# Securing Dynamic Routing for Parallel Queues against Random and Strategic Sensing Faults

Qian Xie
qianxie@nyu.edu

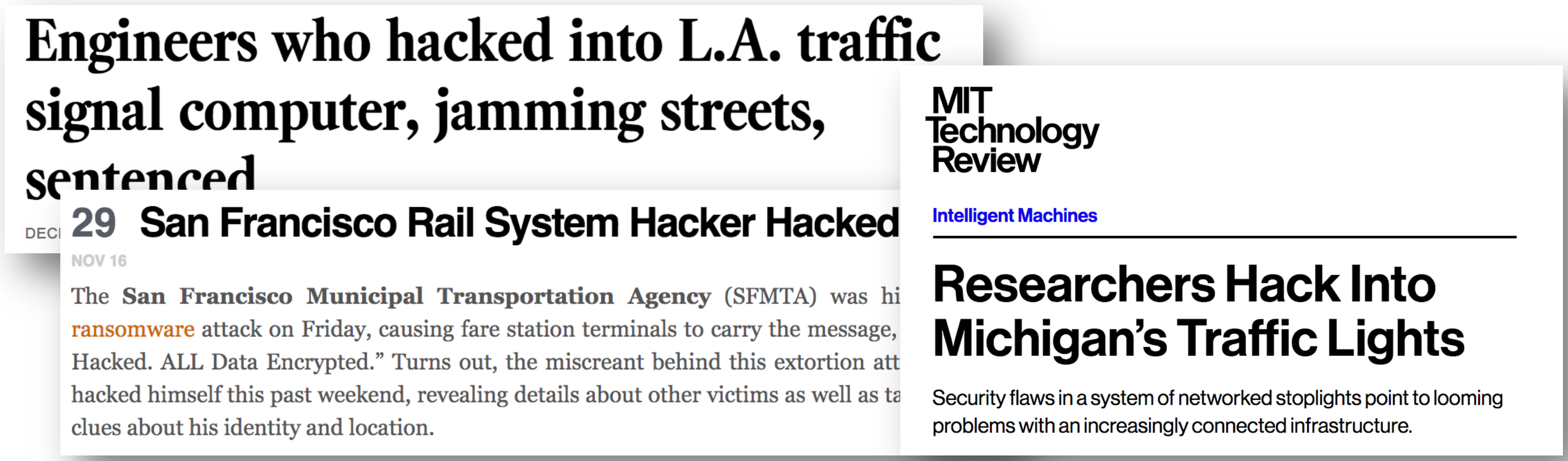
July 10, 2020

# Outline

- Introduction
  - ➢Security risks in CPSs
- Illustrating example
  - ➢Dynamic routing in ITSs
- Research questions
  - ➢Modeling & analysis
  - ➢Resource allocation
  - ➢Control design
- Model formulation
  - ➢Parallel-queuing system
  - ➢Attacker-defender game
- Results
  - ➢Stability criteria
  - ➢Equilibria properties
  - ➢Comparison with benchmark
- Extension

# Security risks in cyber-physical systems

- Cyber-physical systems rely on data flowing through the network
  - intelligent transportation systems (ITSs)
  - manufacturing systems
  - communication networks
- Cyber components are vulnerable to malicious attacks that bring security risks
  - e.g. In ITS, traffic sensors and traffic lights can be easily intruded and manipulated


**Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced**

DEC

29 **San Francisco Rail System Hacker Hacked**

NOV 16

The **San Francisco Municipal Transportation Agency** (SFMTA) was hi ransomware attack on Friday, causing fare station terminals to carry the message, Hacked. ALL Data Encrypted." Turns out, the miscreant behind this extortion att hacked himself this past weekend, revealing details about other victims as well as ta clues about his identity and location.

