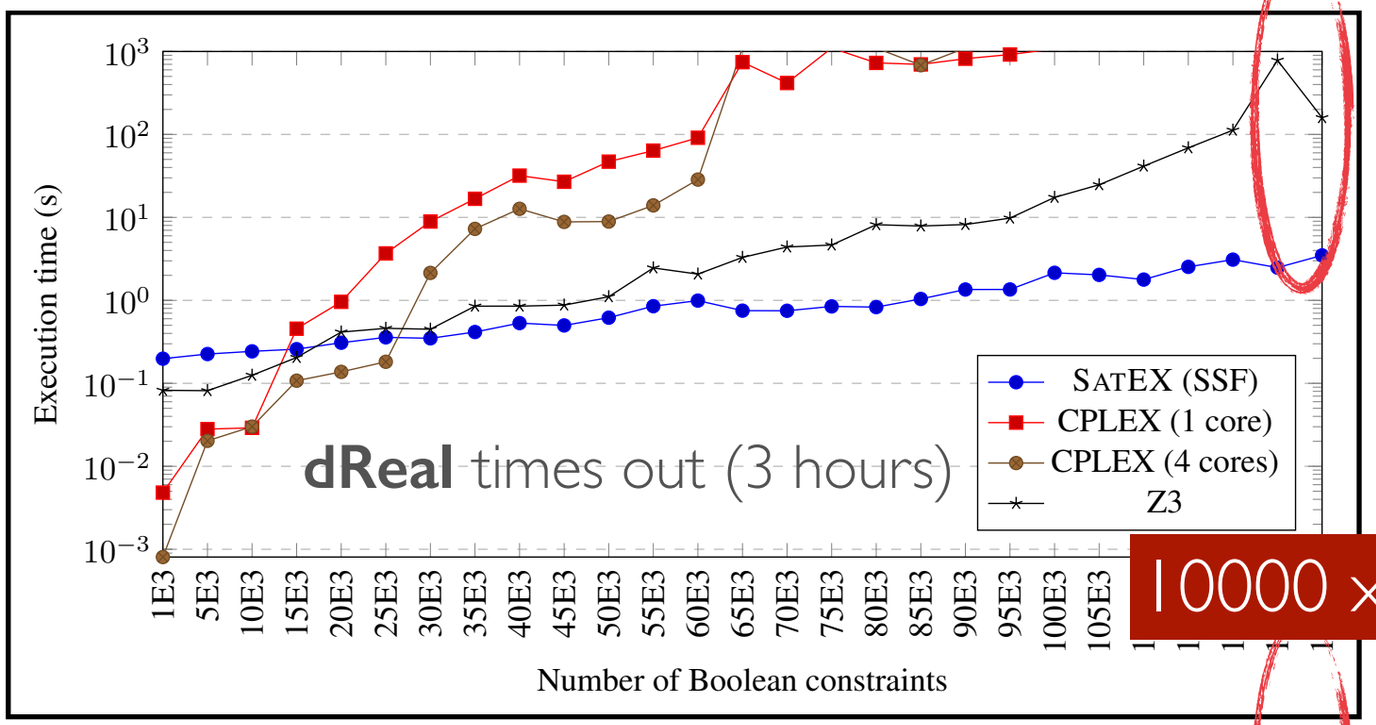


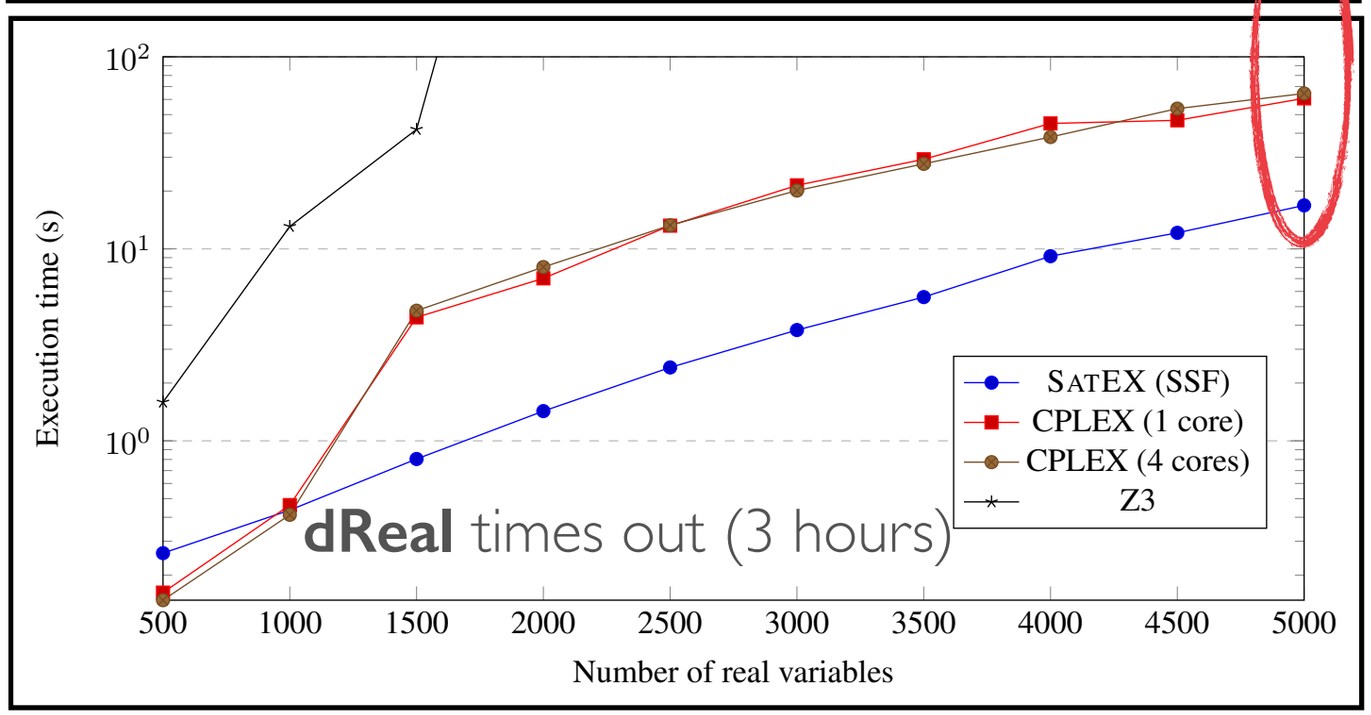
Scalability Results

Increase the number of
Boolean constraints
#Boolean variables = 4800
#Real variables = 100

http://yshoukry.bitbucket.io/SatEX

Increase the number of
Real variables
#Boolean variables = 4800
#Boolean constraints = 7000



