Secure state-estimation and control for cyber-physical systems under adversarial attacks

Paulo Tabuada, Yasser Shoukry,
and several other collaborators

Cyber-Physical Systems Laboratory
Department of Electrical Engineering
University of California at Los Angeles

The setup

- Physical process modeled as a linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{N}_0.$$

The setup

- Physical process modeled as a linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{N}_0.$$

- A total of p sensors monitor state of plant ($y(t) \in \mathbb{R}^p$):

$$y(t) = Cx(t)$$

The setup

- Physical process modeled as a linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{N}_0.$$

- A total of p sensors monitor state of plant ($y(t) \in \mathbb{R}^p$):

$$y(t) = Cx(t) + \underbrace{e(t).}_{\text{attack vector}}$$

- Some sensors are attacked:
  - $e_i(t) \neq 0 \longrightarrow$ sensor i is attacked at time t;

The setup

- Physical process modeled as a linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{N}_0.$$

- A total of p sensors monitor state of plant ($y(t) \in \mathbb{R}^p$):

$$y(t) = Cx(t) + \underbrace{e(t).}_{\text{attack vector}}$$

- Some sensors are attacked:
  - $e_i(t) \neq 0 \longrightarrow$ sensor i is attacked at time t;
  - If sensor i is attacked, $e_i(t)$ can be **arbitrary** (no boundedness assumption, no stochastic model, etc.);

CyPhyLab UCLA

The setup

- Physical process modeled as a linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{N}_0.$$

- A total of p sensors monitor state of plant ($y(t) \in \mathbb{R}^p$):

$$y(t) = Cx(t) + \underbrace{e(t).}_{\text{attack vector}}$$

- Some sensors are attacked:
  - $e_i(t) \neq 0 \longrightarrow$ sensor i is attacked at time t;
  - If sensor i is attacked, $e_i(t)$ can be **arbitrary** (no boundedness assumption, no stochastic model, etc.);
- Set of attacked sensors (unknown) has cardinality q.

Questioning the setup

- Are physical systems really linear?

Questioning the setup

- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.

Questioning the setup

- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.
- Why is the set of attacked sensors fixed throughout the game?

Questioning the setup

- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.
- Why is the set of attacked sensors fixed throughout the game?
  - Compromising a sensor takes time.
  - While the attacker is working to compromise one additional sensor we can treat the set of attacked sensors as fixed.

Questioning the setup

- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.
- Why is the set of attacked sensors fixed throughout the game?
  - Compromising a sensor takes time.
  - While the attacker is working to compromise one additional sensor we can treat the set of attacked sensors as fixed.
- Is the attacker attacking the sensors or the communication between the sensors and the controller?

CyPhyLab UCLA

Questioning the setup

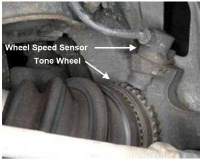
- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.
- Why is the set of attacked sensors fixed throughout the game?
  - Compromising a sensor takes time.
  - While the attacker is working to compromise one additional sensor we can treat the set of attacked sensors as fixed.
- Is the attacker attacking the sensors or the communication between the sensors and the controller?
  - Our results are independent of where and how the attack is conducted.

Questioning the setup

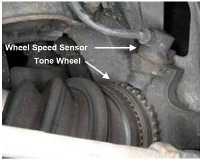
- Are physical systems really linear?
  - No! Our first results used ideas from compressed sensing and error correction over the reals, hence linearity.
  - The current understanding allows for nonlinear systems, conceptually.
- Why is the set of attacked sensors fixed throughout the game?
  - Compromising a sensor takes time.
  - While the attacker is working to compromise one additional sensor we can treat the set of attacked sensors as fixed.
- Is the attacker attacking the sensors or the communication between the sensors and the controller?
  - Our results are independent of where and how the attack is conducted.
- Can you not protect the sensors or the communication using cyber-security techniques?

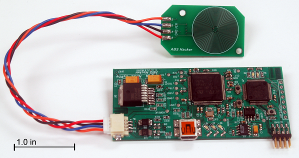
CyPhyLab UCLA

Attacking sensors



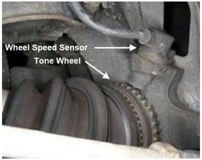
Attacking sensors

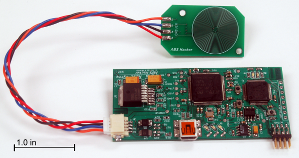


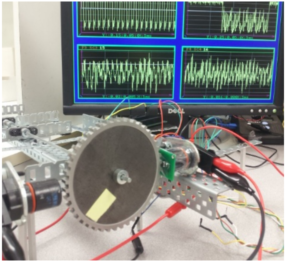


Noninvasive spoofing attacks for Anti-Lock Braking systems
Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava.
Workshop on Cryptographic Hardware and Embedded Systems, 2013 (CHES 2013).