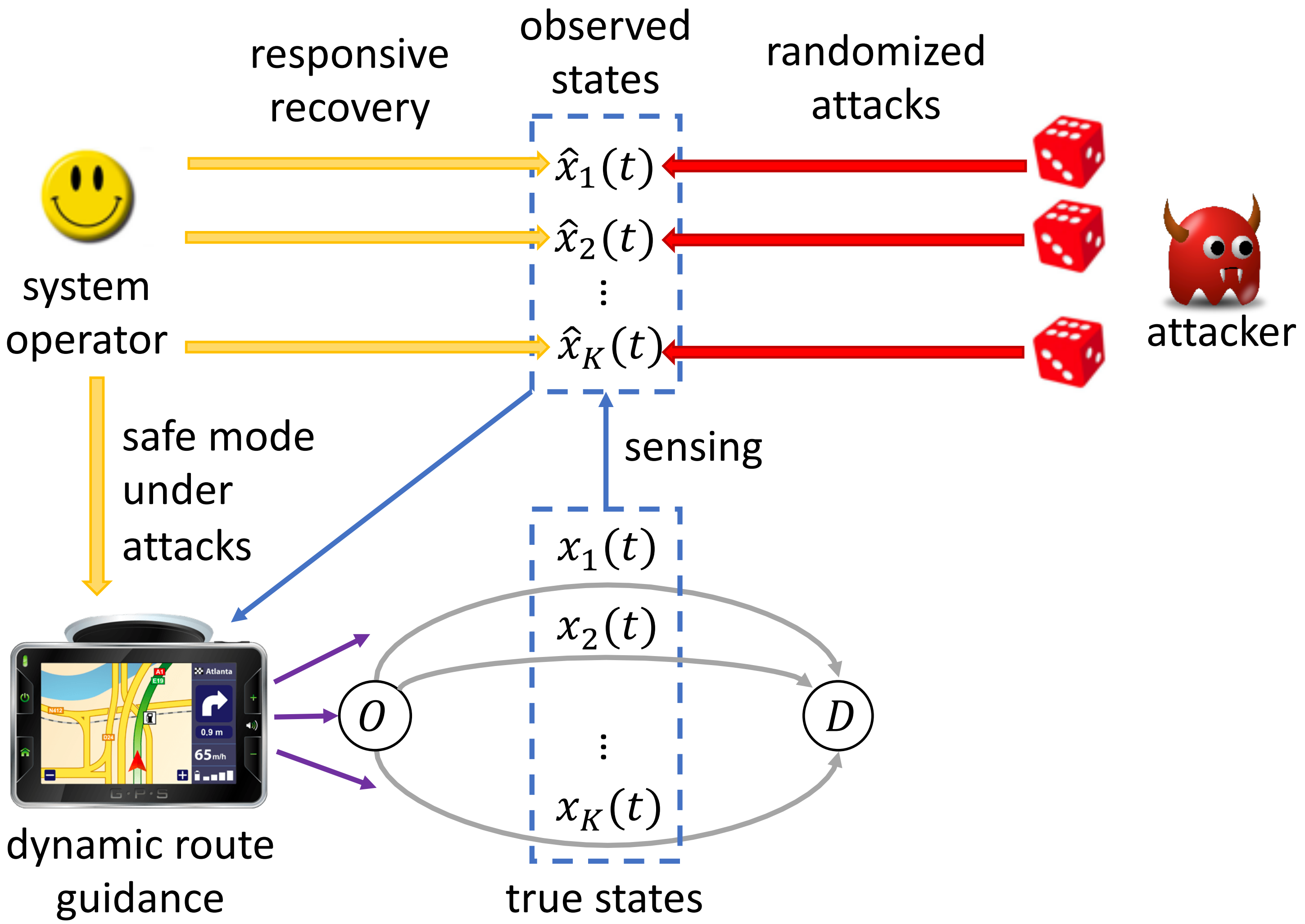
MIT Technology Review

Intelligent Machines

**Researchers Hack Into Michigan's Traffic Lights**

Security flaws in a system of networked stoplights point to looming problems with an increasingly connected infrastructure.

# Security vulnerabilities in ITSs



## Research questions

Modeling & analysis

- How to model stochastic & recurrent attacks?
- How to quantify attacker’s incentive?
- How to quantify the impact due to attacks?
- How to evaluate security risk?