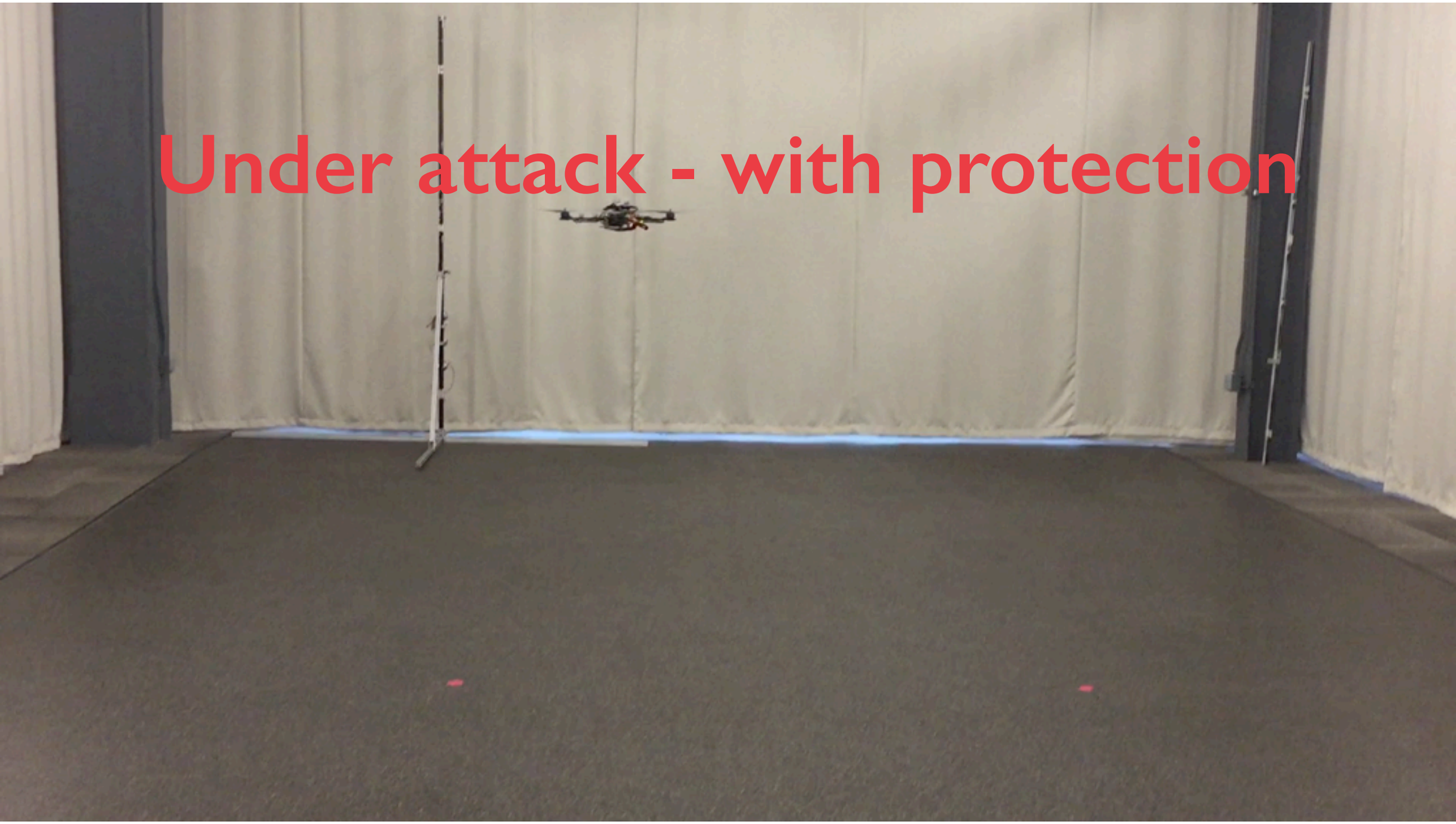


Scalability Results

Approach	Scalability? (500 sensors with 100 being under attack)	Performance? (State estimation error)
Brute force search [F. Pasqualetti et al. TAC 2013] [M. S. Chong et al. ACC 2015] [S. Mishra et al. ISIT 2015]	**Time out (> 7 hours)**	**Optimal**
Mixed-integer programming [M. Pajic et al. ICCPS 2014]	**1.5 hours**	**sub-optimal**
SMT solvers (Z3/dReal) [M. Rahman et al. ICCPS 2014]	**Time out (> 3 hours)**	**??**
Our approach [Y. Shoukry et al. ICCPS 2016, TAC 2017] Best paper award	**< 15 seconds (no heuristics, no relaxation)**	**Optimal (in the worst case), sub-optimal (in general)**

Experimental Results

Experimental Results



Outline

False Data Injection Attacks

TAC 2016, CDC 2017, ICCPS 2016
(Best paper award)

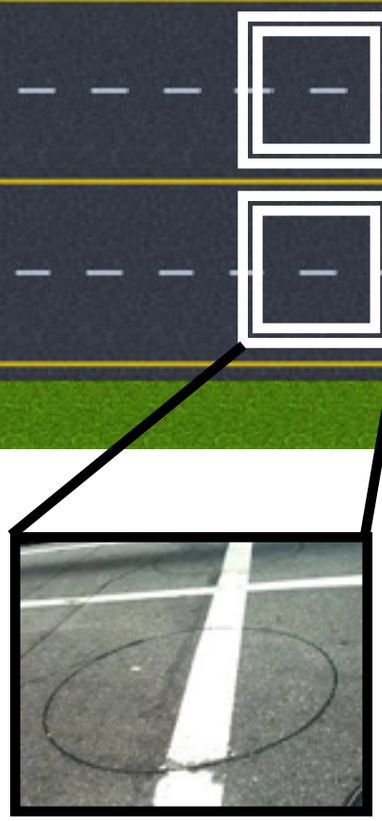
Sybil Attacks + False Data Injection

ICCPS 2018

Privacy-preserving Sensor Fusion + False Data Injection

CDC 2016, IPSN 2017
(Best demo award)