Attacking sensors



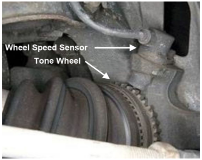


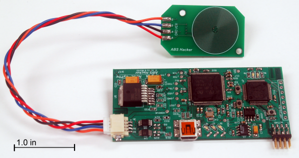


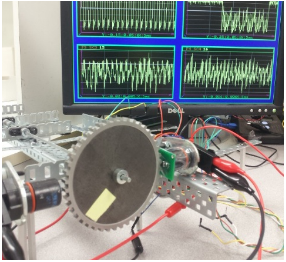
Noninvasive spoofing attacks for Anti-Lock Braking systems
Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava.
Workshop on Cryptographic Hardware and Embedded Systems, 2013 (CHES 2013).

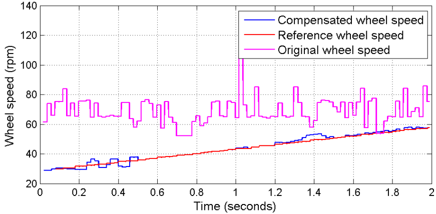
CyPhyLab UCLA

Attacking sensors









Noninvasive spoofing attacks for Anti-Lock Braking systems
Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava.
Workshop on Cryptographic Hardware and Embedded Systems, 2013 (CHES 2013).

A separation result

- The attacks are arbitrary, in particular they can be nonlinear and time-varying.
- Do we need to design a nonlinear and time-varying controller to be resilient to attacks?

A separation result

- The attacks are arbitrary, in particular they can be nonlinear and time-varying.
- Do we need to design a nonlinear and time-varying controller to be resilient to attacks?

Theorem

Consider the linear control system:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + e(t). \end{aligned}$$

If there exists a controller $u(t) = \phi(t, y(0), \ldots, y(t))$ rendering the closed-loop system exponentially stable[a] *despite an adversarial attack to q sensors then there exists a decoder $D : \mathbb{R}^{n \times p} \to \mathbb{R}^n$ that correctly reconstructs the state in n steps:*

$$x(t-n+1) = D(y(t-n+1), \ldots, y(t)).$$

[a] for a rate of decay smaller than the smallest eigenvalue of A.

A separation result

Theorem

Consider the linear control system:

$$x(t+1) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + e(t).$$

If there exists a controller $u(t) = \phi(t, y(0), \ldots, y(t))$ *rendering the closed-loop system exponentially stable*[a] *despite an adversarial attack to q sensors then there exists a decoder* $D : \mathbb{R}^{n \times p} \to \mathbb{R}^n$ *that correctly reconstructs the state in n steps:*

$$x(t-n+1) = D(y(t-n+1), \ldots, y(t)).$$

[a] for a rate of decay smaller than the smallest eigenvalue of A.

We can design a controller resilient to attacks in two steps:

1. design the decoder (observer) D;
2. design a linear static controller.

CyPhyLab UCLA

Error correction

$$
\begin{aligned}
x(t+1) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + e(t)
\end{aligned}
$$

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;

CyPhyLab UCLA

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;
- A decoder (observer) D processes observations $y(0), \ldots, y(T-1)$ and produces an estimate of the initial state $x(0)$.

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;
- A decoder (observer) D processes observations $y(0), \ldots, y(T-1)$ and produces an estimate of the initial state $x(0)$.
- We say that a decoder $D : (\mathbb{R}^p)^T \to \mathbb{R}^n$ corrects q errors after T steps if it is resilient against any attack of q sensors, i.e., if for any initial condition $x(0) \in \mathbb{R}^n$, and for any attack vectors $e(0), \ldots, e(T-1)$ on q sensors we have:

$$D(y(0), \ldots, y(T-1)) = x(0).$$

CyPhyLab UCLA

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;
- A decoder (observer) D processes observations $y(0), \dots, y(T-1)$ and produces an estimate of the initial state $x(0)$.
- We say that a decoder $D : (\mathbb{R}^p)^T \to \mathbb{R}^n$ corrects q errors after T steps if it is resilient against any attack of q sensors, i.e., if for any initial condition $x(0) \in \mathbb{R}^n$, and for any attack vectors $e(0), \dots, e(T-1)$ on q sensors we have:

$$D(y(0), \dots, y(T-1)) = x(0).$$

- We say that q errors are correctable, for the system (A, C), if there exists a decoder that can correct q errors.

