Cyber-Physical Systems Security: A Pacemaker Case Study

Partha Roop

BioEmulation Research Group
The University of Auckland

26 March 2021

Outline

Acknowledgements

This presentation is a result of collaborations with several colleagues and PhD students.

- Cooperation with Stavros Tripakis's group at UC Berkeley and Aalto University on run-time verification and enforcement.
- The follow up cooperation with Srinivas Pinisetty and Gerardo Schneider from Chalmers in 2017 on Pacemaker security.
- Joint work with PhD student Hammond Pearce on CPS security for smart grids. We are exploring hardware security [1].
- The following publications are relevant for this presentation. [2]–[4].

---

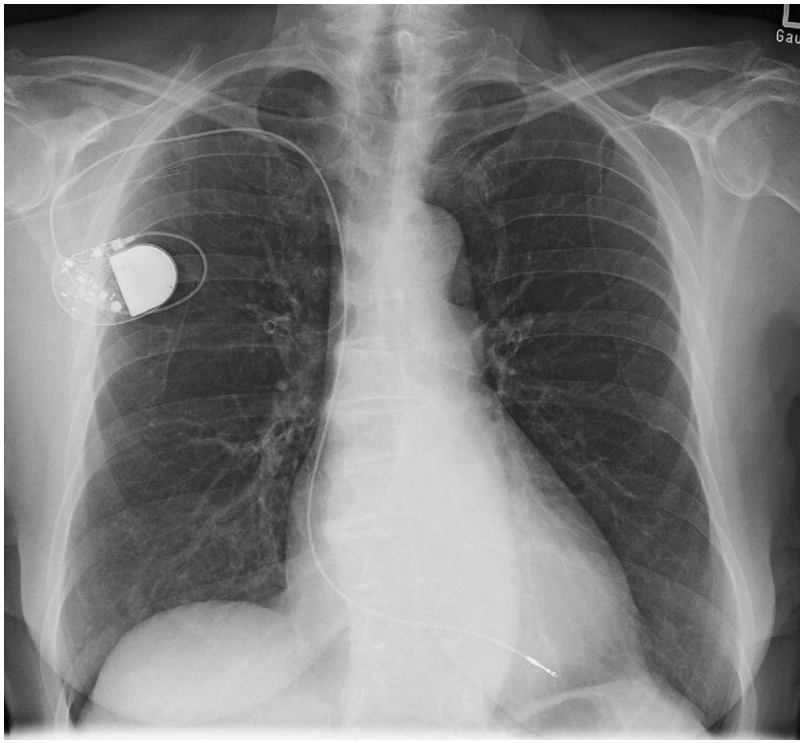
[1]. H. Pearce, S. Pinisetty, P. S. Roop, *et al.*, "Smart i/o modules for mitigating cyber-physical attacks on industrial control systems," *IEEE Transactions on Industrial Informatics*, 2019

[2]. S. Pinisetty, P. S. Roop, S. Smyth, *et al.*, "Runtime enforcement of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, p. 178, 2017

[3]. S. Pinisetty, P. S. Roop, S. Smyth, *et al.*, "Runtime enforcement of reactive systems using synchronous enforcers," in *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software*, ACM, 2017, pp. 80–89

[4]. S. Pinisetty, P. S. Roop, V. Sawant, *et al.*, "Security of pacemakers using run-time verification," in *Proceedings of the 16th ACM-IEEE International Conference on Formal Methods and Models for System Design*, IEEE, 2018 (October)

Implantable Medical Devices (IMDs)

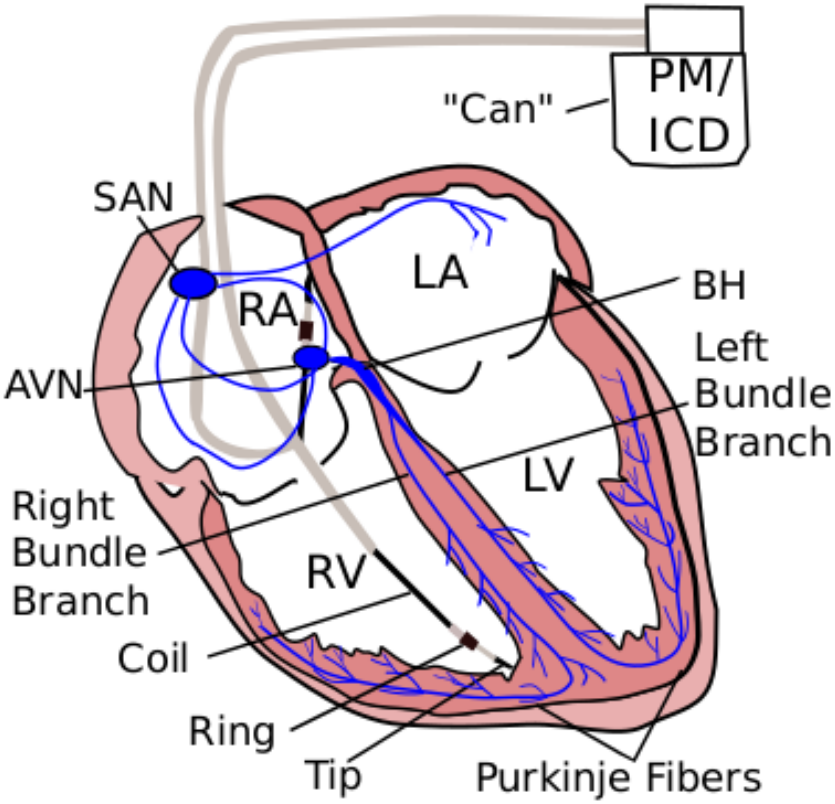


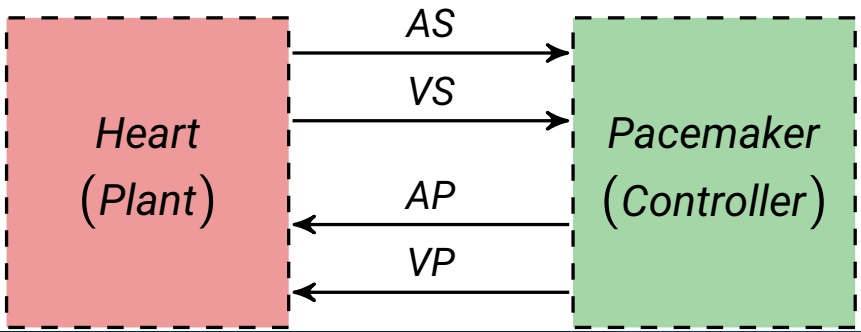
[5]

- Include pacemakers, defibrillators, insulin pumps, neurological pulse generators, ...
- **Safety-critical operation**: medical emergencies on malfunctions

[5]. T. Zimmerman. (Jun. 2012), VVI pacemaker THWZ, CC 3.0, [Online]. Available: `https://commons.wikimedia.org/wiki/File:VVI_Schrittmacher_THWZ.jpg`

Pacemaker IMD





- Heart and Pacemaker communicate through **4** signals
  - *AS* and *VS* from the heart
  - *AP* and *VP* from the pacemaker
- Pacemaker ensures timing properties between signals

Industry trend

IMDs are becoming “smarter” and more connected

- Increasingly complex sensors + software
- Wireless, internet-enabled features [6]