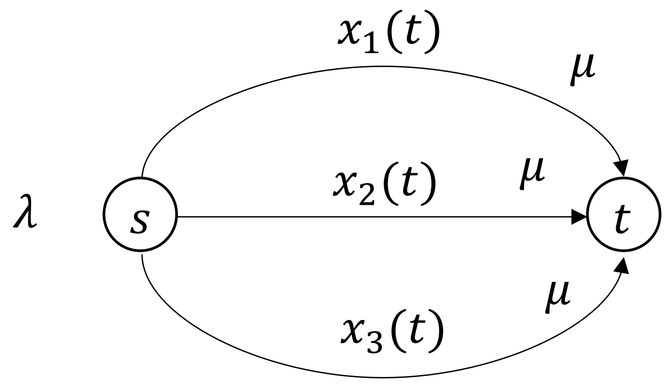
Resource allocation

- How to allocate security resources, including redundant components, diagnosis mechanisms, etc.?

Control design

- How to design traffic control strategies that are less sensitive to various types of attacks?

# Queuing model



Basic model

- Poisson arrivals of rate $\lambda$
- Parallel queuing servers with service rate $\mu$
- State: vector of queues

$$X(t) = [X_1(t), X_2(t), \ldots, X_K(t)]$$

- Dynamic routing: optimal control strategy to route jobs (e.g. vehicles, components, data packets)
- Provably optimal routing policy: send-to-shortest-queue [Ephremides, Varaiya & Walrand 80]
- Note: implementing the optimal routing policy requires perfect observation of system state $X(t)$
- If observation imperfect, then closed-loop can be worse than open-loop (e.g. round robin or Bernoulli routing)

## Failure (attacker) model

- Denial-of-service (DoS):
  - Attacker compromise sensing
  - Operator loses observation temporarily
  - With constant probability $a$, a job does not go to the shortest queue (e.g. join-a-random-queue)
- Spoofing:
  - Attacker modifies sensing
  - Operator makes decision according to manipulated sensing
  - With state-dependent probability $\alpha(x)$, an attacker manipulates the routing (e.g. send-to-longest-queue)
- Objective: balance queuing cost and attacking cost

$$V_A^*(x,\beta) = \max_\alpha \mathbb{E}[\int_0^\infty e^{-\rho t} R\big(X(t)\big) dt | X(0) = x]$$

$$\text{where } R(\xi) = |\xi| + c_b\beta(\xi) - c_a\alpha(\xi)$$

## Defender model

- Decision making:
  - With probability $\beta(x)$, the system operator (defender) secures the routing (i.e. ensuring correct routing)
- Objective: balance queuing cost and defending cost

$$V_B^*(x,\alpha) = \min_\beta \mathbb{E}[\int_0^\infty e^{-\rho t} C\big(X(t)\big)dt | X(0) = x]$$

where $C(\xi) = |\xi| + c_b \beta(\xi) - c_a \alpha(\xi)$

- Routing is compromised if and only if attacked & not defended
  - i.e. $\alpha(x) = 1$ & $\beta(x) = 0$ or $\alpha(x)\big(1 - \beta(x)\big) = 1$