Secure Traffic Routing

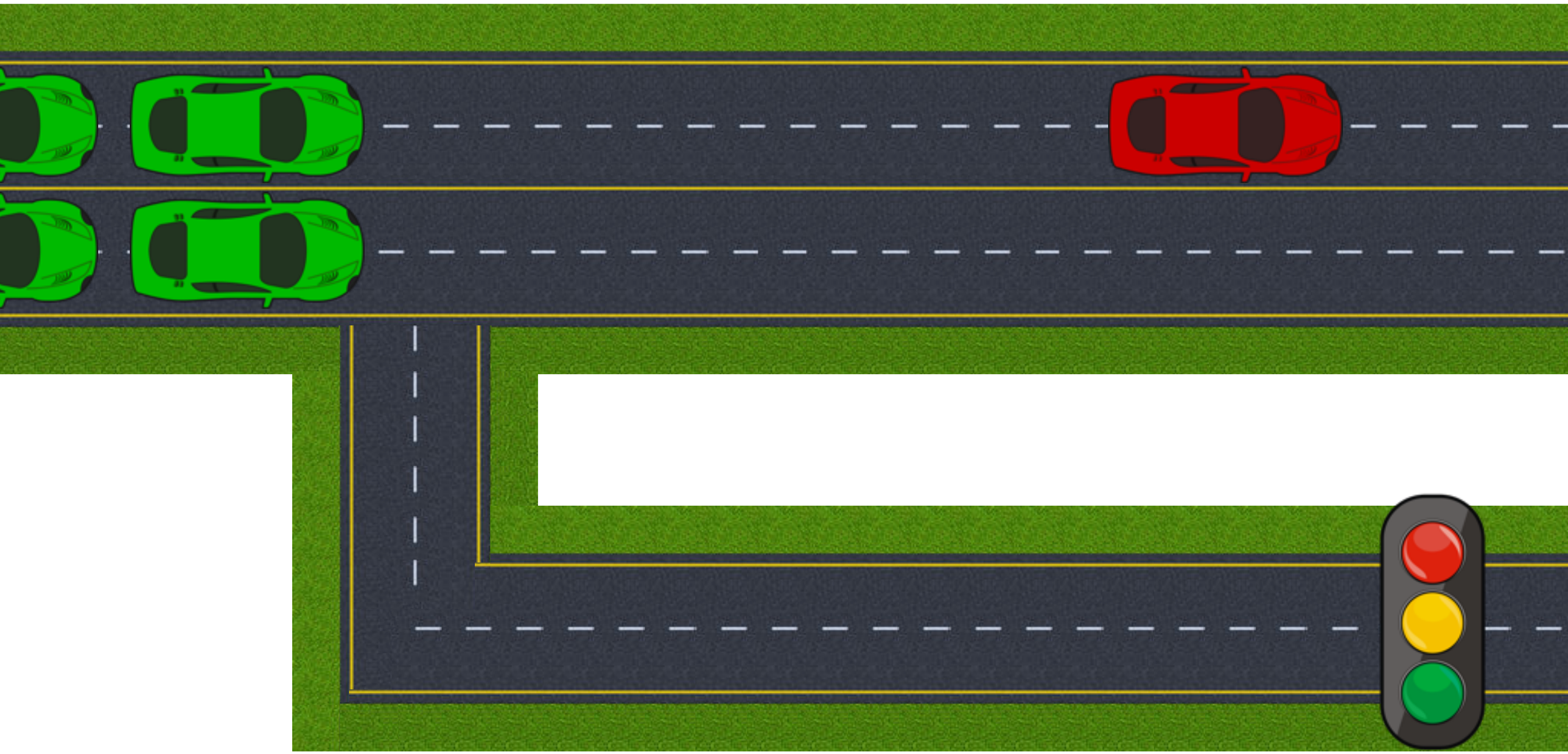


Loop Sensors

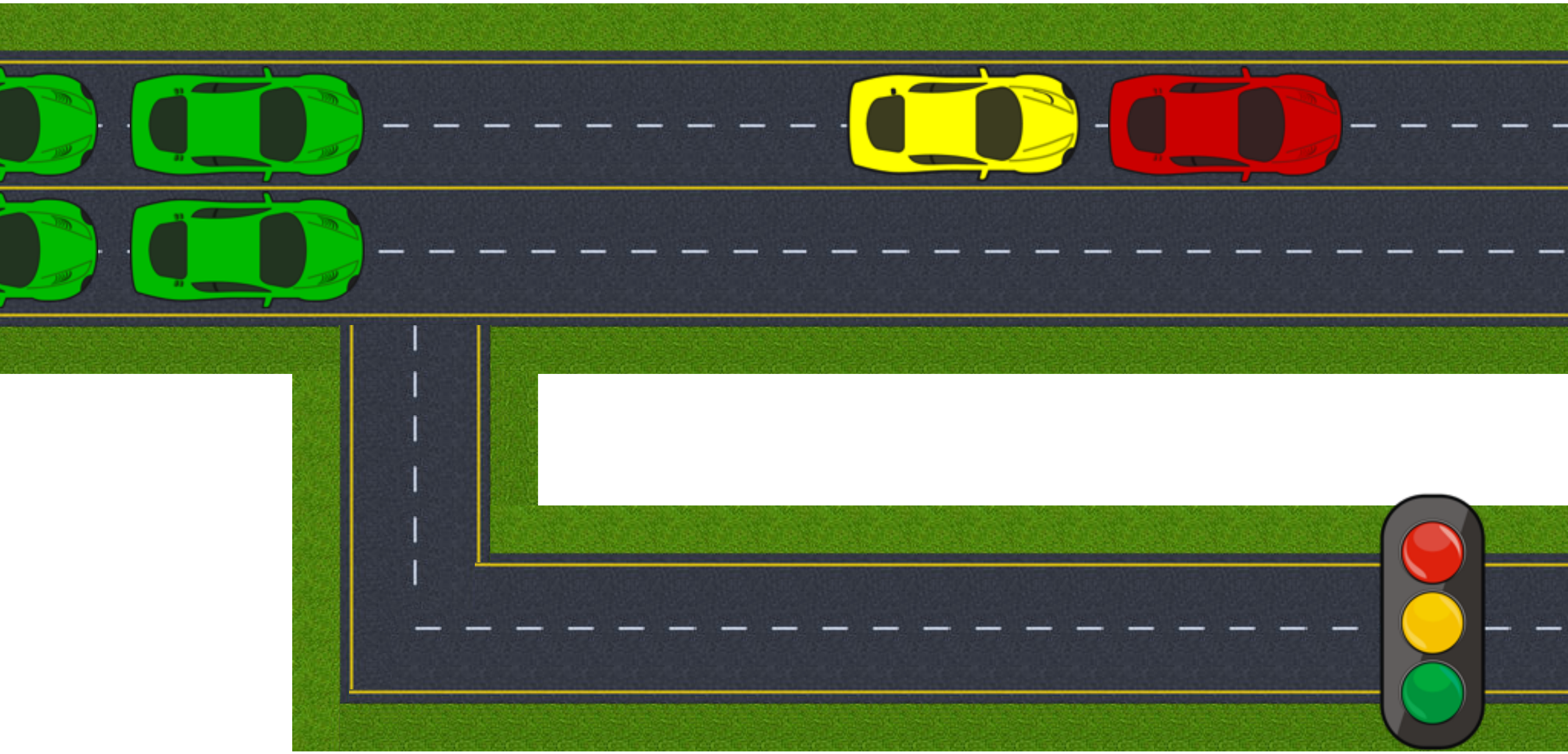
Secure Traffic Estimation

Routing Algorithm

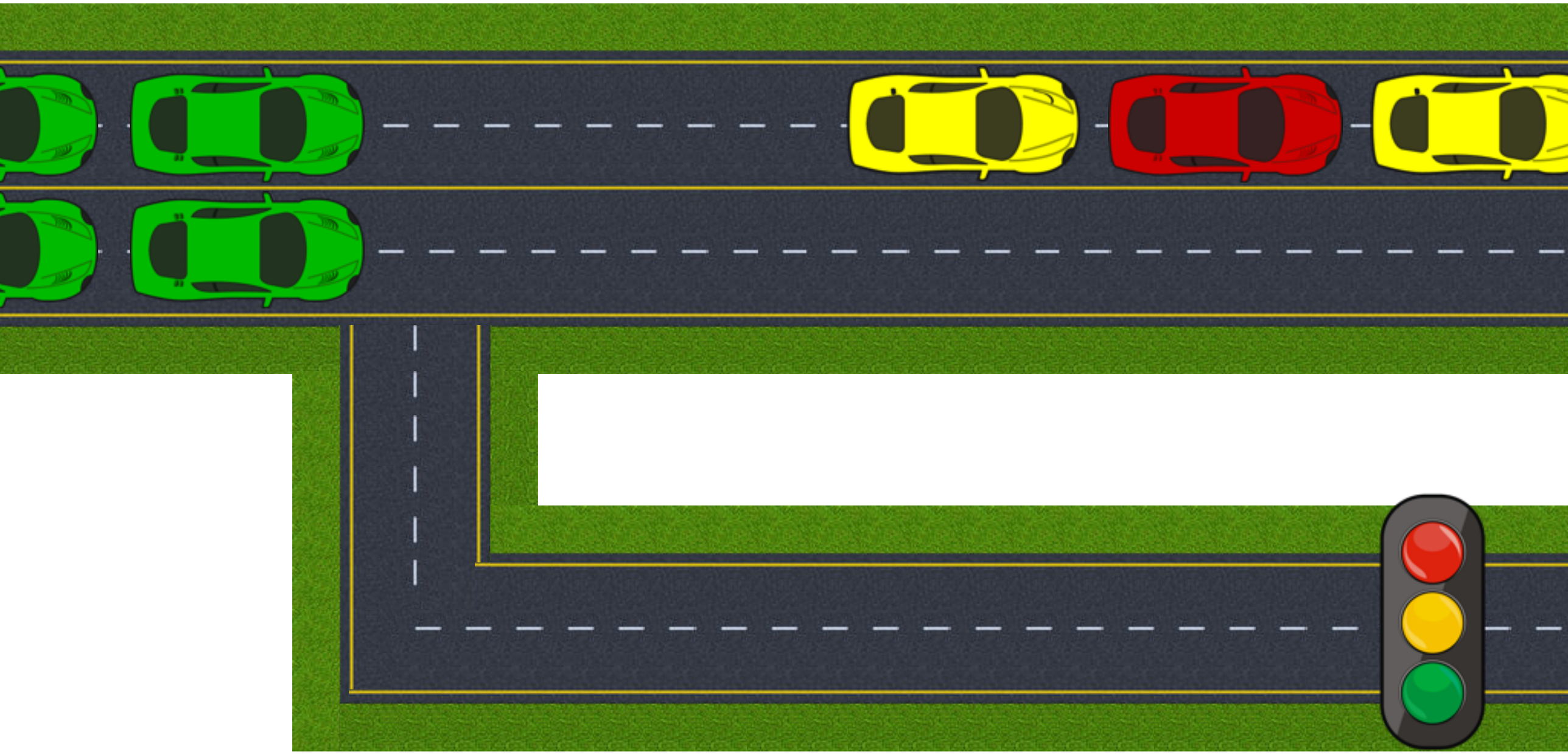
Secure Traffic Routing



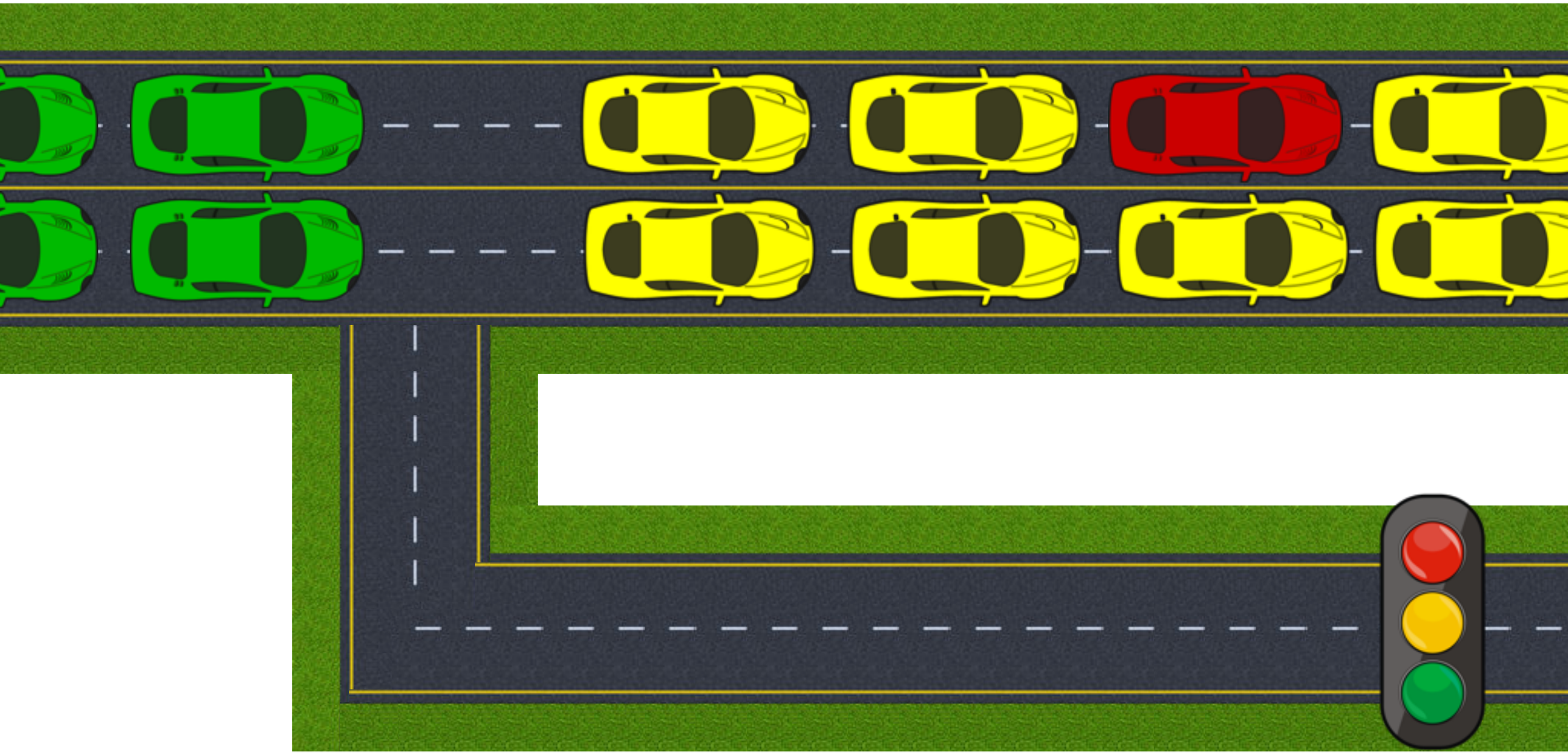
Secure Traffic Routing



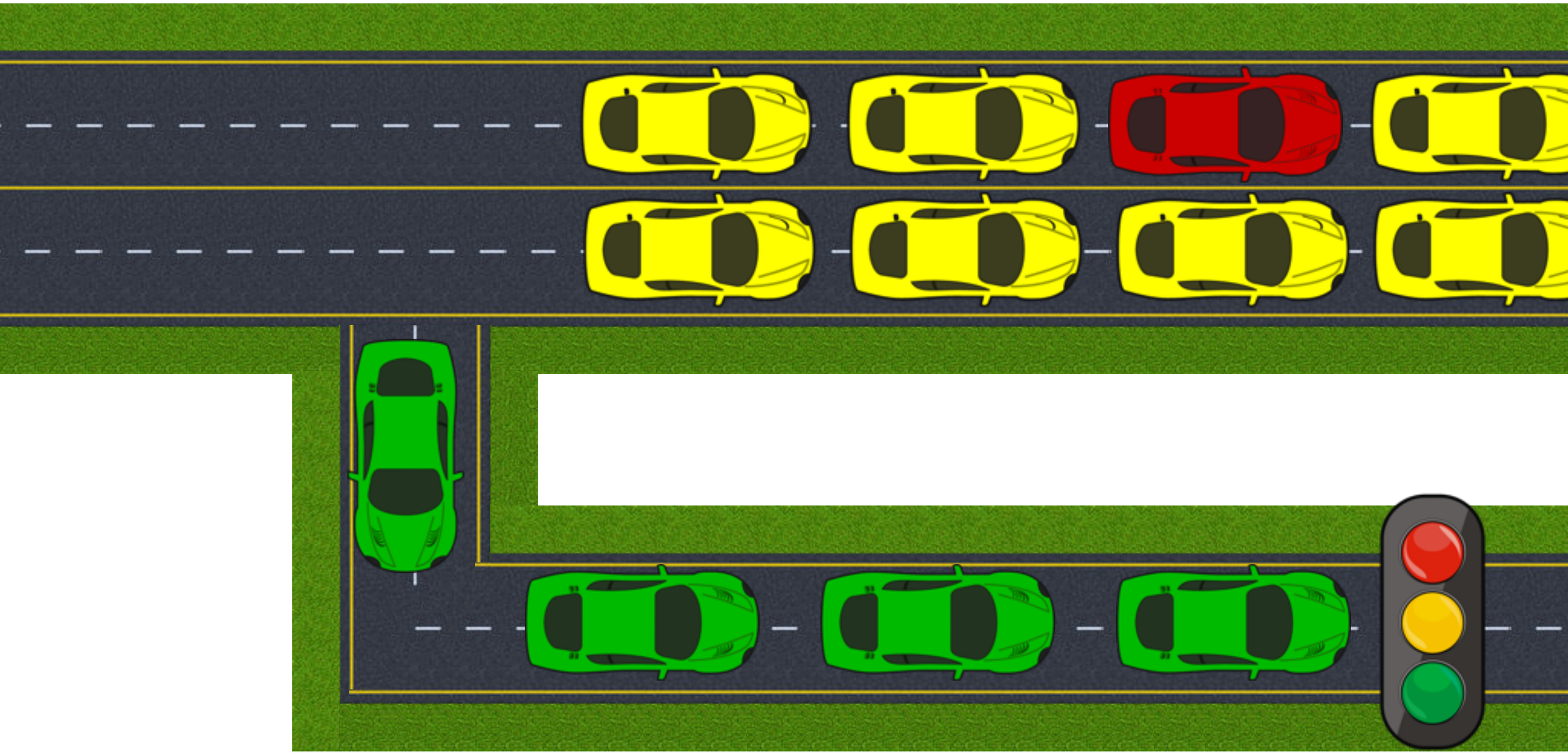
Secure Traffic Routing



Secure Traffic Routing

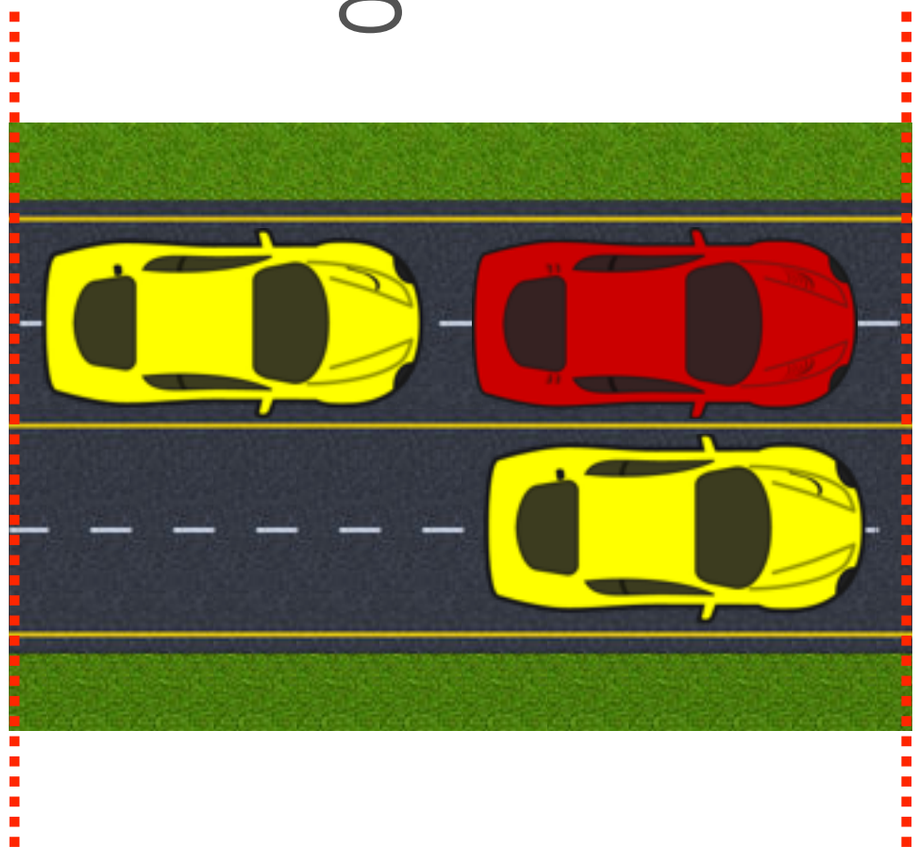


Secure Traffic Routing



Secure Traffic Routing

Threat Model:

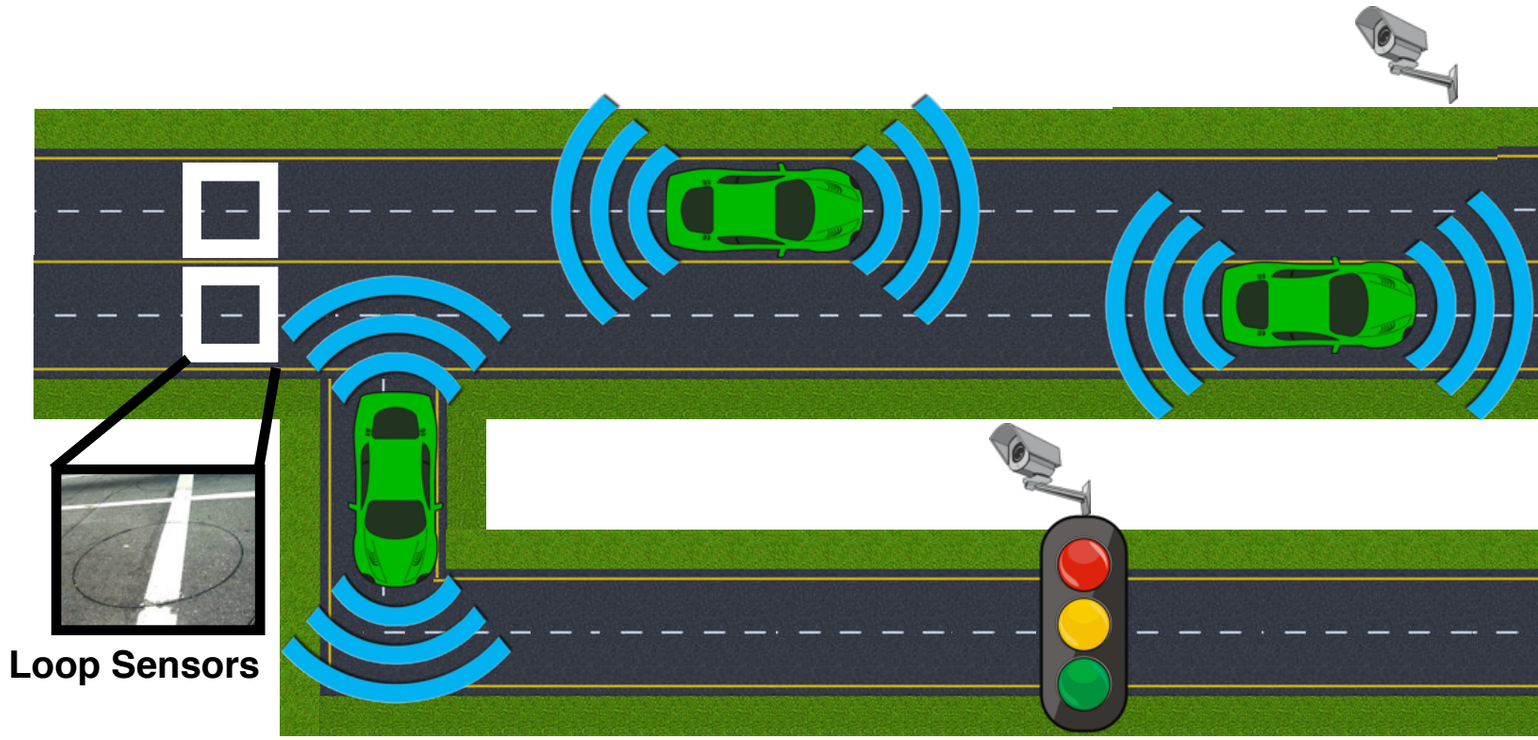


- **False Data Injection:** a car that physically exists on the road is reporting *maliciously corrupted information* (wrong position, speed, and/or speed of nearby vehicles).

- **Sybil attacks:** a car which physically exists on the road reports the presence of nearby cars that do *not physically exist (ghost cars)*. Ghost (Sybil) cars may also report the presence of more nearby ghost cars.

Root-of-Trust

- Sensor redundancy is compromised!
- We need another root-of-trust.



- Legacy sensors (e.g., loop sensors and cameras) provide noisy and sporadic measurements.
- Can we use legacy sensors (placed thousands of miles away from the attack position) to detect attacks?