[6]. L. Pycroft and T. Z. Aziz, “Security of implantable medical devices with wireless connections: The dangers of cyber-attacks,” *Expert Review of Medical Devices*, vol. 15, no. 6, pp. 403–406, 2018

Unintended consequences

A new potential for *malicious attacks*

- "We are aware of hundreds of medical devices that have been infected by malware" — Bill Maisel, FDA [7]
- Notable examples:
  - Pacemakers which give deadly shocks to their patients [8]
  - Pumps remotely programmed to deliver incorrect insulin levels [9]
  - DoS attacks on implantable cardiac defibrillators [10]

---

[7]. C Weaver, "Patients put at risk by computer viruses," *Wall Street Journal*, 2013

[8]. J Kirk, "Pacemaker hack can deliver deadly 830-volt jolt," *Computerworld*, vol. 17, 2012

[9]. J. D. Rockoff, "J&J warns insulin pump vulnerable to cyber hacking," *Wall Street Journal*, 2016

[10]. E. Marin, D. Singelée, F. D. Garcia, *et al.*, "On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ser. ACSAC '16, Los Angeles, California, USA: ACM, 2016, pp. 226–236. [Online]. Available: `http://doi.acm.org/10.1145/2991079.2991094`

Problem space

Traditional security mechanisms suggested [11] but may not be suitable

- Low-power long-life devices may not be capable of de/encryption [12]
- SW updates not often provided due to regulatory framework [13]
- **In practice: impractical/infeasible to secure *all* attack vectors** [14]

---

[11]. U.S. Food and Drug Administration, "Postmarket management of cybersecurity in medical devices," Guidance for Industry, Food, and Drug Administration Staff, Tech. Rep., 2016. [Online]. Available: `https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf`

[12]. D Takahashi, "Insulin pump hacker says vendor Medtronic is ignoring security risk," *Venturebeat*, 2011. [Online]. Available: `https://venturebeat.com/2011/08/25/insulin-pump-hacker-says-vendor-medtronic-is-ignoring-security-risk/`

[13]. D Clery, "Could your pacemaker be hackable?" *Science*, vol. 347, no. 6221, pp. 499–499, 2015

[14]. J. Sametinger, J. Rozenblit, R. Lysecky, *et al.*, "Security challenges for medical devices," *Commun. ACM*, vol. 58, no. 4, pp. 74–82, Mar. 2015. [Online]. Available: `http://doi.acm.org/10.1145/2667218`

Proposed solutions: Literature

Two general approaches

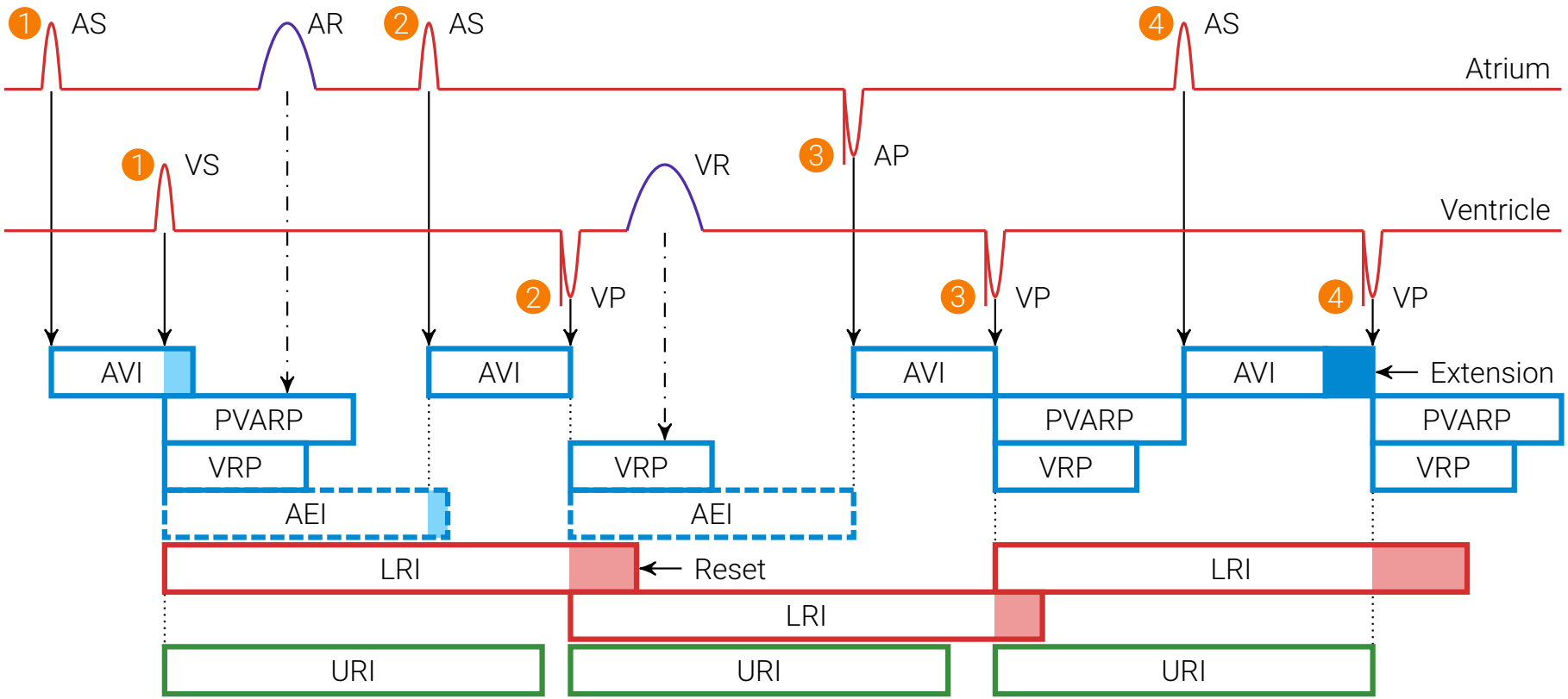
- Access Control (still has potential to be bypassed)
  - E.g. Heart2Heart [15]
  - E.g. Ultrasonic bounding [16]

---

[15]. S. Gollakota, H. Hassanieh, B. Ransford, *et al.*, "They can hear your heartbeats: Non-invasive security for implantable medical devices," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 2–13, Aug. 2011. [Online]. Available: `http://doi.acm.org/10.1145/2043164.2018438`

[16]. K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, *et al.*, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09, Chicago, Illinois, USA: ACM, 2009, pp. 410–419. [Online]. Available: `http://doi.acm.org/10.1145/1653662.1653712`

Pacemaker Timing Requirements (EGMs)