CyPhyLab UCLA

Error correction

$$\begin{aligned} x(t+1) &= Ax(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

- We assume the input to be known since we design the controller. For simplicity we will take $u(t) = 0$ for all $t \in \mathbb{N}_0$;
- A decoder (observer) D processes observations $y(0), \ldots, y(T-1)$ and produces an estimate of the initial state $x(0)$.
- We say that a decoder $D : (\mathbb{R}^p)^T \to \mathbb{R}^n$ corrects q errors after T steps if it is resilient against any attack of q sensors, i.e., if for any initial condition $x(0) \in \mathbb{R}^n$, and for any attack vectors $e(0), \ldots, e(T-1)$ on q sensors we have:

$$D(y(0), \ldots, y(T-1)) = x(0).$$

- We say that q errors are correctable, for the system (A, C), if there exists a decoder that can correct q errors.
- Note: correcting $q = 0$ errors is equivalent to observability.

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

Correction of q errors
Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

- No more than $p/2$ errors can be corrected since $2q$ is necessarily smaller than p.

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

- No more than $p/2$ errors can be corrected since $2q$ is necessarily smaller than p.
- This is a fundamental limitation: if an attacker has access to more than half of the sensors ($> p/2$), it is impossible to reconstruct the state.

CyPhyLab UCLA

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

- No more than $p/2$ errors can be corrected since $2q$ is necessarily smaller than p.
- This is a fundamental limitation: if an attacker has access to more than half of the sensors ($> p/2$), it is impossible to reconstruct the state.
- Information theoretic interpretation: if a pair (A, C) is θ-sparse observable, the Hamming distance between two sequences of outputs is at least $\theta + 1$.

CyPhyLab UCLA

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

- No more than $p/2$ errors can be corrected since $2q$ is necessarily smaller than p.
- This is a fundamental limitation: if an attacker has access to more than half of the sensors ($> p/2$), it is impossible to reconstruct the state.
- Information theoretic interpretation: if a pair (A, C) is θ-sparse observable, the Hamming distance between two sequences of outputs is at least $\theta + 1$.
- Can we efficiently check sparse observability?

CyPhyLab UCLA

Correction of q errors

Necessary and sufficient conditions

- A pair (A, C) is said to be q-sparse observable if all the pairs (A, C'), obtained from (A, C) by removing q rows from C, remain observable.

Theorem

For any pair (A, C), q errors are correctable iff (A, C) is $2q$-sparse observable.

- No more than $p/2$ errors can be corrected since $2q$ is necessarily smaller than p.
- This is a fundamental limitation: if an attacker has access to more than half of the sensors ($> p/2$), it is impossible to reconstruct the state.
- Information theoretic interpretation: if a pair (A, C) is θ-sparse observable, the Hamming distance between two sequences of outputs is at least $\theta + 1$.
- Can we efficiently check sparse observability?

Proposition

Let A be a diagonalizable matrix with eigenvalues of different magnitudes. Then, for any C of compatible dimensions, q errors are correctable for the pair (A, C) iff $|supp(Cv)| > 2q$ for every eigenvector v of A.

State reconstruction under sensor attacks

Convex relaxation approach

- First approach: decoding as an ℓ_0-optimization problem. Use $\ell_0 \to \ell_1$ relaxation.

[1] cf. [Pasqualetti, Dorfler, Bullo 2010]. Thanks to Fabio Pasqualetti from UCR for the data!

State reconstruction under sensor attacks
Convex relaxation approach

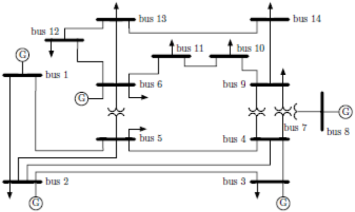
- First approach: decoding as an ℓ_0-optimization problem. Use $\ell_0 \to \ell_1$ relaxation.
- Example:
  - IEEE 14-bus power network (5 generators, 14 buses);
  - $n = 2 \times 5 = 10$ states for the rotor angles δ_i and the frequencies $d\delta_i/dt$ of each generator i;
  - $p = 35$ sensors to measure: real power injections at every bus (14 sensors), real power flows along every branch (20 sensors), rotor angle at generator 1 (1 sensor) [1].

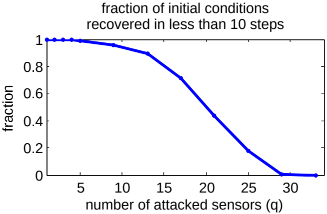
[1] cf. [Pasqualetti, Dorfler, Bullo 2010]. Thanks to Fabio Pasqualetti from UCR for the data!

State reconstruction under sensor attacks

Convex relaxation approach

- First approach: decoding as an ℓ_0-optimization problem. Use $\ell_0 \to \ell_1$ relaxation.
- Example:
  - IEEE 14-bus power network (5 generators, 14 buses);
  - $n = 2 \times 5 = 10$ states for the rotor angles δ_i and the frequencies $d\delta_i/dt$ of each generator i;
  - $p = 35$ sensors to measure: real power injections at every bus (14 sensors), real power flows along every branch (20 sensors), rotor angle at generator 1 (1 sensor) [1].





[1] cf. [Pasqualetti, Dorfler, Bullo 2010]. Thanks to Fabio Pasqualetti from UCR for the data!