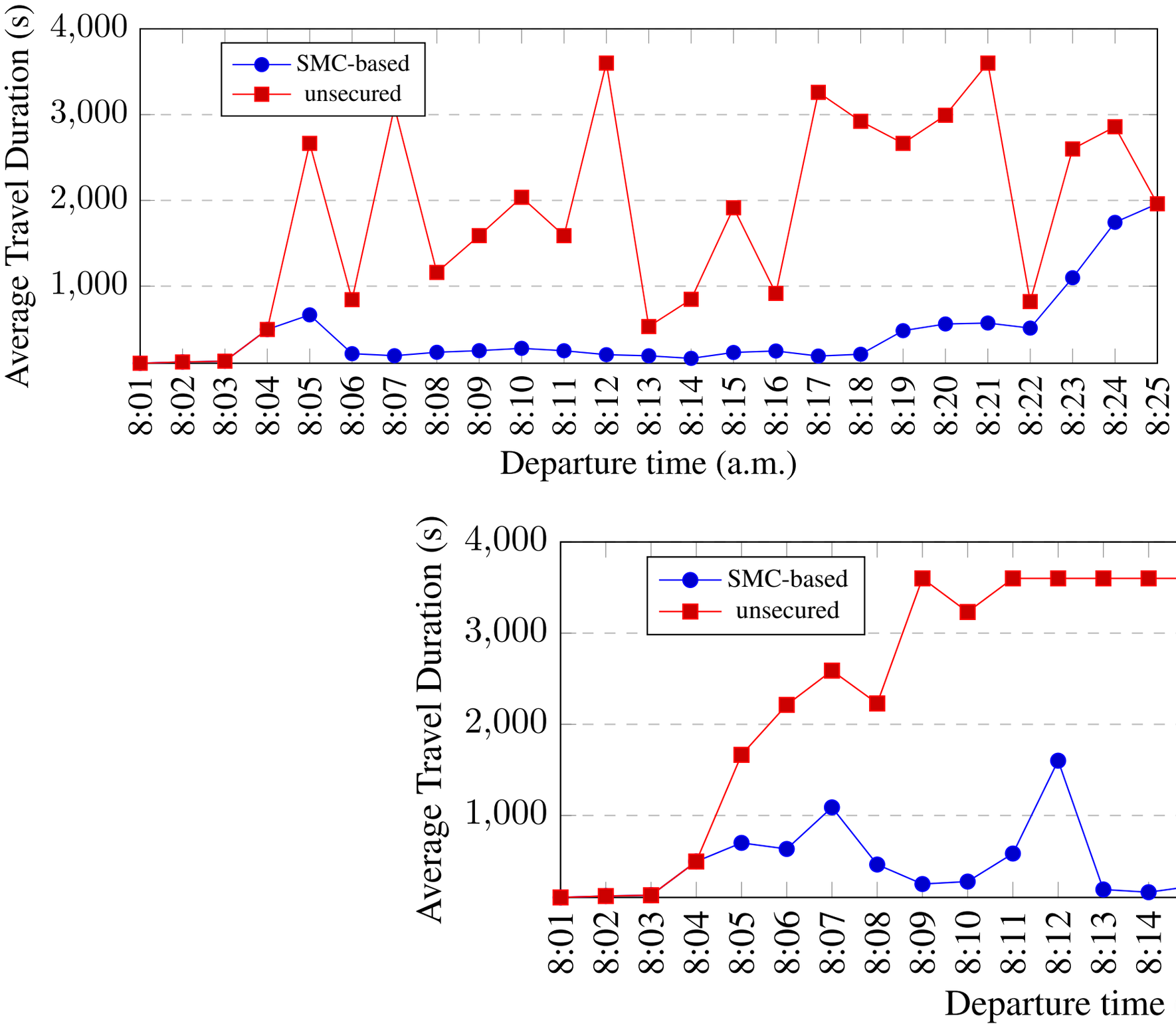
Key idea:
exploit physics and dynamics to propagate the trust.

Case Study: Bologna City

- Bologna Ringway dataset
- Typical day's traffic between 8:00 am and 9:00 am (rush hour)
- More than 22000 vehicles
- To simulate the dynamics because of injected attacks, we use Simulation of Urban MObility (SUMO) simulator.

Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, "Sybil Attack Resilient Traffic Networks: A Physics-Based Trust Propagation Approach," ICCPS 2018

Numerical Results



Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, “Sybil Attack Resilient Traffic Networks: A Physics-Based Trust Propagation Approach,” ICCPS 2018

Outline

False Data Injection Attacks

TAC 2016, CDC 2017, ICCPS 2016
(Best paper award)

Sybil Attacks + False Data Injection

ICCPS 2018

Privacy-preserving Sensor Fusion + False Data Injection

CDC 2016, IPSN 2017
(Best demo award)

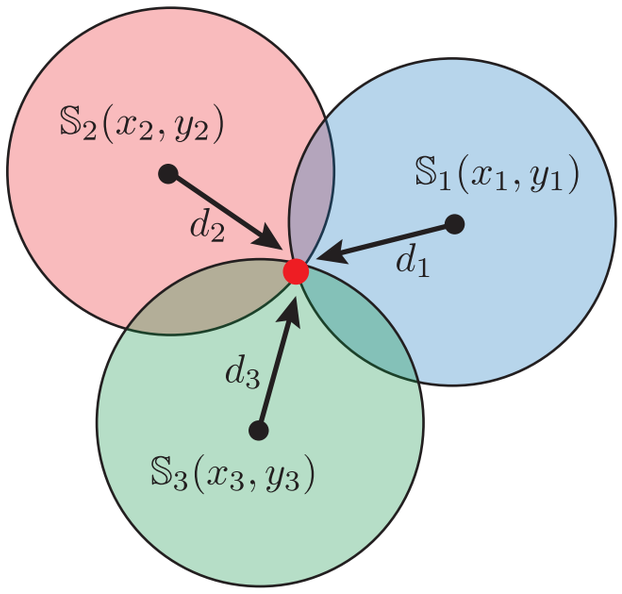
Privacy-Aware Sensor Fusion

- Sensor data collected from various sources are aggregated at a centralized node.

- Two leak scenarios:
  - Malicious aggregators
  - Information leaks from trusted aggregators

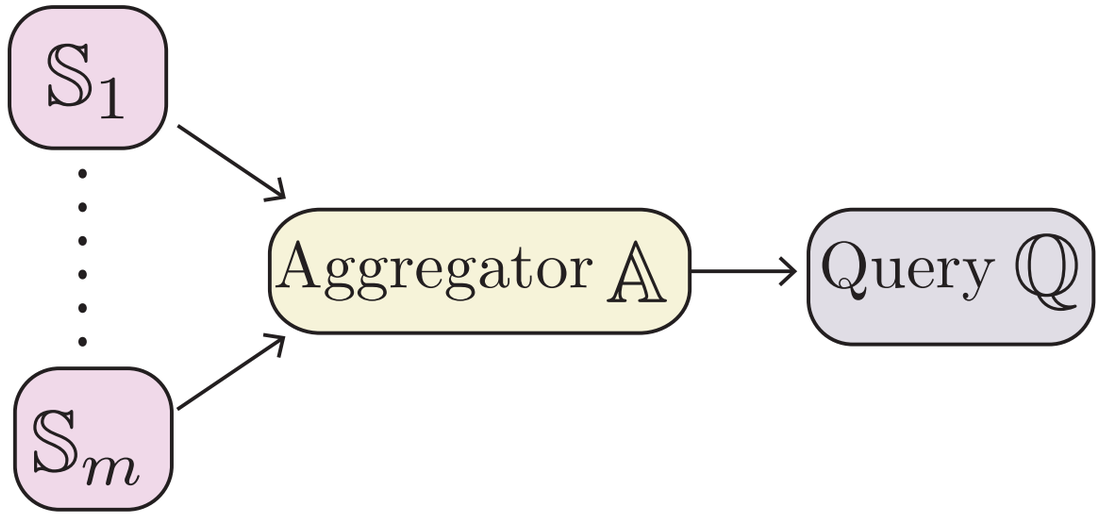
- False data injection attacks:
  some of these sensors are malicious

Problem Setup



$\mathbb{S}_1 \ : \ (x_1, y_1), d_1$
$\mathbb{S}_2 \ : \ (x_2, y_2), d_2$
$\mathbb{S}_3 \ : \ (x_3, y_3), d_3$

● target location

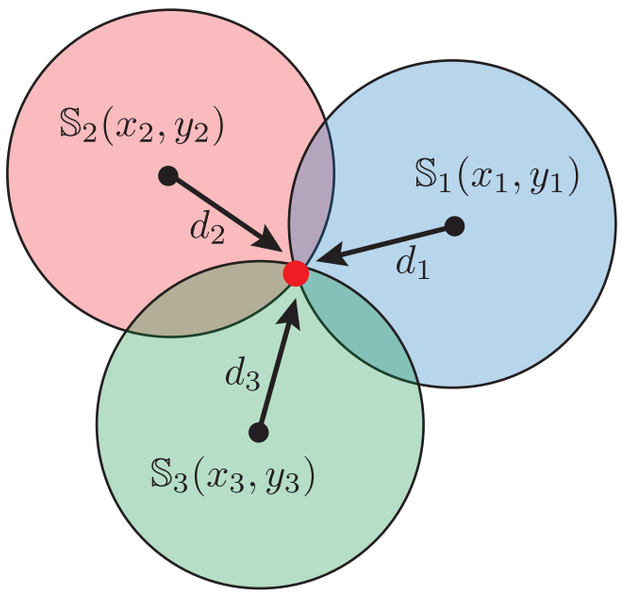


- Sensors (or anchors) provide their own location and an estimated range to the target of interest.