Equivalent ECG timings

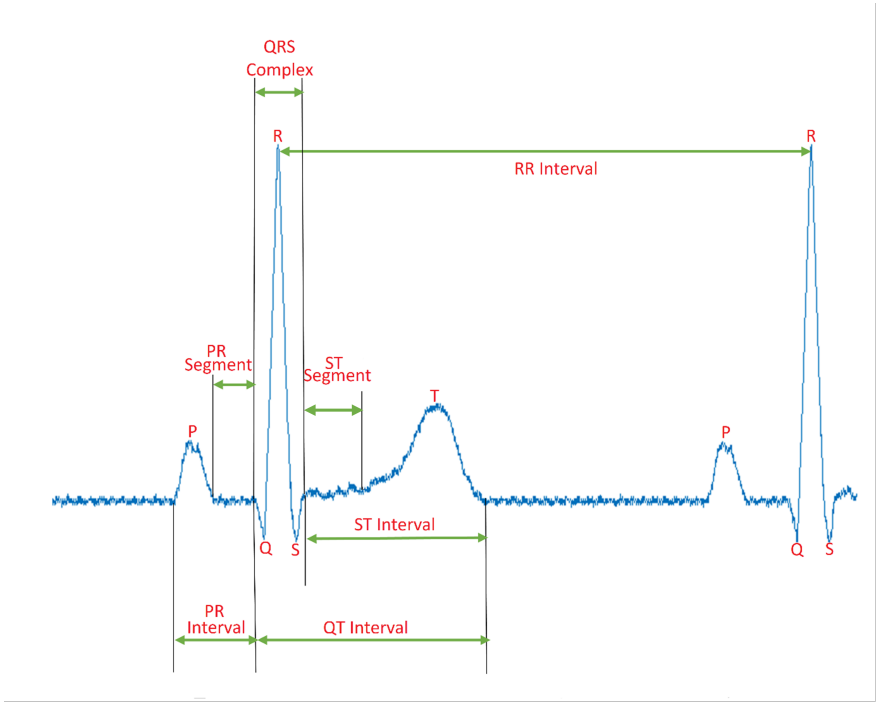
P_1 P — *wave* and R — *wave* cannot happen simultaneously.

P_2 R — *wave* must arrive within PR interval after a P — *wave*.

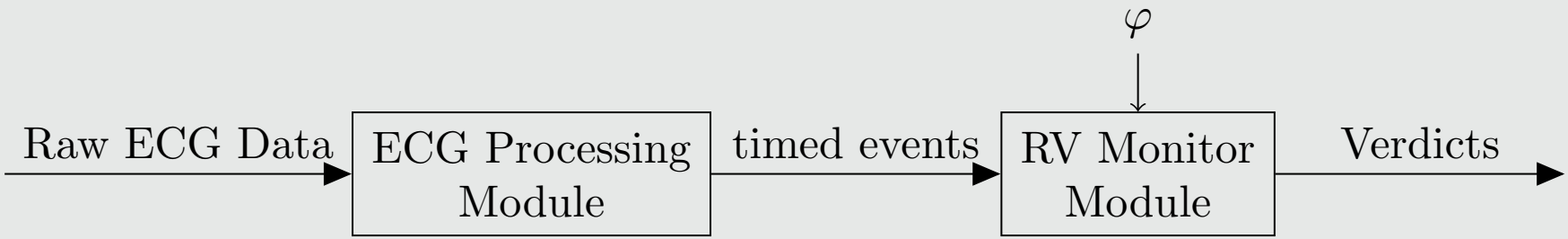
P_3 P — *wave* must be true within R — P interval after an R — *wave*.

P_4 After an R — *wave*, another R — *wave* can come only after R — P interval.

P_5 After an R — *wave*, another R — *wave* should come within R — R interval.



Runtime verification



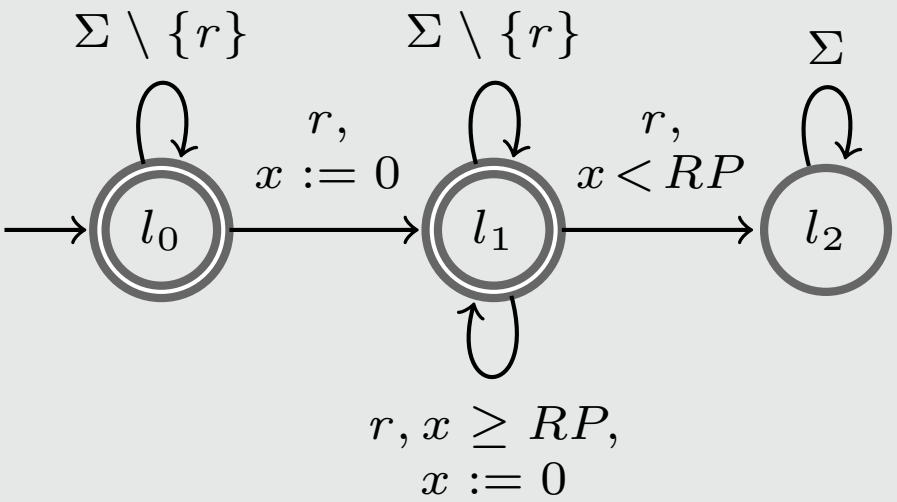
- Does σ satisfy φ ?
- φ is a **timed automaton**.
- Output: stream of **verdicts**.

Runtime verification example

Example P4

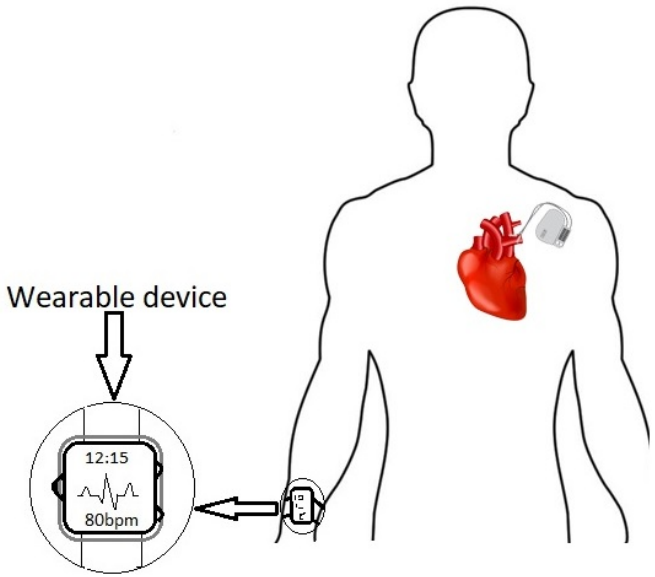
Definition (Timed automata)

A *timed automaton* $\mathcal{A} = (L, l_0, X, \Sigma, \Delta, F)$ is a tuple, s.t. L is a finite set of *locations* with the *initial location* $l_0 \in L$, a finite set of *clocks* X, Σ is a finite set of *actions*, $\Delta \subseteq L \times \mathcal{G}(X) \times \Sigma \times 2^X \times L$ is the *transition relation*. $F \subseteq L$ is a set of *accepting locations*.



Overview of the solution

- Pacemaker timing parameters are programmed simultaneously on the monitoring device and the pacemaker.
- The wearable device monitors the familiar ECG to ensure that there have been no hacks.
- In the event of any timing violation an alarm is sounded.



