

Formally Modelling De-synchronisation Attacks

Hunting for Verifiable Controllability & Observability

James G. Wright PhD

July 9, 2026

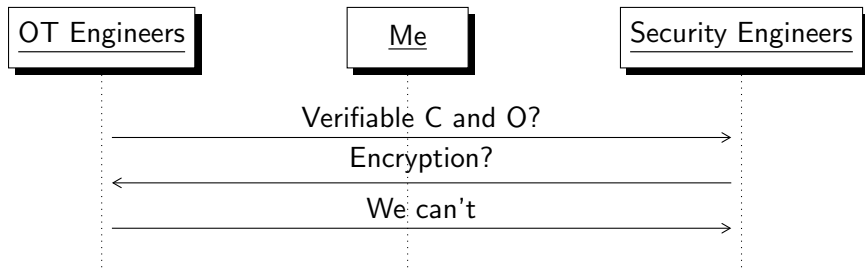
Outline

- 1 The Deadlock so far
- 2 Formulating CPS Security Verification Model
- 3 Capabilities and De-synchronisation: An example

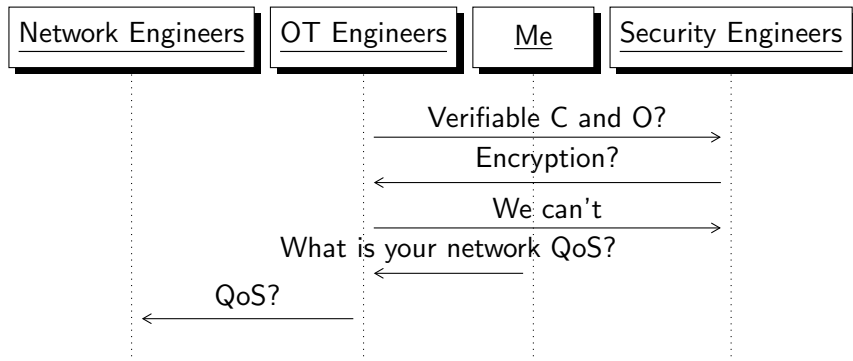
Outline

- 1 The Deadlock so far
- 2 Formulating CPS Security Verification Model
- 3 Capabilities and De-synchronisation: An example

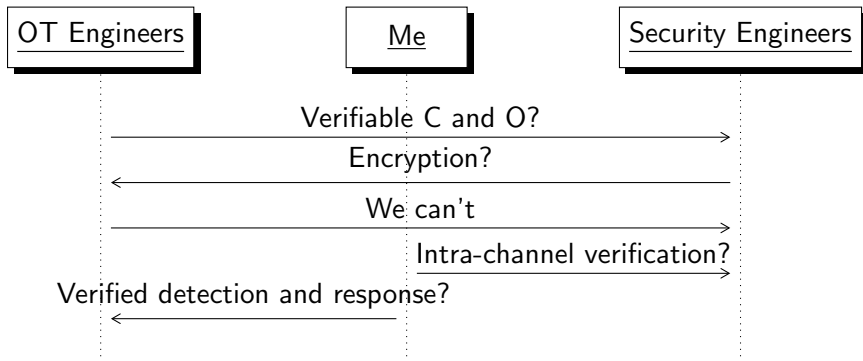
The Original Deadlock



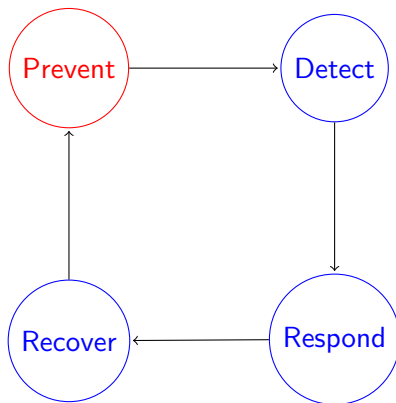
The Improved Deadlock



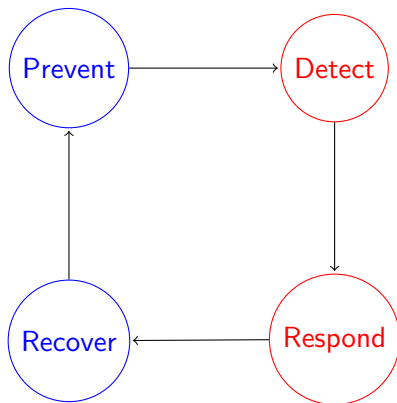
The Current Deadlock



The Traditional Verification Focus



Shifting the Verification Focus



Network Ataraxia



- Cryptography negates a risk assessment every time we click 'send'
- It restricts the set of possibilities that we have to consider

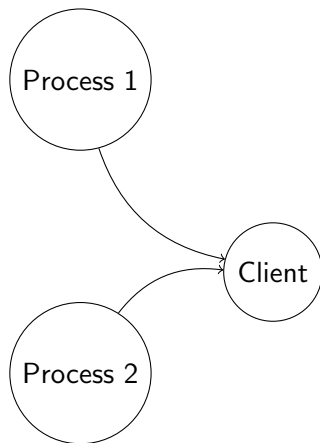
Outline

- 1 The Deadlock so far
- 2 Formulating CPS Security Verification Model
- 3 Capabilities and De-synchronisation: An example

What does CPS Security Need to Encapsulate?