- **Objective:** calculate the target location (x_T , y_T) while ensuring the privacy of all observer locations (x_i , y_i) as well as the distance to the target, x_i.

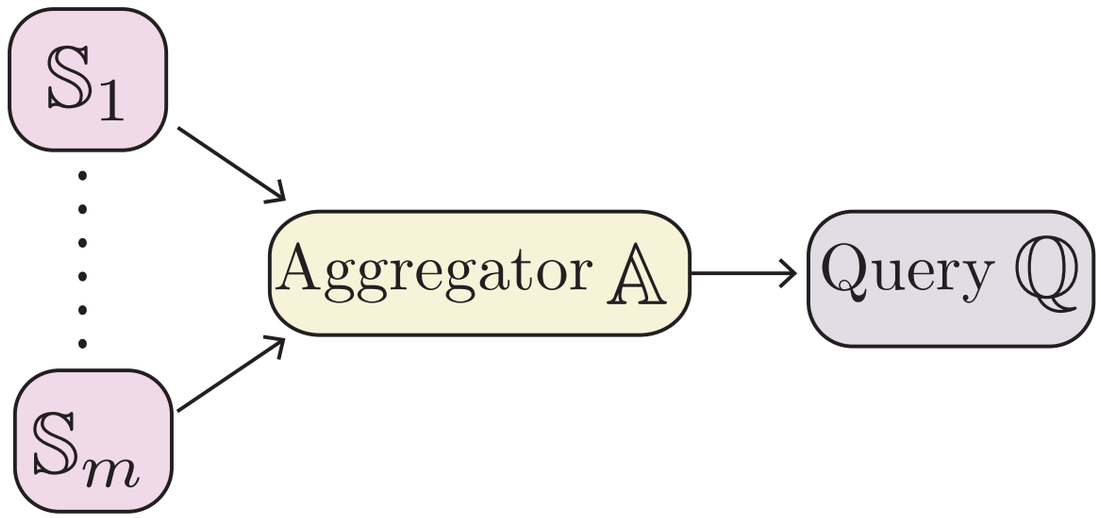
- Semi-honest adversary (honest but curious)

Threat Model



$\mathbb{S}_1 \ : \ (x_1, y_1), d_1$
$\mathbb{S}_2 \ : \ (x_2, y_2), d_2$
$\mathbb{S}_3 \ : \ (x_3, y_3), d_3$

● target location

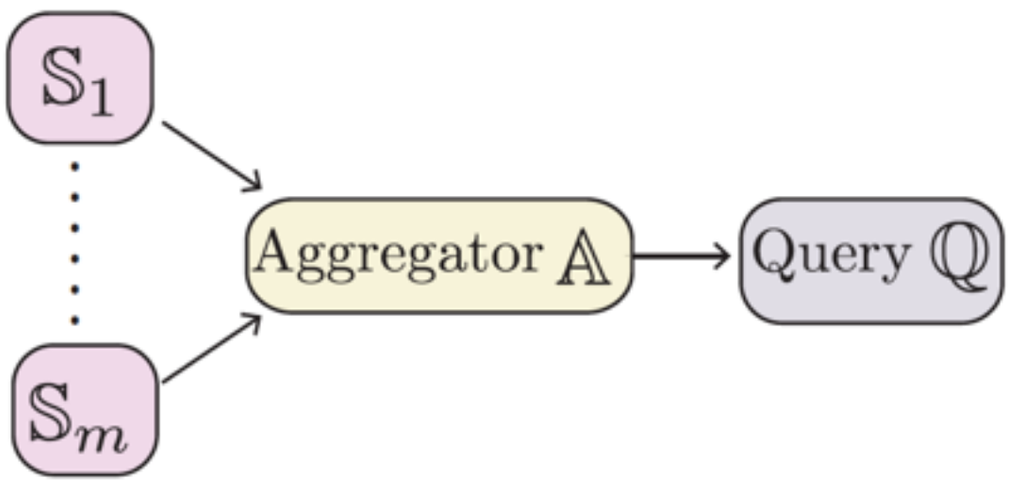


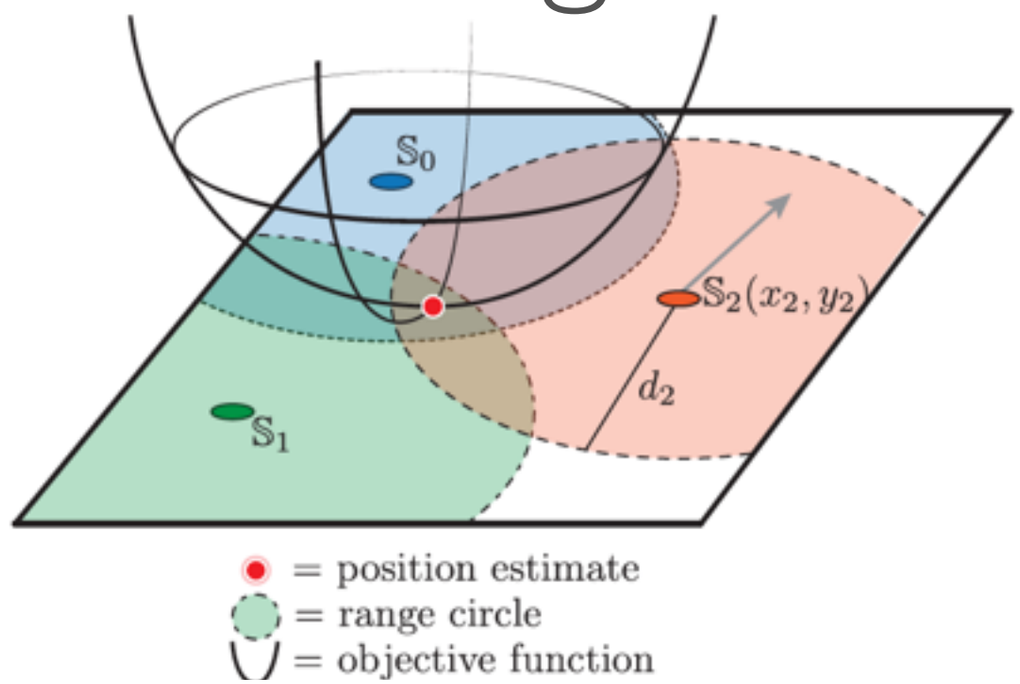
- **Privacy against sensor coalitions:**
  If **any sensor** colludes with up to (m-1) other sensors, the coalition should learns nothing about the non-colluding agents' private information (other than the information contained in the immutable privacy leak).

- **Privacy against aggregator coalitions:**
  If the **aggregator** sensor colludes with up to (m-1) other sensors, the coalition should learns nothing about the non-colluding agents' private information (other than the information contained in the immutable privacy leak).