Illustration

Example P4

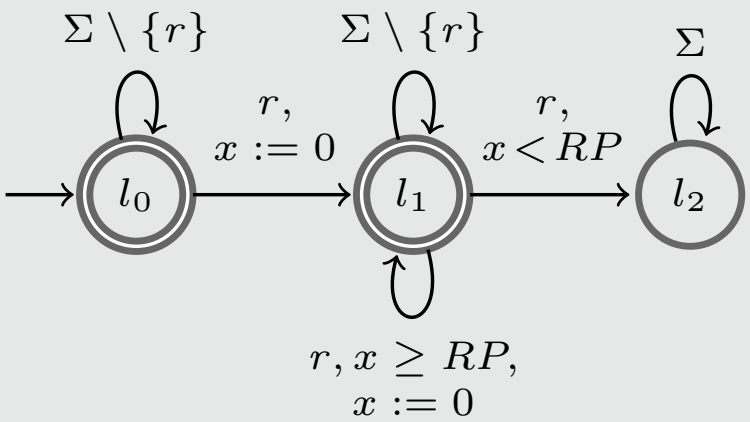


Table: Property P_4 monitoring with $RP = 900$

σ	$M_\varphi(\sigma)$
$(50, p)$	c_{true}
$(50, p) \cdot (208, r)$	c_{true}
$(50, p) \cdot (208, r) \cdot (300, p)$	c_{true}
$(50, p) \cdot (208, r) \cdot (300, p) \cdot (451, r)$	false

Limitation of RV Monitors

- Monitoring can detect but is unable to intervene.
- CPS attacks are complex and vulnerabilities may be exploited more easily than conventional cyber security.
- Run-Time Enforcement has some interesting potential.

Cyber-Physical System Attacks [17]

Example Targeted Attacks

- (2000) Maroochy Shire wastewater attack, where raw sewage was released around a town by ex-employee.
- (2006) Los Angeles traffic system hack, disrupting four of the busiest intersections for days.
- (2008) Turkish pipeline explosion by suspected Russian operators to cut off oil to Georgia.
- (2008) Pacific Energy Resources SCADA attack, where system functions were impaired by ex-employee.
- (2008) Lodz, Poland, tram system was taken over by a teen hacker, causing injuries.

Cyber-Physical System Attacks [17]

Example Targeted Attacks

- (2009) Well-known Stuxnet attack on Iranian centrifuges.
- (2011) Springfield IL water distribution malfunction, pump destroyed, attributed to Romanian hacker.
- (2014) Unnamed German Steel mill, hackers caused massive damage to equipment by disabling shut-off procedures, including a blast furnace.
- (2015) Jeep Cherokee, remote hijacking leading to total control by researchers Charlie Miller and Chris Valasek.
- (2016) Tesla S, remote hacking of some functions by Chinese researchers.

Cyber-Physical System Attacks [17]

Example Targeted Attacks

- (2016) Unnamed water facility, where Syrian hacktivists took control of PLCs controlling toxic chemicals.
- (2016) San Francisco municipal rail system ransomware hack, free rides for commuters.
- (2017) Austrian ski resort ransomware hack, "smart locks" compromised, guests couldn't access their rooms.
- (2017) Well-known WannaCry ransomware attack, which also infected hospital equipment such as MRI scanners, radiotherapy machines, oncology equipment etc.
- (2017) U.S. DHS reports govt. team hacking passenger jet controls.

Classifying attacks [17], [18]

In order to mitigate attacks, we must understand them.

Classifying attacks [17], [18]

In order to mitigate attacks, we must understand them.

Passive Attacks

Exfiltrate data, gain knowledge of system, non-damaging.

Classifying attacks [17], [18]

In order to mitigate attacks, we must understand them.

Passive Attacks

Exfiltrate data, gain knowledge of system, non-damaging.

Disruptive Attacks

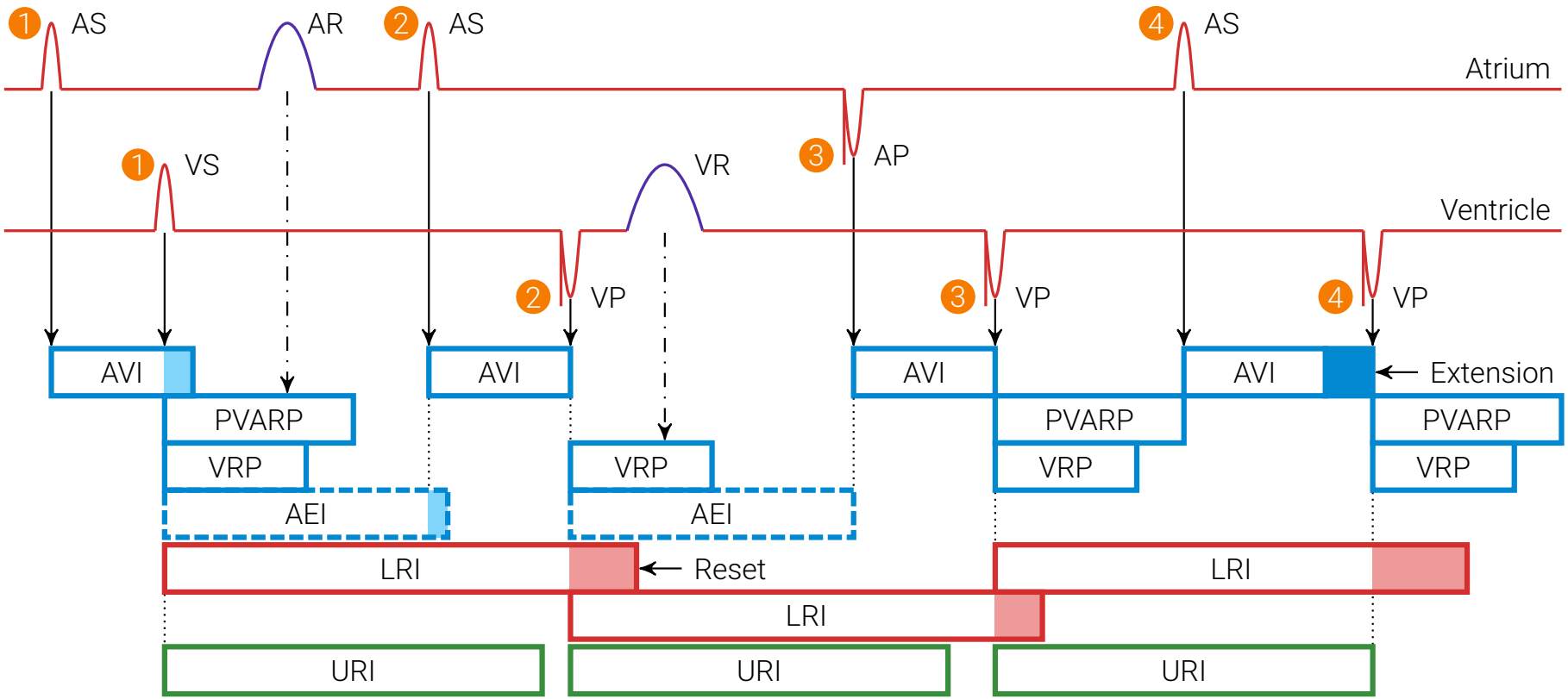
- Physical-Cyber Attack - originates in physical domain, aims to disrupt cyber domain, e.g. cutting cables.
- Cyber-Physical Attack - originates in cyberspace and impacts ability for cyber system to control physical process, e.g. DoS, Cryptolocker.
- Cyber-Kinetic Attack - originates in cyberspace and intends to cause tangible physical damage, e.g. Stuxnet.