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach

System Dynamics:

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

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach

System Dynamics:

$$\Sigma_a \begin{cases} x(t+1) = Ax(t) \\ y(t) \end{cases}$$

Collect τ measurements:

$$\underbrace{\begin{bmatrix} y_i(t-\tau+1) \\ y_i(t-\tau) \\ \vdots \\ y_i(t) \end{bmatrix}}_{Y_i} = \underbrace{\begin{bmatrix} C_i \\ C_iA \\ \vdots \\ C_iA^{\tau-1} \end{bmatrix}}_{\mathcal{O}_i} x + \underbrace{\begin{bmatrix} a_i(t-\tau+1) \\ a_i(t-\tau) \\ \vdots \\ a_i(t) \end{bmatrix}}_{E_i}$$

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach

System Dynamics:

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

Collect τ measurements:

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

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach

System Dynamics:

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

Collect τ measurements:

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

- For each individual sensor, we define a binary indicator variable $b_i \in \mathbb{B}$ by declaring $b_i = 1$ when the ith sensor is under attack and $b_i = 0$ otherwise.

System Dynamics:

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

Collect τ measurements:

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

- For each individual sensor, we define a binary indicator variable $b_i \in \mathbb{B}$ by declaring $b_i = 1$ when the ith sensor is under attack and $b_i = 0$ otherwise.

Problem (secure state-estimation)

For the linear control system under attack Σ_a, construct $\eta = (x, b) \in \mathbb{R}^n \times \mathbb{B}^p$ such that $\eta \models \phi$, i.e., η satisfies the formula ϕ defined by:

$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow Y_i = \mathcal{O}_i x \right) \quad \bigwedge \quad \left(\sum_{i=1}^{p} b_i \leq q \right).$$

CyPhyLab UCLA

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach

System Dynamics:

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

Collect τ measurements:

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

- For each individual sensor, we define a binary indicator variable $b_i \in \mathbb{B}$ by declaring $b_i = 1$ when the ith sensor is under attack and $b_i = 0$ otherwise.

Problem (secure state-estimation)

For the linear control system under attack Σ_a, construct $\eta = (x, b) \in \mathbb{R}^n \times \mathbb{B}^p$ such that $\eta \models \phi$, i.e., η satisfies the formula ϕ defined by:

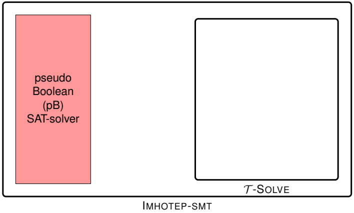
$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \| Y_i - \mathcal{O}_i x \|_2^2 \leq 0 \right) \quad \bigwedge \quad \left(\sum_{i=1}^{p} b_i \leq q \right).$$

CyPhyLab UCLA

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture I

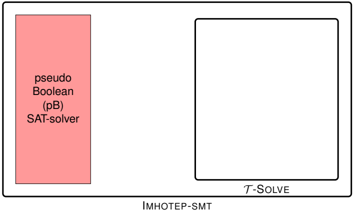
- SMT = pB-SAT solver + $\mathcal{T}$-Solver.



State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture I

- SMT = pB-SAT solver + $\mathcal{T}$-Solver.
- pB-SAT solver: solves the "boolean version" of the problem.

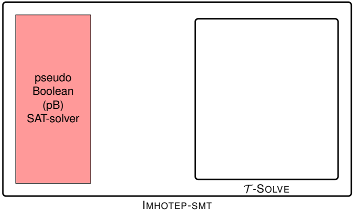


State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture I

- SMT = pB-SAT solver + $\mathcal{T}$-Solver.
- pB-SAT solver: solves the "boolean version" of the problem.
  - Original formula:

$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \|Y_i - \mathcal{O}_i x\|_2^2 \leq 0 \right)$$
$$\bigwedge \left(\sum_{i \in 1}^{p} b_i \leq q \right).$$



State reconstruction under sensor attacks

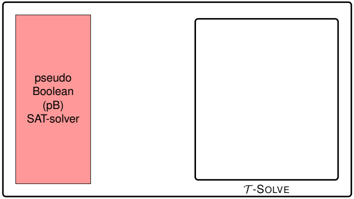
A Satisfiability Modulo Theory Approach: Lazy SMT Architecture I

- SMT = pB-SAT solver + $\mathcal{T}$-Solver.
- pB-SAT solver: solves the "boolean version" of the problem.
  - Original formula:

$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \|Y_i - \mathcal{O}_i x\|_2^2 \leq 0 \right) \wedge \left(\sum_{i\in 1}^{p} b_i \leq q \right).$$

  - Replace non-boolean variables with boolean ones

$$\phi_{initial} ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow c_i \right) \bigwedge \left(\sum_{i=1}^{p} b_i \leq q \right)$$