- **Resilience against data injection attacks**

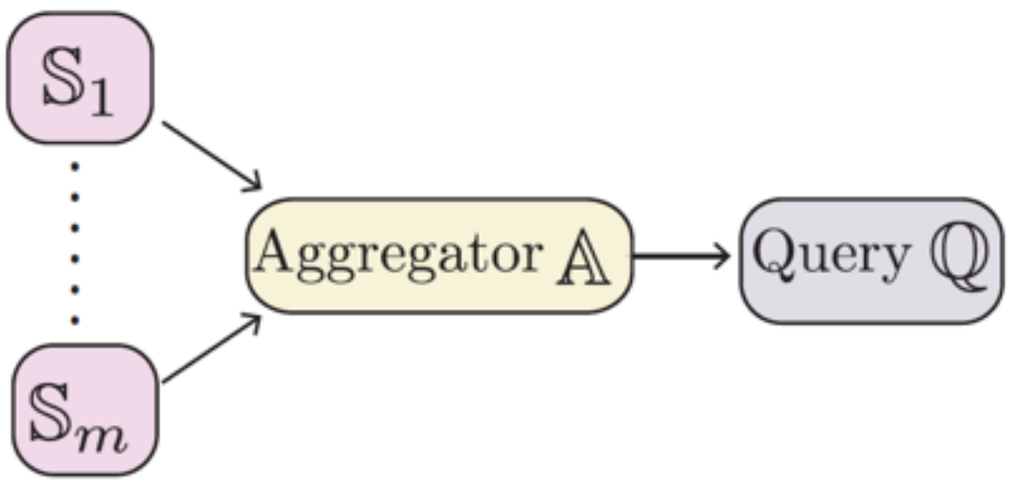
Encrypted Trilateration Algorithm

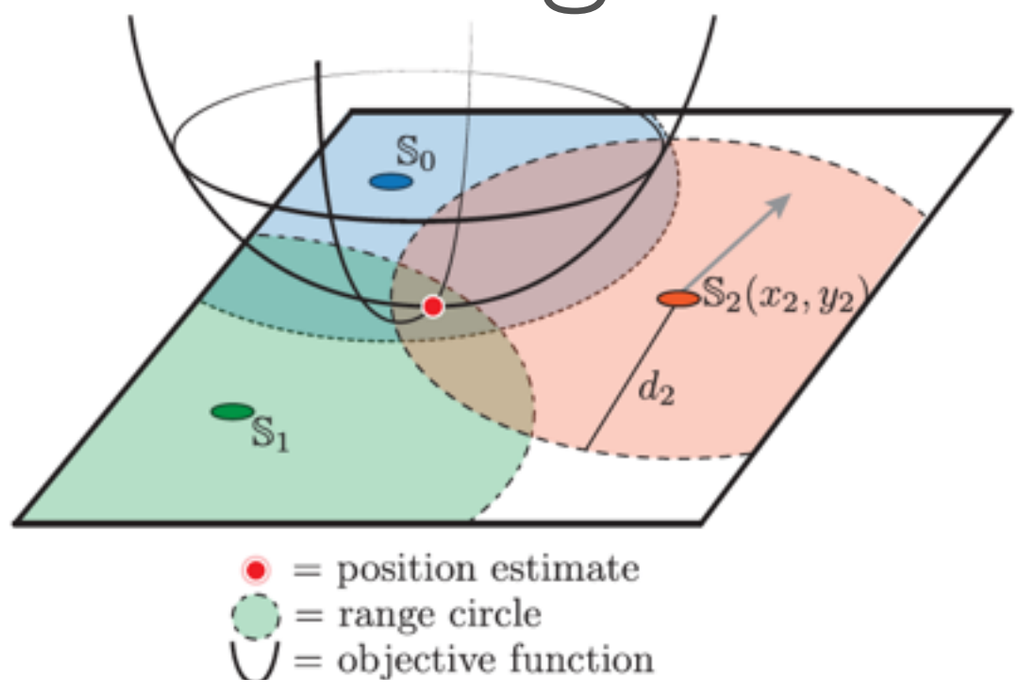




- **Step1-Encrypt:** each observer performs the following steps:
  - Encrypts its distance measurement using the public key of the query node $\text{pk}_{\mathbb{Q}}$.
  - Encrypts the message once more using the public key of the aggregator $\text{pk}_{\mathbb{A}}$.
- **Step2-Aggregate:** The aggregator decrypts all the messages msg_i using his private key $\text{sk}_{\mathbb{A}}$ and constructs the following matrices using the extracted data:
- **Step3-Decrypt:** The aggregator $\mathbb{A}$ sends the final estimate $[\![\hat{z}_{\mathbb{T}}^{(\mathbb{I})}]\!]_{\text{pk}_{\mathbb{Q}}}^{\text{FHE}}$ to the query node $\mathbb{Q}$. The query node decrypts the message using its private key $\text{sk}_{\mathbb{Q}}$ to retrieve the final estimate $\hat{z}_{\mathbb{T}}^{(\mathbb{I})}$.

Encrypted Trilateration Algorithm



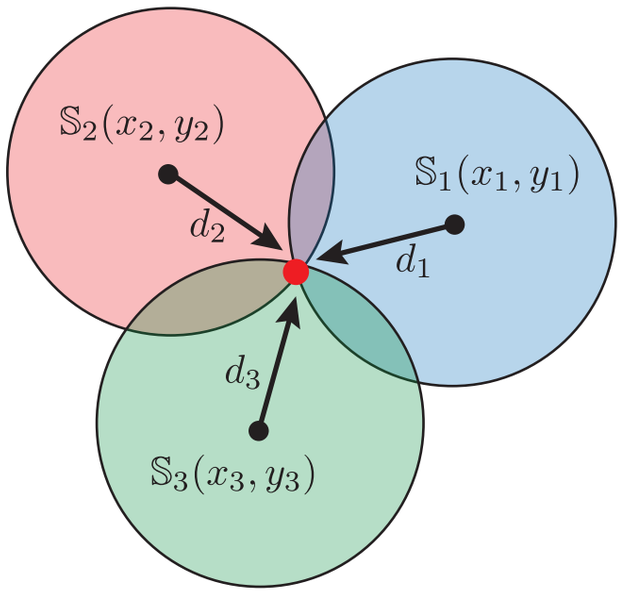


- **Step1-Encrypt:** each observer performs the following steps:
  - Encrypts its distance measurement using the public key of the query node $\mathrm{pk}_{\mathbb{Q}}$.
  - Encrypts the message once more using the public key of the

Execution time = 11.7 hours!

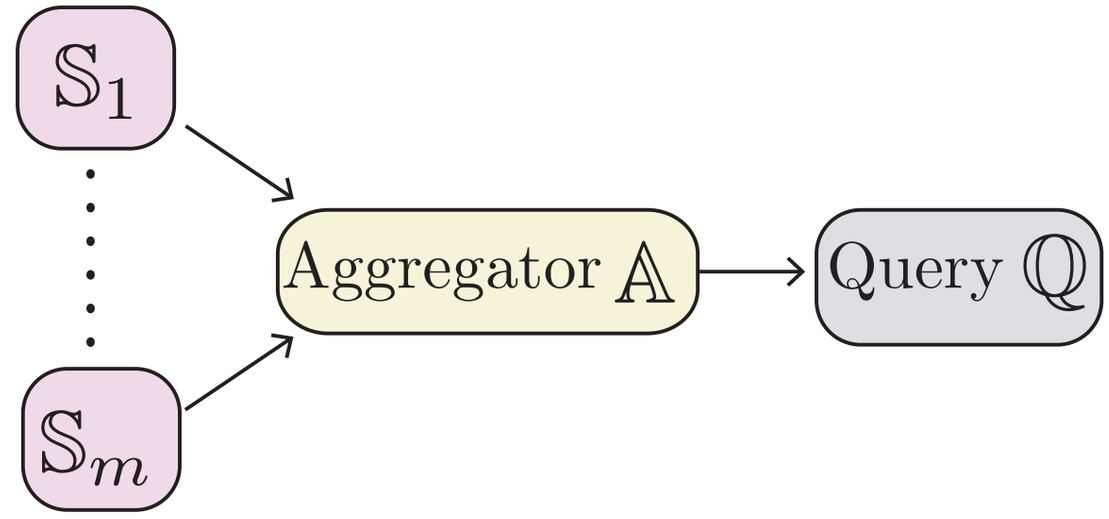
- **Step3-Decrypt:** The aggregator $\mathbb{A}$ sends the final estimate $[\![\hat{z}_{\mathbb{T}}^{(\mathbb{I})}]\!]_{\mathrm{pk}_{\mathbb{Q}}}^{\mathrm{FHE}}$ to the query node $\mathbb{Q}$. The query node decrypts the message using its private key $\mathrm{sk}_{\mathbb{Q}}$ to retrieve the final estimate $\hat{z}_{\mathbb{T}}^{(\mathbb{I})}$.

How To Reason About Privacy



$\mathbb{S}_1 : (x_1, y_1), d_1$
$\mathbb{S}_2 : (x_2, y_2), d_2$
$\mathbb{S}_3 : (x_3, y_3), d_3$

● target location



Theorem:
Assume that all entities are honest-but-curious and under standard cryptographic assumptions (namely the Decisional Composite Residuosity (DCRA) assumption), the proposed localization protocol ensures:

(i) **Sensor obliviousness** (any m-1 sensors can not reveal the information of the remaining sensor)

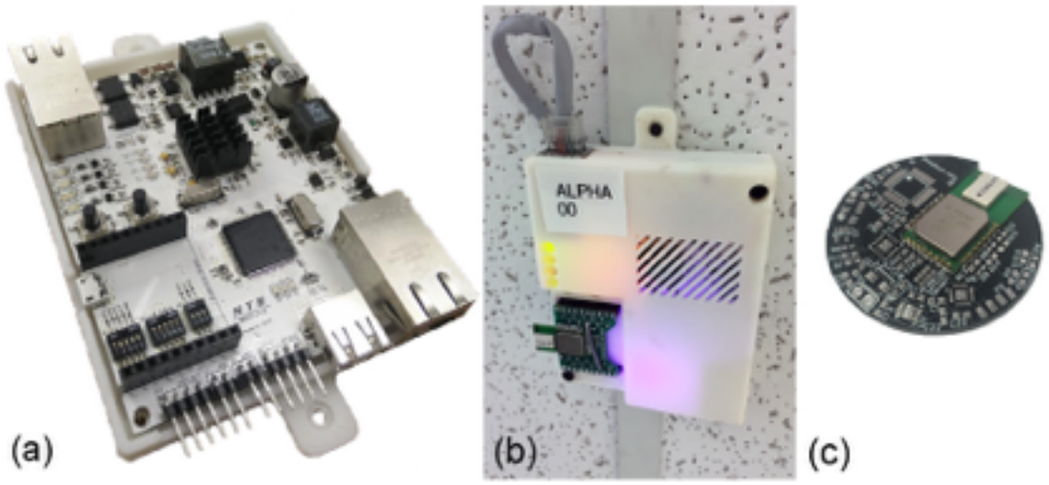
(ii) **Aggregator obliviousness** (aggregator + any m-1 sensors can not reveal the information of the remaining sensor)

(iii) **Resilience** against any $m - \lfloor \frac{m-3}{2} \rfloor$ sensor attacks

A. Alnwar, **Y. Shoukry**, S. Chakraborty, P. Martin, M. Srivastava, and P. Tabauda, "ProLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption," IPSN 2017 (best demo award)

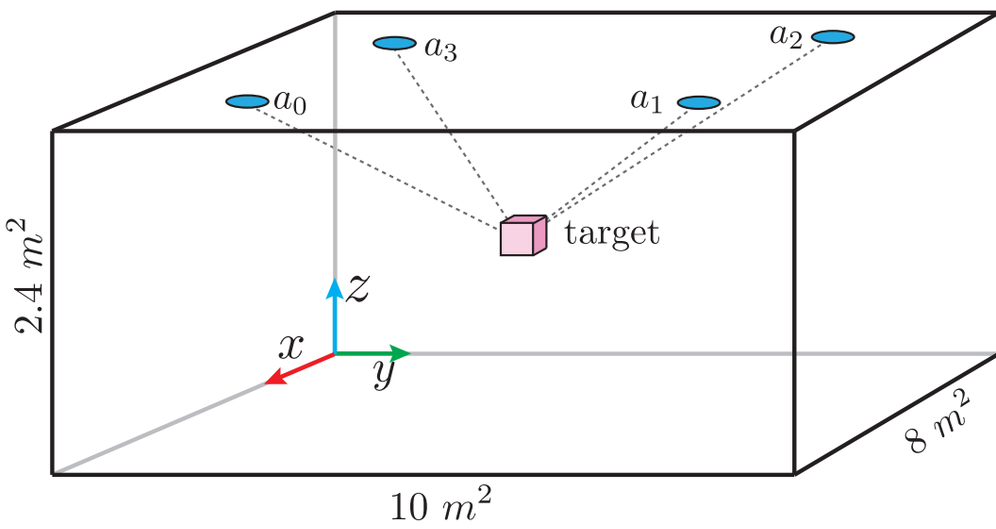
Experimental Results

- Experiments with a mobile query node and sensor nodes mounted on the ceiling



- Cloud and target nodes implemented on a MacBook Pro.