Classifying attacks [17], [18]

Table: Classified list of attacks

Cyber-Physical	Cyber-Kinetic
2006 LA Traffic	2000 Maroochy Shire Wastewater
2008 Pacific Energy Resources	2008 Turkish Pipeline
2016 Syrian Water Facility	2008 Lodz Trams
2016 San Francisco Rail	2009 Stuxnet
2017 Austrian Ski Resort	2011 Springfield Water Distribution
2017 WannaCry	2014 German Steel Mill
	2015 Jeep Cherokee
	2016 Tesla S
	2017 Passenger Jet

Pacemaker Timing



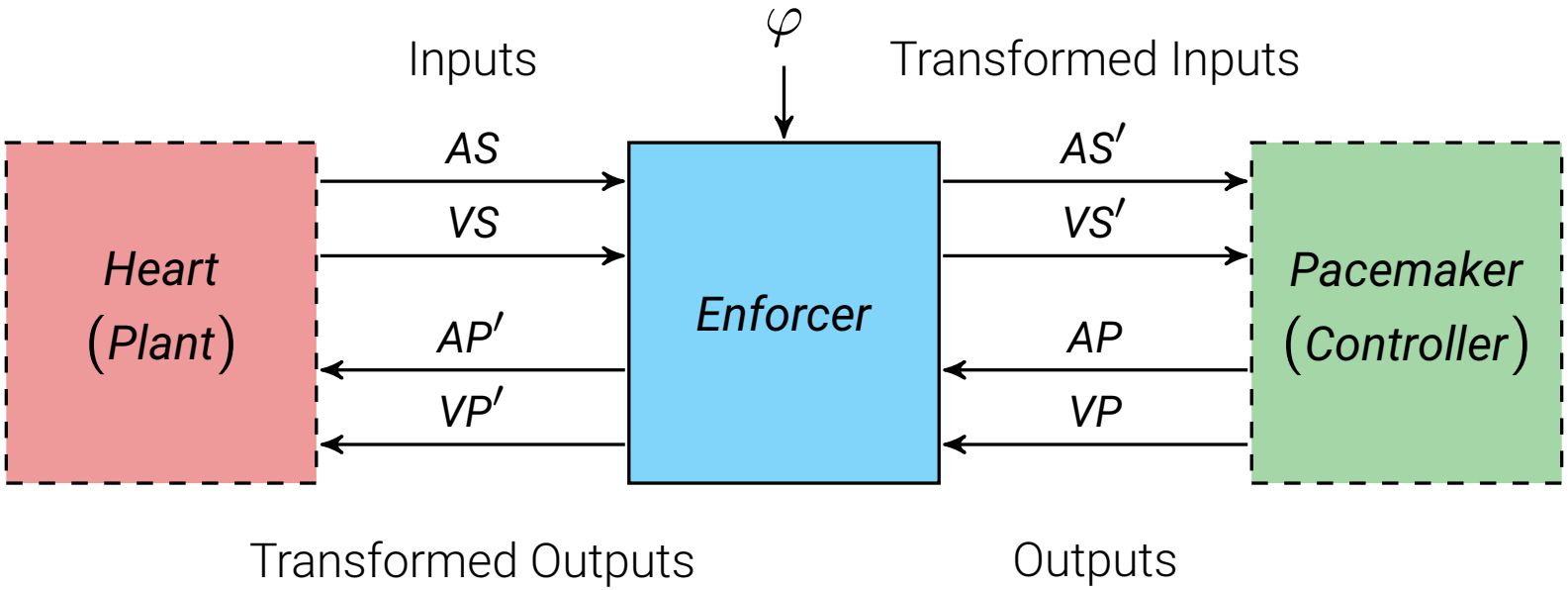
Discretised Properties – φ

- Real valued times AVI, AEI, URI, LRI
- Discrete valued times AVI_{cycles}, AEI_{cycles}, URI_{cycles}, LRI_{cycles}

Properties

- P_1: AP and VP cannot happen simultaneously.
- P_2: VS or VP must be true within AVI_{cycles} after an atrial event AS or AP.
- P_3: AS or AP must be true within AEI_{cycles} after a ventricular event VS or VP.
- P_4: After a ventricular event, another ventricular event can happen only after URI_{cycles}.
- P_5: After a ventricular event, another ventricular event should happen within LRI_{cycles}.

Proposed Approach



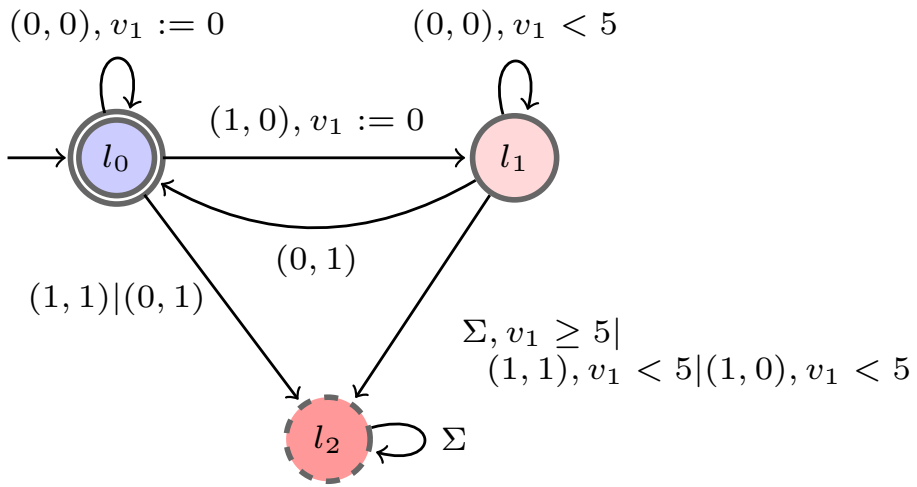
Discrete Timed Automata – $\mathcal{A}$

A policy specification language from [19]

- Automata extended with integer variables as discrete clocks
  - Discrete time more efficient than Dense time
- Clocks count in synchronous "ticks"

Example Property as DTA – $\mathcal{A}_\varphi$

S_1: *"A and B cannot happen simultaneously, A and B alternate starting with an A. B should be true with in 5 ticks after A occurs."*



[19]. S. Pinisetty, P. S. Roop, S. Smyth, *et al.*, "Runtime enforcement of cyber-physical systems," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 5s, 178:1–178:25, Sep. 2017. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/3126500

DTA Restrictions

Deterministic DTA

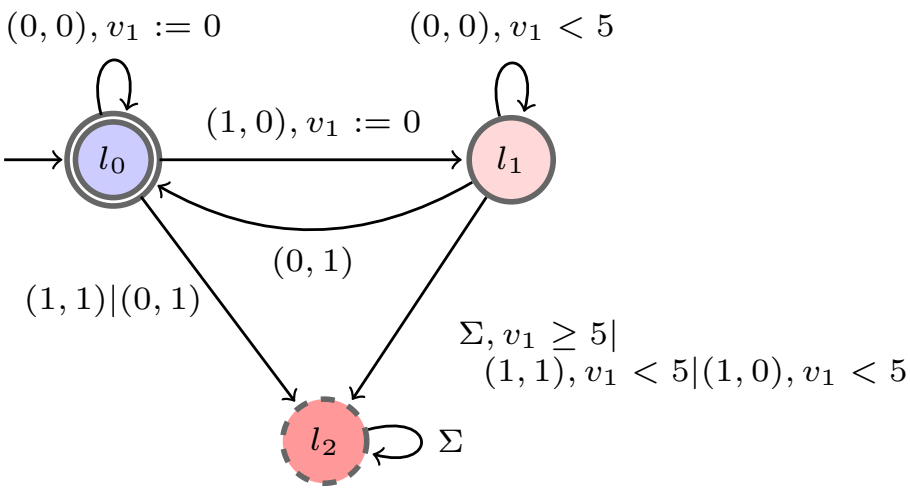
- For any location l and any two distinct transitions $(l, g_1, a, Y_1, l'_1) \in \Delta$ and $(l, g_2, a, Y_2, l'_2) \in \Delta$ with same source l, the conjunction of guards $g_1 \wedge g_2$ is unsatisfiable.