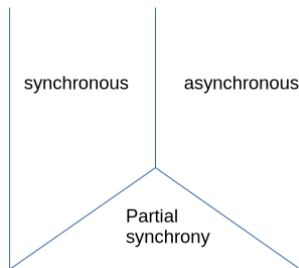
- Message flow control – availability, message expectations
 - Knowledge of physical state of the system
 - Distribution of devices – Jitter, snapshots, consensus
 - Time – time/event driven, temporal accuracy
 - Representation of the internal physical and device state
- Universal Goal: Read the Encrypted Message
 - Symbolic Goal: Remain undetected
 - Subsumes network topology
 - No limits on packet dropping
 - No limits on message injection
 - No limits on number of queries

An Old Model: Confused Deputy Problems



New Verification Game

- 'The adversary wants to change the state machine'
- What can they change?
- What knowledge & abilities do they need to make the change?
- What is the impact of the change?



The Limitations of the CPS Adversary

- Incomplete local traces and synchronization of knowledge
- Knowledge of the internal physical state
- Deductions regarding Temporal Accuracy
- Deductions about their manipulation of the communication channel
- Deductions about their manipulation of a state machine

Outline

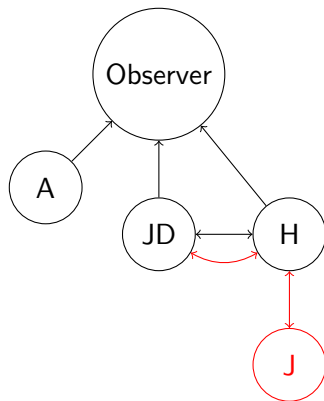
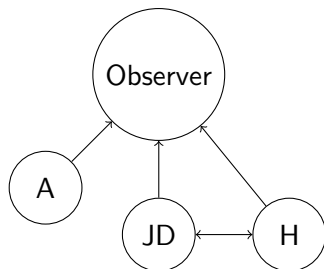
- 1 The Deadlock so far
- 2 Formulating CPS Security Verification Model
- 3 Capabilities and De-synchronisation: An example



POLITICS

The Trump Administration Accidentally Texted Me Its War Plans

The Game



Vector Clocks

$$\begin{aligned} SO \equiv & \nu VC_{init}^i.cm(VC_{init}^i) \mid (cc < VC_{\alpha}^i > .IncEle(VC_{\alpha}^i, i).cm(VC_{\alpha}^i))! \mid \\ & (cm < VC_m^i > .(cc(VC_m^i) + cs(VC_m^i) + cr(VC_m^i)))! \mid \\ & (cs < VC_o^i > .IncEle(VC_o^i, i).(cm(VC_o^i) \mid tr^{i,\mu}(VC_o^i)))! \mid \\ & ((tr^{\mu,i} < VC_r^{\mu} > \mid cr(VC_s^i).IncEle(VC_s^i, i)).MaxVec(VC_r^{\mu}, VC_s^i).cm(VC_s^i))! \\ & (1) \end{aligned}$$

$$|(cuc < VC_u^i > .tr^{i,SCIF-O}(VC_u^i))! \quad (2)$$

$$\begin{aligned} & ((cs < VC_o^i > |\nu m).IncEle(VC_o^i, i).(cm(VC_o^i)|(h^{s,p}(m)))! | \\ & (((h^{s,p} < sm >)|cr(VC_s^i).IncEle(VC_s^i, i)).MaxVec(VC_r^\mu, VC_s^i).cm(VC_s^i))! | \\ & (h^{s,p} < um >)|!(h^{s,p}(ue))! \end{aligned} \quad (3)$$

$$(a < z >)! | (\nu x.b(x))! \quad (4)$$

$$(sigr^{i,sig} < cn, r > .fu(cn, r))! | (fu < cn, r > .sigs^{sig,r}(cn))! \quad (5)$$

Fail States

$$\begin{aligned} JD | un^{jd,h} \langle nrms \rangle \mid un^{h,jd}(nrmr) \\ H | un^{h,jd} \langle nrms \rangle \mid un^{jd,h}(nrmr) \end{aligned} \quad (6)$$

$$JD \mid a^{j,u} \langle nrms \rangle \quad (7)$$

Lessons for CPS

- ① The Controllers are delegating capabilities, **not data** to sensors/actuators.
- ② The processes, **not the channel**, need to be made secure and resilient
- ③ The affect of the CPS adversary **is not** deceit!

Questions?