- Custom ranging hardware for time of flight connected to a Nexus 5

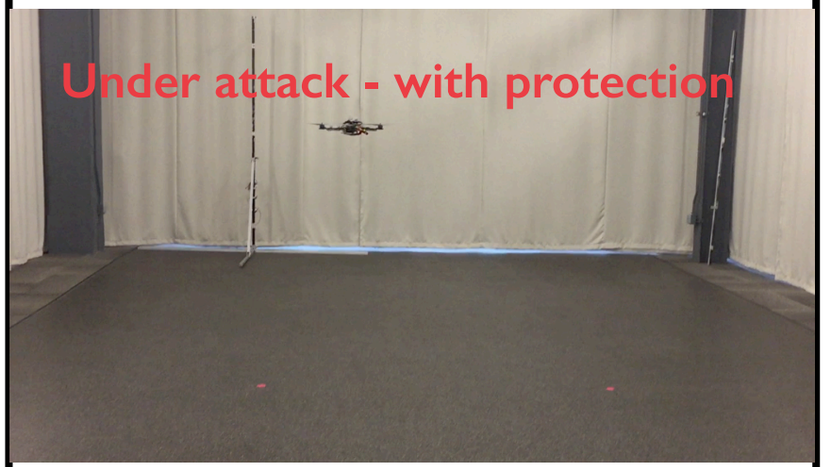


	Mean error (m)	Standard deviation	Computation time (ms)	Communication cost # messages, # Kbytes
Least squares	0.2341	0.18738	0.74	7, 1.2
Proposed algorithm	0.2381	0.18634	103	7, 8.5

A. Alnwar, **Y. Shoukry**, S. Chakraborty, P. Martin, M. Srivastava, and P. Tabauda, "ProLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption," IPSN 2017 (best demo award)

Summary

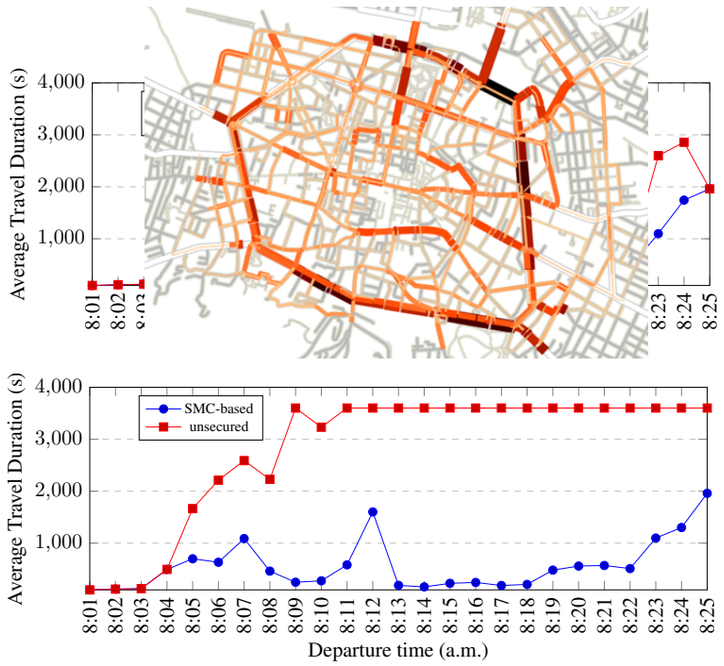
False Data Injection Attacks



TAC 2016, CDC 2017, ICCPS 2016

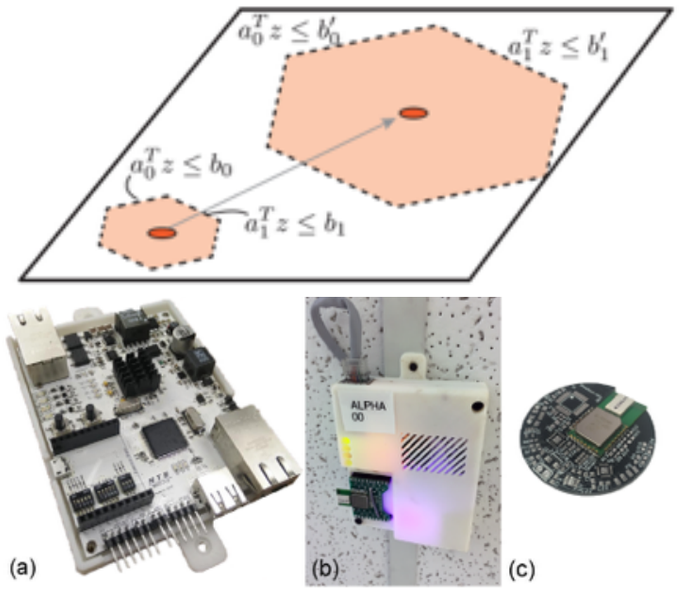
(Best paper award)

Sybil Attacks + False Data Injection



ICCPS 2018

Privacy-preserving Sensor Fusion + False Data Injection



CDC 2016, IPSN 2017

(Best demo award)

Threat Models

GPS/Sensor Spoofing Attacks

Software Vulnerabilities

Denial of Service Attacks

Privacy Leaks

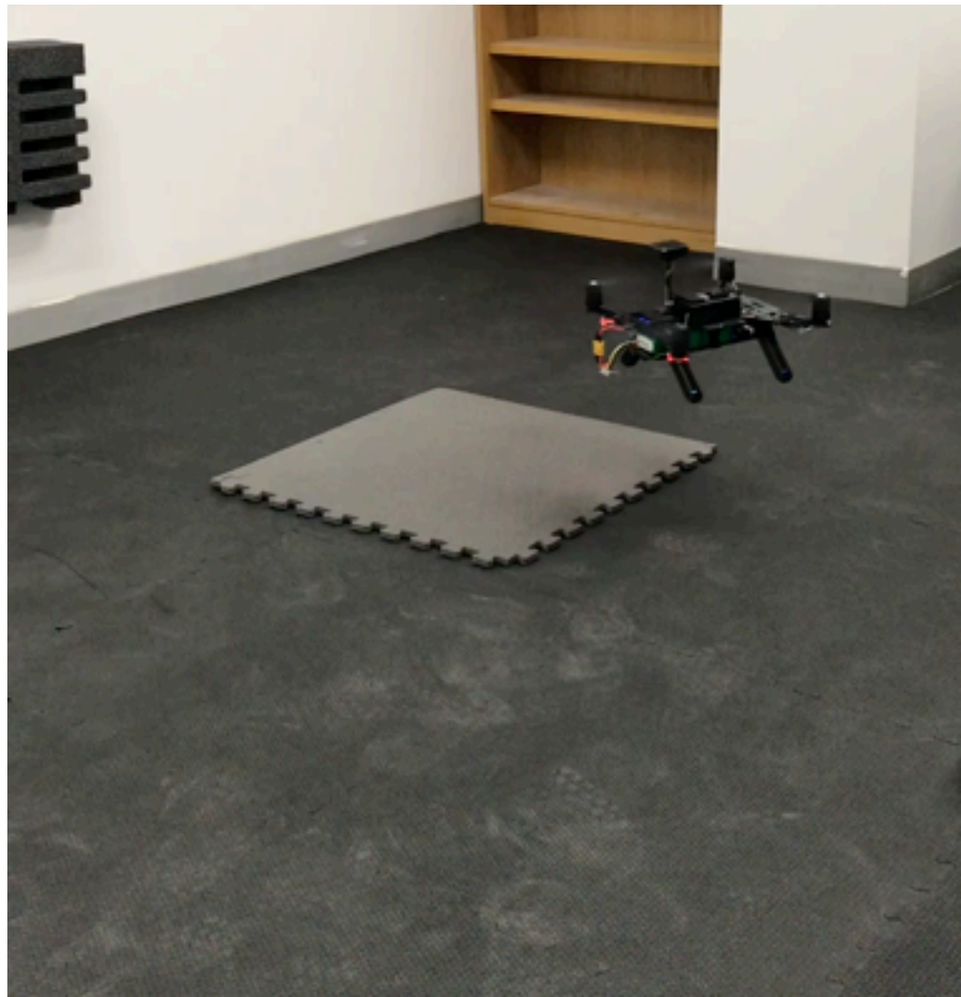
Software Vulnerabilities

- So far we assumed that our software is performing the right algorithms.

- What happens when hackers exploit vulnerabilities inside the drone software?

- Example:
  Software update while flying!

Traditional software security mechanisms do not treat this as security vulnerability!



ACKNOWLEDGMENTS

P. S. Krishnaprasad (UMD)

Paulo Tabuada (UCLA)

Mani Srivastava (UCLA)

Suhas Diggavi (UCLA)

Sanjit Seshia (Berkeley)

George Pappas (UPenn)

Alberto Sangiovanni Vincentelli (Berkeley)

Pierluigi Nuzzo (USC)

Rohitkrishna Nambiar (UMD)