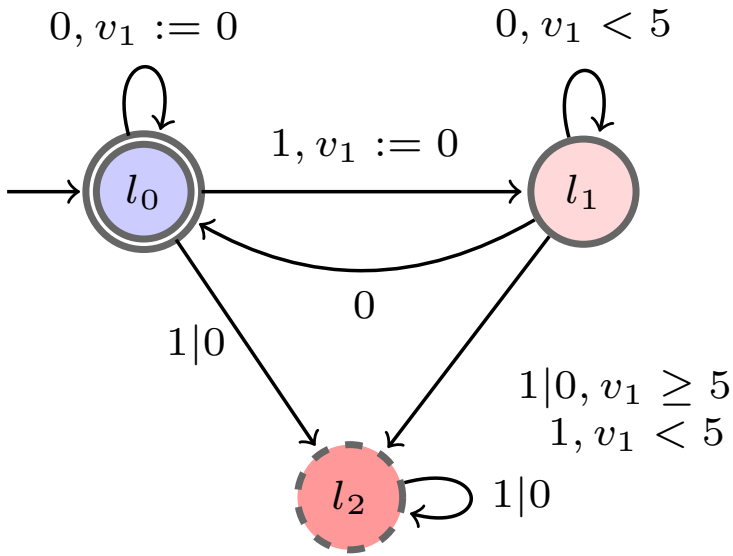
Complete DTA

- For any location $l \in L$ and any event $a \in \Sigma$, the disjunction of the guards of the transitions leaving l and labelled by a evaluates to *true*.

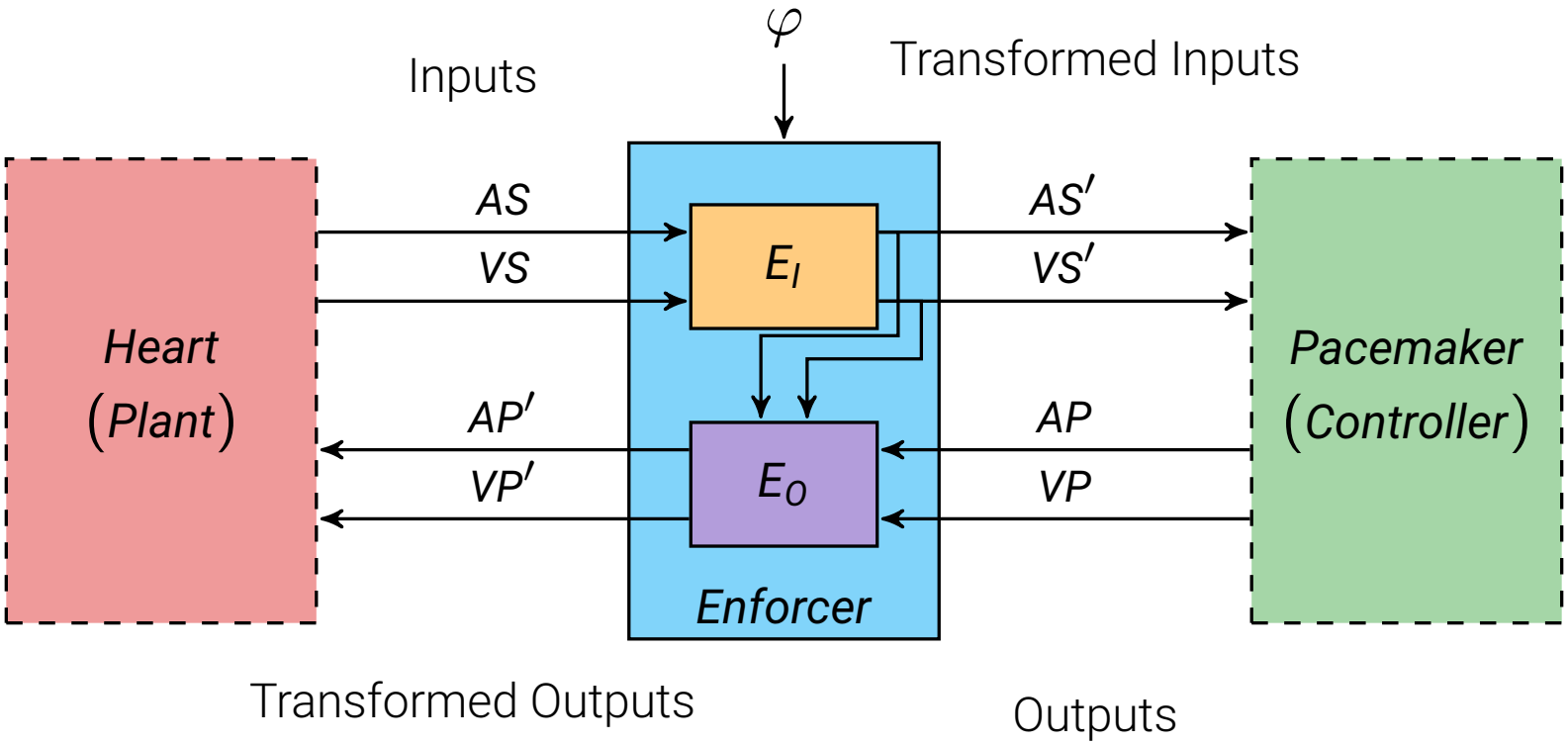
Input DTA – $\mathcal{A}_{\varphi_I}$

- Bidirectional enforcement requires a property solely reliant on inputs
- Achieved by projecting DTA $\mathcal{A}_\varphi$ on inputs
  - All locations and clocks remain
  - All transitions remain, with outputs removed from guards





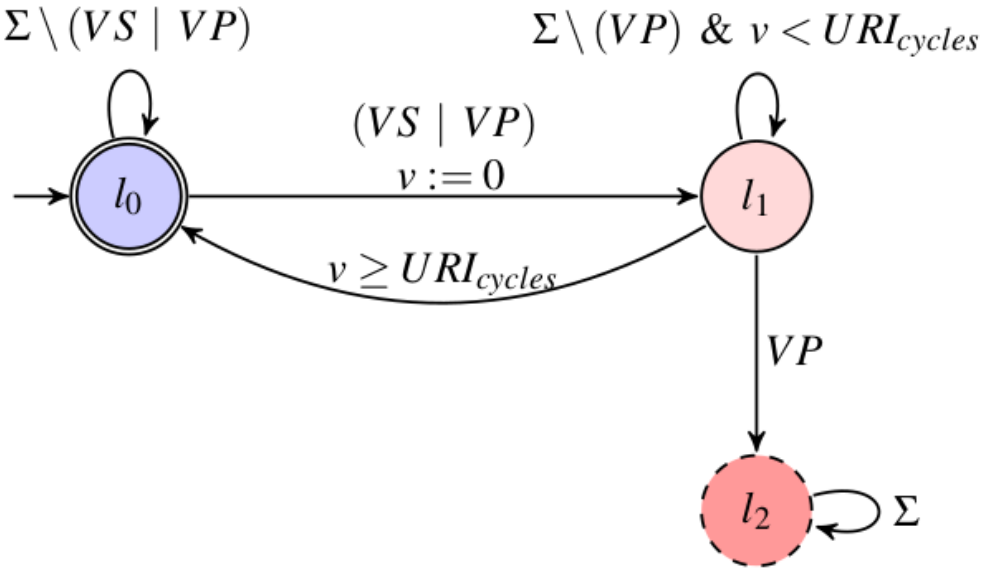
Enforcer Operation



- Enforcers operate **iteratively**
- They first edit inputs (if necessary) & emit, then edit outputs (if necessary) & emit.
- Then, they advance their internal DTA state.