IMHOTEP-SMT

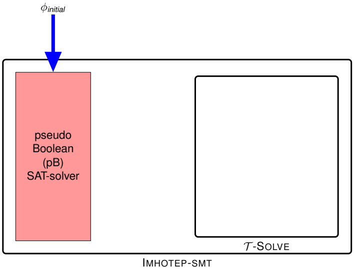
- SMT = pB-SAT solver + $\mathcal{T}$-Solver.
- pB-SAT solver: solves the "boolean version" of the problem.
  - Original formula:

$$\phi ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \|Y_i - \mathcal{O}_i x\|_2^2 \leq 0 \right) \bigwedge \left(\sum_{i \in 1}^{p} b_i \leq q \right).$$

  - Replace non-boolean variables with boolean ones

$$\phi_{initial} ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow c_i \right) \bigwedge \left(\sum_{i=1}^{p} b_i \leq q \right)$$

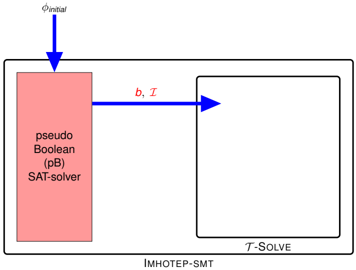
  - Pass $\phi_{initial}$ to the SAT solver.



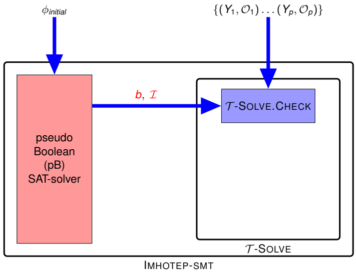
State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture II

- pB-SAT solver returns an assignment for the variable b.
- We extract which sensors are "hypothesized" to be attack free $\mathcal{I}$.

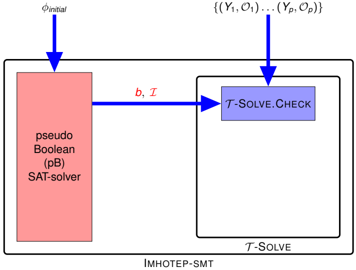


- pB-SAT solver returns an assignment for the variable b.
- We extract which sensors are "hypothesized" to be attack free $\mathcal{I}$.
- Check this assignment.



- pB-SAT solver returns an assignment for the variable b.
- We extract which sensors are "hypothesized" to be attack free $\mathcal{I}$.
- Check this assignment.

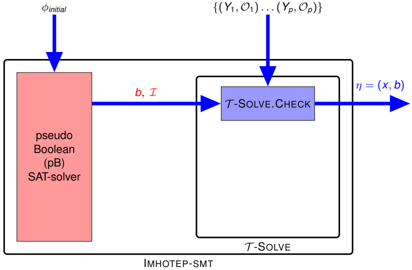
1: **Solve:**
$x := \text{argmin}_{x \in \mathbb{R}^n} \| Y_{\mathcal{I}} - \mathcal{O}_{\mathcal{I}} x \|_2^2$



- pB-SAT solver returns an assignment for the variable b.
- We extract which sensors are "hypothesized" to be attack free $\mathcal{I}$.
- Check this assignment.

Solve:
$x := \text{argmin}_{x \in \mathbb{R}^n} \| Y_{\mathcal{I}} - \mathcal{O}_{\mathcal{I}} x \|_2^2$
if $\| Y_{\mathcal{I}} - \mathcal{O}_{\mathcal{I}} x \|_2^2 = 0$ **then**
    status = `SAT`; ☺

end if
return (status, x);

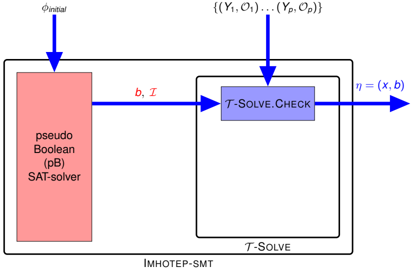


State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture II

- pB-SAT solver returns an assignment for the variable b.
- We extract which sensors are "hypothesized" to be attack free $\mathcal{I}$.
- Check this assignment.

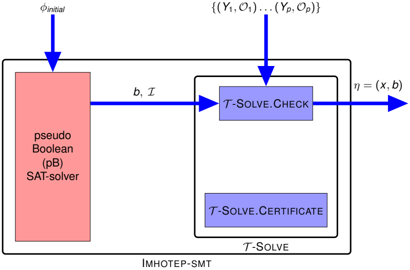
Solve:
    $x := \text{argmin}_{x \in \mathbb{R}^n} \| Y_{\mathcal{I}} - \mathcal{O}_{\mathcal{I}} x \|_2^2$
if $\| Y_{\mathcal{I}} - \mathcal{O}_{\mathcal{I}} x \|_2^2 = 0$ **then**
    status = `SAT`; ☺
else
    status = `UNSAT`; ☹
end if
return (status, x);



State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture III

- Generate "theory lemma", "counter example", or "UNSAT certificate".