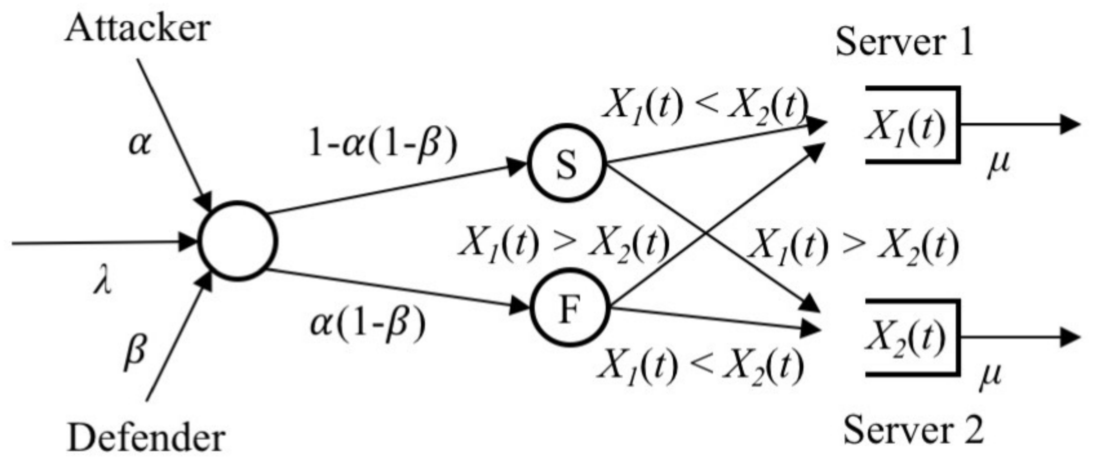
## Stability

**Theorem 1.** The n-queue system is stable if there exists a compact set $\chi_0 = [0, \theta]^n$ such that for any $x \in \chi_0^c$,

$$\alpha(x)\big(1 - \beta(x)\big) < \frac{\mu - \lambda x_{min}/|x|}{\lambda(x_{max} - x_{min})/|x|}.$$

*Proof sketch.* Consider the quadratic Lyapunov function $V(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^2$ and apply the infinitesimal generator

$$\mathcal{L}V(x) = \alpha(x)(1-\beta(x))\lambda x_{\max} + \Big(1 - \alpha(x)(1-\beta(x))\Big)\lambda x_{\min} - \sum_{i=1}^{n} \mu x_i + \frac{1}{2}\lambda + \frac{1}{2}\sum_{i=1}^{n} \mathbb{I}_{x_i>0}\mu.$$



## Security game

Infinite-horizon, dynamic, two-player zero-sum stochastic game

Markovian, state-dependent policies

**Definition 2.** The optimal attacking (resp. defending) strategy $\alpha^*$ (resp. $\beta^*$) satisfies that for any state $x \in \mathbb{Z}^n_{\geq 0}$,

$$\alpha^*(x) = \text{argmax}_\alpha V_A^*(x, \beta^*),$$
$$\beta^*(x) = \text{argmin}_\beta V_B^*(x, \alpha^*).$$

The value of the attacker (resp. defender) is $V_A^*(x, \beta^*)$ (resp. $V_B^*(x, \alpha^*)$). In particular, $(\alpha^*, \beta^*)$ is a Markovian perfect equilibrium.

**Remark.** According to Shapley's extension on minimax theorem,

$$V_A^*(x, \beta^*) = V_B^*(x, \alpha^*) = V^*(x)$$

**Proof idea.** Value iteration.

**Question.** Existence of MPE? (Countable infinite state space!)

## Dynamic programming

HJB equation

$$0 = |x| + c_b\beta(x) - c_a\alpha(x) - \rho V^*(x) + \mathcal{L}V^*(x)$$

Assume $\rho + \lambda + n\mu = 1$, it is equivalent to

$$V^*(x) = |x| + c_b\beta(x) - c_a\alpha(x) + \mu\sum_i V^*((x-e_i)^+) + \lambda\min_j V^*(x+e_j)$$
$$+ \alpha(x)(1-\beta(x))\lambda\Big(\max_j V^*(x+e_j) - \min_j V^*(x+e_j)\Big).$$

Value iteration

# Defending strategy (constant DoS probability)

**Theorem 3**. Consider a two-queue system with a constant DoS probability. The optimal defending strategy $\beta^*(x)$ has the following properties:

- Defender either defends or does not defend (no probabilistic defense), i.e. $\beta^*(x) \in \{0,1\}$
- No need to defend ($\beta^* = 0$) when $x_1 = x_2$
- Fixing $x_1 + x_2$, defend for larger $|x_1 - x_2|$
$$|x_1 - x_2| \uparrow \Rightarrow \beta^*(x) \uparrow$$
- Fixing $|x_1 - x_2|$, defend for smaller $x_1 + x_2$
$$x_1 + x_2 \uparrow \Rightarrow \beta^*(x) \uparrow$$

Proof idea: analyze properties of cumulative discounted cost using Hamiltonian Jacobian equation and induction on value iteration.