Example Enforcement Trace

Policy P_4: After a ventricular event ($VP \mid VS$), a VP may happen only after URI_{cycles}.



Assume URI_{cycles} = 3

t	0	1	2	3	4	5	6	7	8	9	10
VS	1										
VS'	1										
VP					1		1		1		1
VP'					1				1		

Edit Functions

Input Edit — $\textbf{editI}_{\varphi_I}(\sigma_I)$

- A set of possible next input events of an Input Word (σ_I)
- Such that the word can still be extended to satisfy the property φ_I

Output Edit — $\textbf{editO}_{\varphi}(\sigma, x)$

- A set of possible next output events of an Input-Output Word (σ, x)
- Such that the word can still be extended to satisfy the property φ

Variants

- Random Edit — $\textbf{rand-editI}_{\varphi_I}(\sigma_I)$ and $\textbf{rand-editO}_{\varphi}(\sigma, x)$
  - Randomly selects an element from the respective edit function
- Minimum Distance Edit — $\textbf{minD-editI}_{\varphi_I}(\sigma_I, x)$ and $\textbf{minD-editO}_{\varphi}(\sigma, x, y)$
  - Selects an element from the respective edit function with minimum distance from the current value

Why hardware?

The power of an enforcer

- Runtime enforcers are **omnipotent** — they can edit any I/O
- Potentially catastrophic if an enforcer is faulty or could be compromised

The consistency (security) of hardware

- Software can be altered/updated
- Hardware can be built to be permanent using ASICs/discrete components
- Software is intrinsically difficult to analyse
  - May require analysis of entire appication/runtime/RTOS **(program could halt!)**
  - Requires processor model
- Hardware can more easily be checked for timing/functional properties
  - Requires analysis of just enforcer hardware module

Statically compiling synthesis algorithm

1: $t \leftarrow 0$
2: $q \leftarrow q_0$
3: **while** true **do**
4: $x_t \leftarrow$ read_in_chan()
5: **if** $\exists \sigma'_I \in \Sigma_I^* : q \xrightarrow{x_t \cdot \sigma'_I}_I q' \wedge q' \in Q_F$ **then**
6: $x'_t \leftarrow x_t$
7: **else**
8: $x'_t \leftarrow$ rand-editI$_{\mathcal{A}_{\varphi_I}}(q)$
9: **end if**
10: release(x'_t)
11: $y_t \leftarrow$ read_out_chan()
12: **if** $\exists \sigma' \in \Sigma^* : q \xrightarrow{(x'_t, y_t) \cdot \sigma'} q' \wedge q' \in Q_F$ **then**
13: $y'_t \leftarrow y_t$
14: **else**
15: $y'_t \leftarrow$ rand-editO$_{\mathcal{A}_\varphi}(q, x'_t)$
16: **end if**
17: release(y'_t)
18: $q \leftarrow q''$ where $q \xrightarrow{(x'_t, y'_t)} q'' \wedge q'' \notin q_v$
19: $t \leftarrow t + 1$
20: **end while**

LUT

LUT

Multiplexer to edit inputs if non-accepting

Replacement values decided statically

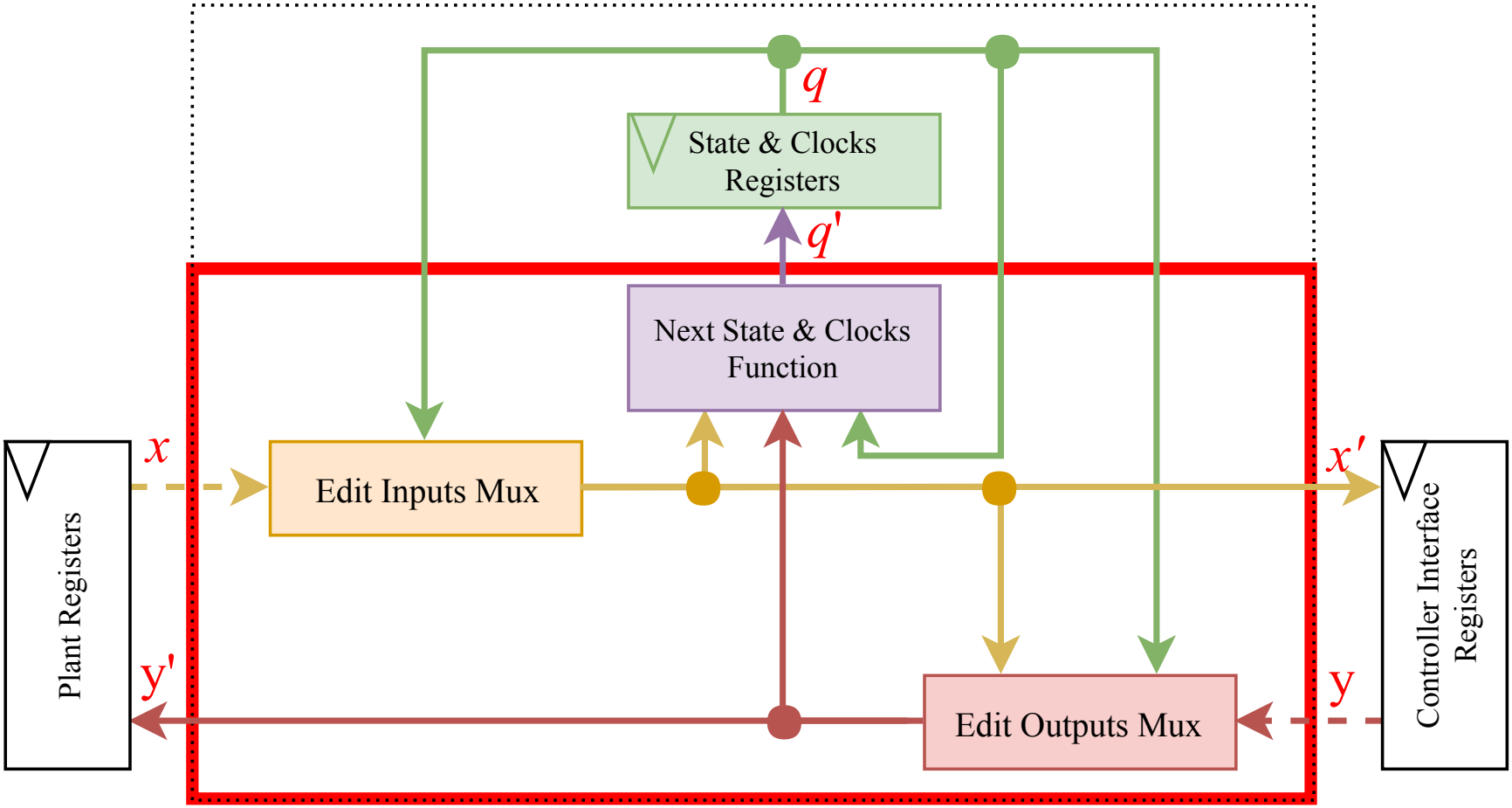
Multiplexer to edit outputs if non-accepting