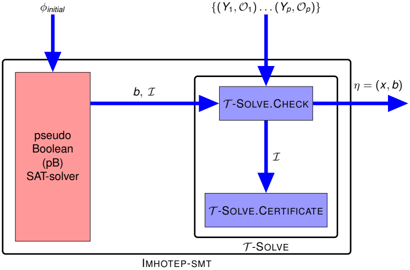


State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Lazy SMT Architecture III

- Generate “theory lemma”, “counter example”, or “UNSAT certificate”.

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$



State reconstruction under sensor attacks

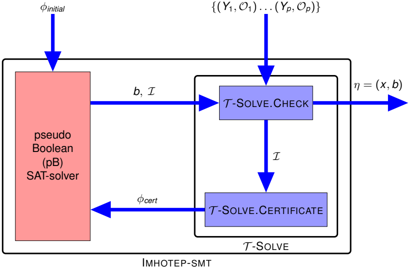
A Satisfiability Modulo Theory Approach: Lazy SMT Architecture III

- Generate "theory lemma", "counter example", or "UNSAT certificate".

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$

- Add this "certificate" to the original constraints:

$$\phi := \phi_{\text{initial}} \wedge \phi_{\text{triv-cert}}$$



State reconstruction under sensor attacks

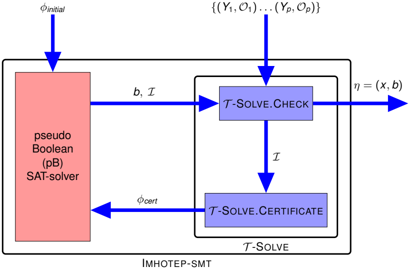
A Satisfiability Modulo Theory Approach: Lazy SMT Architecture III

- Generate "theory lemma", "counter example", or "UNSAT certificate".

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$

- Add this "certificate" to the original constraints:

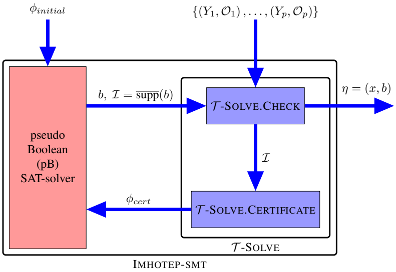
$$\phi := \phi_{\text{initial}} \wedge \phi_{\text{triv-cert}}$$



REPEAT

System Dynamics:

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



Proposition

Let the linear dynamical system Σ_a be $2q$-sparse observable. Then, IMHOTEP-SMT*:*

- *terminates,*
- *identifies the attacked sensors,*
- *and reconstructs the state.*

Moreover, the number of iterations is upper bounded by $\sum_{s=0}^{q} \binom{p}{s}$.

- Why is performance bad?

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: UNSAT certificates

- Why is performance bad?

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$

- To enhance performance, we need to generate *compact certificates*.

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: UNSAT certificates

- Why is performance bad?

$$\phi_{\text{triv-cert}} = \sum_{i \in \mathcal{I}} b_i \geq 1$$

- To enhance performance, we need to generate *compact certificates*.

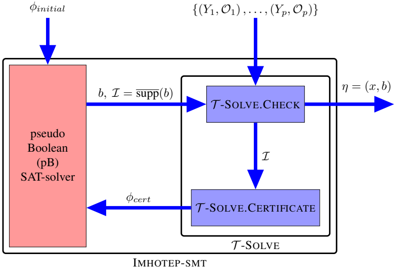
Lemma

*Let the linear dynamical system Σ_a be $2q$-sparse observable. If $\mathcal{T}$-*SOLVE.CHECK$(\mathcal{I})$ *is* $\mathtt{UNSAT}$ *then there exists a subset $\mathcal{I} \subset supp(b)$ with $|\mathcal{I}| \leq p - 2q + 1$ such that* $\mathcal{T}$-SOLVE.CHECK$(\mathcal{I}_{temp})$ *is also* $\mathtt{UNSAT}$*.*

- Trivial certificates have $p - q$ sensors.
- The proof of this lemma is constructive.
- In practice we can do better by exploiting the convex geometry (observability Gramian).

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: UNSAT certificates



Theorem

Let the linear dynamical system Σ_a be $2q$-sparse observable. Then, IMHOTEP-SMT*:*

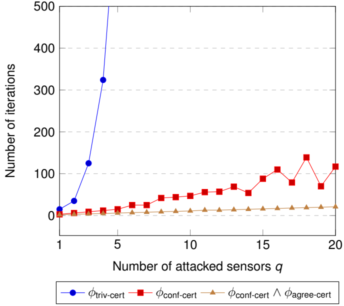
- *terminates,*
- *identifies the attacked sensors,*
- *and reconstructs the state.*

Moreover, the number of iterations is upper bounded by $\binom{p}{p-2q+1}$ (compare to: $\sum_{s=0}^{q}\binom{p}{s}$).