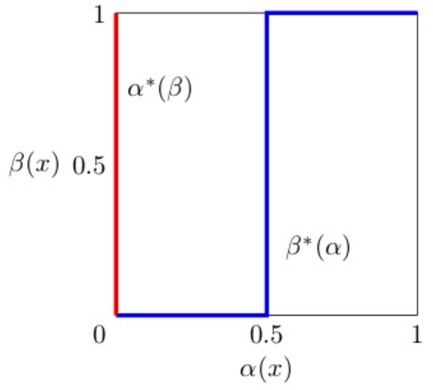
## Best response and MPE

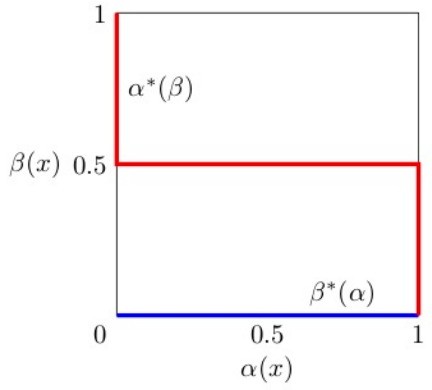
**Theorem 4**. Assume both players use Markovian strategies, then

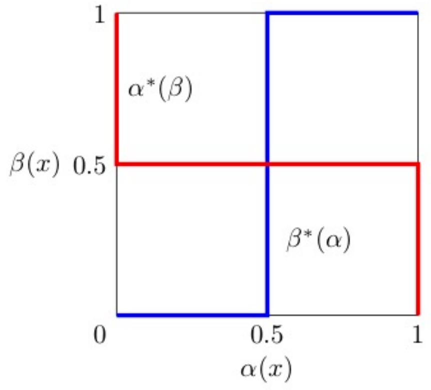
- best responses $\alpha^*(\beta)$ & $\beta^*(\alpha)$ are bang-bang (threshold-based)
- $\beta(x) = 1 \Rightarrow \alpha(x) = 0, \alpha(x) = 0 \Rightarrow \beta(x) = 0$
- the regimes of MPE are depending on $c_a,\ c_b$ and

$$d = \lambda(\max_j V^*(x+e_j) - \min_j V^*(x+e_j))$$

  - $d < c_a \Rightarrow \alpha^*(x) = 0,\ \beta^*(x) = 0;$
  - $c_a \le d < c_b \Rightarrow \alpha^*(x) = 1,\ \beta^*(x) = 0;$
  - $d > \max(c_a, c_b) \Rightarrow \alpha^*(x) = c_b/d,\ \left(1 - \beta^*(x)\right) = 1 - c_a/d$







## Numerical computation

- Hard to analytically compute equilibrium strategies; use dynamic programming
- Use stability criteria to refine the search space
- Build auxiliary matrix game at each iteration

$$M(x,V) = \left(|x| + \mu \sum_i V((x-e_i)^+)\right) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} \lambda \min_j V(x+e_j)) & c_b + \lambda \min_j V(x+e_j)) \\ -c_a + \lambda \max_j V(x+e_j) & -c_a + c_b + \lambda \min_j V(x+e_j)) \end{bmatrix}$$

- Compute the value of matrix game with Shapley-Snow method or linear programming
- How to deal with countable infinite state space? Function approximation?!

# Algorithm

**Algorithm 2** Shapley's algorithm for estimating $V \approx V^*$, $\beta \approx \beta^*$, $\alpha \approx \alpha^*$ (continuing)

```
Set $V(x) = 0$ for all $x \in \mathcal{X}$
repeat
    $\Delta \leftarrow 0$
    foreach $x \in \mathcal{X}$ do
        $v \leftarrow V(x)$
        Build auxiliary matrix game $M(x, V)$
        Compute the value $val(M)$ by using Shapley-Snow method
        $V(x) \leftarrow val(M)$
        $\Delta \leftarrow |v - V(x)|$
    end
until $\Delta < \epsilon$;
foreach $x \in \mathcal{X}$ do
    Build auxiliary matrix game $M(x, V)$
    Compute $\alpha$ and $\beta$ from $M(x, V)$ by using Shapley-Snow method
end
```