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Simulation results

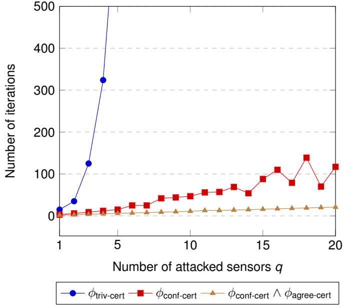
- Random system with 25 states 60 sensors and an increasing number of attacked sensors.



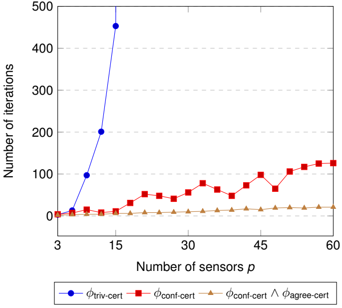
State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Simulation results

- Random system with 25 states 60 sensors and an increasing number of attacked sensors.



- Random systems with 25 states, 1/3 of sensors under attack, and increasing number of sensors.

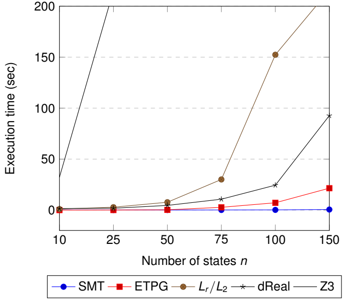


State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Simulation results

- Comparison with 2 convex-relaxation algorithms and 2 logic-based encodings.

- Random systems with 60 sensors (20 under attack) and an increasing number of states.

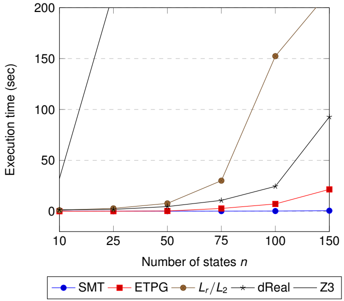


State reconstruction under sensor attacks

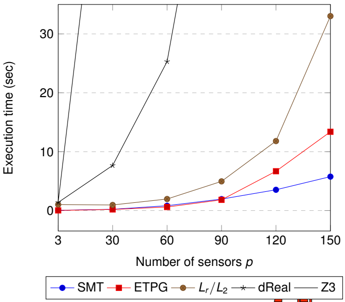
A Satisfiability Modulo Theory Approach: Simulation results

- Comparison with 2 convex-relaxation algorithms and 2 logic-based encodings.

- Random systems with 60 sensors (20 under attack) and an increasing number of states.



- Random systems with 50 states, 1/3 of sensors under attack, and increasing number of sensors.



State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Examples

State reconstruction under sensor attacks

A Satisfiability Modulo Theory Approach: Some extensions

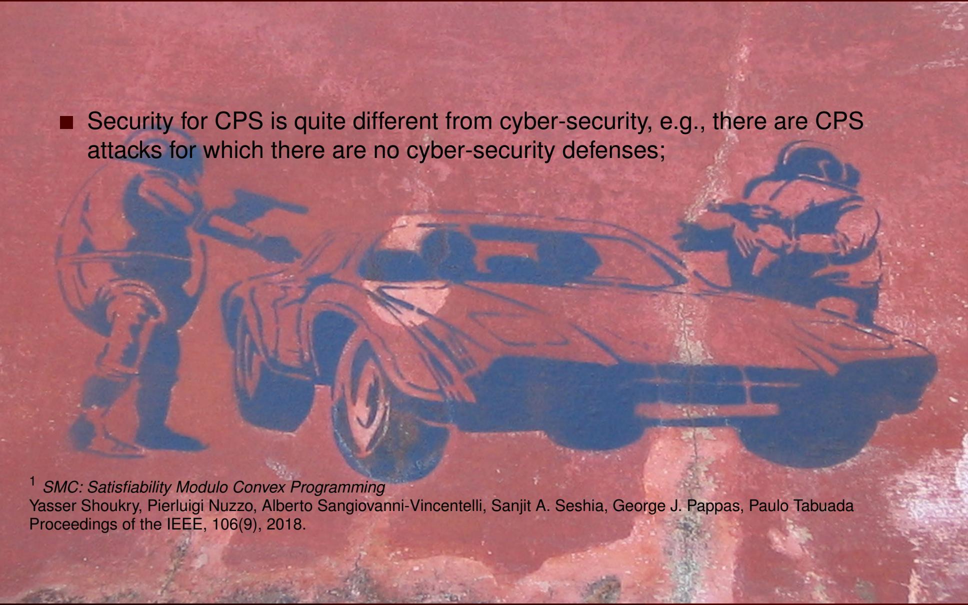
- Stochastic noise:
  - combine Kalman filters with SMT solving;
  - optimal performance: as good as a minimum mean squared error (MMSE) estimator that knows the attacked sensors[1].
- Nonlinear systems: differential flatness and applications to quadcopters[2].

[1] *Secure State Estimation Against Sensor Attacks in the Presence of Noise*
Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas Diggavi, Paulo Tabuada
IEEE Transactions on Control of Network Systems, 4(1), 49-59, 2017
Special issue on Secure Control of Cyber-Physical Systems

[2] *Secure State Reconstruction in Differentially Flat Systems Under Sensor Attacks Using Satisfiability Modulo Theory Solving*
Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, P. Tabuada
IEEE Conference on Decision and Control, 2015.

Securing Cyber-Physical Systems

Final thoughts

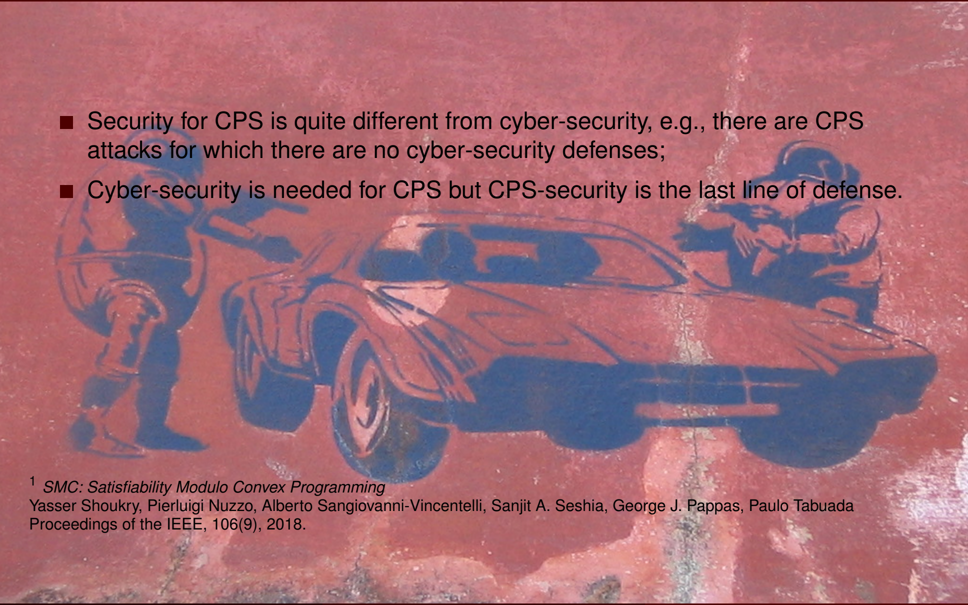


- Security for CPS is quite different from cyber-security, e.g., there are CPS attacks for which there are no cyber-security defenses;

[1] *SMC: Satisfiability Modulo Convex Programming*
Yasser Shoukry, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, Paulo Tabuada
Proceedings of the IEEE, 106(9), 2018.

Securing Cyber-Physical Systems

Final thoughts



- Security for CPS is quite different from cyber-security, e.g., there are CPS attacks for which there are no cyber-security defenses;
- Cyber-security is needed for CPS but CPS-security is the last line of defense.

[1] *SMC: Satisfiability Modulo Convex Programming*
Yasser Shoukry, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, Paulo Tabuada
Proceedings of the IEEE, 106(9), 2018.

Securing Cyber-Physical Systems

Final thoughts

- Security for CPS is quite different from cyber-security, e.g., there are CPS attacks for which there are no cyber-security defenses;
- Cyber-security is needed for CPS but CPS-security is the last line of defense.
- Challenging technical problems mixing continuous and discrete variables.

[1] *SMC: Satisfiability Modulo Convex Programming*
Yasser Shoukry, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, Paulo Tabuada
Proceedings of the IEEE, 106(9), 2018.

Securing Cyber-Physical Systems

Final thoughts

- Security for CPS is quite different from cyber-security, e.g., there are CPS attacks for which there are no cyber-security defenses;
- Cyber-security is needed for CPS but CPS-security is the last line of defense.
- Challenging technical problems mixing continuous and discrete variables.
- These techniques led[1] to Satisfiability Modulo Convex optimization (SMC), a new tool capable of handling many of these continuous+discrete challenges across a wide range of application domains (robot motion planning, etc).

[1] *SMC: Satisfiability Modulo Convex Programming*
Yasser Shoukry, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, Paulo Tabuada
Proceedings of the IEEE, 106(9), 2018.

Acknowledgements

- Students and collaborators;
- NSF, DARPA, and ARL;
- Grid Science Organisers.

For more information:
http://www.cyphylab.ee.ucla.edu/
http://www.ee.ucla.edu/∼tabuada