Replacement values decided statically

FSM + clocks

Constraint

Generalised architecture



Constraint: Next State $q' \notin q_v$

Verifying the hardware

Functional correctness using EBMC [20]

- Security vulnerabilities can be present in **implementations** of otherwise-correct systems (e.g. Heartbleed [21])
- EBMC is a model checker for hardware designs.
  - It functions over *assertions* in Verilog Code.
- We can assert $\forall q \in Q, \forall (x, y) \in \Sigma, E(q, x, y) \rightarrow (q', x', y')$ such that $q \notin q_v$.
  - i.e. EBMC will check the combinatorial update implementation for the possibility of any input at any time that could cause a transition to a violation state.
- We use k-induction with k=1.
- As it is analysing a combinatorial function, the analysis is very quick.

---

[20]. University of Oxford. (Jan. 2019), EBMC, [Online]. Available: `http://www.cprover.org/ebmc`

[21]. S. Inc. (2017), Heartbleed bug,

Verifying the hardware

Timing correctness using Quartus TimeQuest [22]

- Quartus TimeQuest will determine the **critical path** and **max** f_{clk} of the system.
- As there are two registers for signals to propagate through, the overhead is $\frac{1}{f_{clk}} \times 2$

Power consumption using Quartus PowerPlay [23]

- Assume f_{clk} = 100kHz = 10μS, so overhead = 20μS.
- I/O toggle rate set at average of 1.5 transitions/S (avg. 90 bpm).
- Vectorless estimation for internal signals (more pessimistic).

---

[22]. *TimeQuest timing analyzer: Quick start tutorial*, UG-TMQSTANZR-1.1, Altera, Dec. 2009

[23]. *PowerPlay early power estimator user guide*, UG-01070, Intel, Feb. 2017

Safety and Security Analysis

Additional hardware risk assessment

- Failure rate of system is not $\mathit{fail}(\mathit{enforcer}) \times \mathit{fail}(\mathit{pacemaker})$.
- Enforcer encapsulates original controller and will take over in failure scenario.
- Failure rate of system is just $\mathit{fail}(\mathit{enforcer})$.

Attacker modelling

- Policies P_1 through P_5 effectively mitigate attack scenarios
  - Attacker switches off pacing? (P_2, P_3, P_5)
  - Attacker reprograms pacemaker to pace too fast? (P_4)
  - Attacker reprograms pacemaker to pace *AP* and *VP* simultaneously? (P_1)
- EBMC validates that all attack traces are mitigated for safe minimum QoS.

Hardware Synthesis Results

Experimental Methodology

- Policies provided for P_1 through P_5
- Enforcer Verilog synthesized with Intel Quartus 16.0 to Max V CPLD
- EBMC verifies enforcer constraint
- Quartus TimeQuest provides information
- Quartus PowerPlay can estimate CPLD power consumption

Results: HW consumption

Enforcer Policy	States	Timers	Transitions	LEs
P_1	2	0	2	8
P_2	3	1	5	158
P_3	3	1	5	158
P_4	3	1	5	158
P_5	3	1	5	158
$P_{1,2,3,4}$	5	2	13	335
$P_{1,2,3,4,5}$	5	2	19	343
$P_1 \wedge P_2 \wedge P_3 \wedge P_4$	9	3	84	494
$P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5$	17	4	304	761

Analysis

- Increasing complexity (States,Timers,Transitions) $\longrightarrow$ more hardware (LEs)

Results: HW performance

Enforcer Policy	LEs	Verification Time (s)	Min OH (ns)	Dynamic Power (mW @ 100kHz)
P_1	8	<0.01	8.2	0.03
P_2	158	<0.01	99	0.05
P_3	158	<0.01	97	0.05
P_4	158	<0.01	90	0.05
P_5	158	<0.01	120	0.05
$P_{1,2,3,4}$	335	0.06	206	0.07
$P_{1,2,3,4,5}$	343	0.08	206	0.07
$P_1 \wedge P_2 \wedge P_3 \wedge P_4$	494	0.06	204	0.08
$P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5$	761	12.6	-	-

Analysis

- More hardware (LEs) $\longrightarrow$ Larger verification time, larger OH, more power req.
- However, order of magnitude smaller overheads than software-based enforcers

Conclusions

- As IMDs grow in complexity/connectivity they are increasingly vulnerable to attack
- Run-time Enforcement can guarantee untrustworthy applications.
- Existing RE implementations not “secure” (they are usually software)
- Furthermore, implementations of Enforcers can themselves feature mistakes.
- We compile DTA policies to hardware-based enforcers.
- Hardware is intrinsically safer and more secure than complex software.
- The synthesized enforcers are automatically checked for correctness.
- Our enforcers guarantee a minimum safe QoS for IMDs.

Source code access

Source code for this project and its examples are available under the MIT license at *https://github.com/PRETgroup/easy-rte*

References I

[1] H. Pearce, S. Pinisetty, P. S. Roop, M. M. Kuo, and A. Ukil, "Smart i/o modules for mitigating cyber-physical attacks on industrial control systems," *IEEE Transactions on Industrial Informatics*, 2019.

[2] S. Pinisetty, P. S. Roop, S. Smyth, N. Allen, S. Tripakis, and R. V. Hanxleden, "Runtime enforcement of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, p. 178, 2017.

[3] S. Pinisetty, P. S. Roop, S. Smyth, S. Tripakis, and R. v. Hanxleden, "Runtime enforcement of reactive systems using synchronous enforcers," in *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software*, ACM, 2017, pp. 80–89.

[4] S. Pinisetty, P. S. Roop, V. Sawant, and S. Gerardo, "Security of pacemakers using run-time verification," in *Proceedings of the 16th ACM-IEEE International Conference on Formal Methods and Models for System Design*, IEEE, 2018 (October).

References II

[5] T. Zimmerman. (Jun. 2012), VVI pacemaker THWZ, CC 3.0, [Online]. Available: https://commons.wikimedia.org/wiki/File:VVI_Schrittmacher_THWZ.jpg.

[6] L. Pycroft and T. Z. Aziz, "Security of implantable medical devices with wireless connections: The dangers of cyber-attacks," *Expert Review of Medical Devices*, vol. 15, no. 6, pp. 403–406, 2018.

[7] C Weaver, "Patients put at risk by computer viruses," *Wall Street Journal*, 2013.

[8] J Kirk, "Pacemaker hack can deliver deadly 830-volt jolt," *Computerworld*, vol. 17, 2012.

[9] J. D. Rockoff, "J&J warns insulin pump vulnerable to cyber hacking," *Wall Street Journal*, 2016.

References III

[10] E. Marin, D. Singelée, F. D. Garcia, T. Chothia, R. Willems, and B. Preneel, "On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ser. ACSAC '16, Los Angeles, California, USA: ACM, 2016, pp. 226–236. [Online]. Available: http://doi.acm.org/10.1145/2991079.2991094.

[11] U.S. Food and Drug Administration, "Postmarket management of cybersecurity in medical devices," Guidance for Industry, Food, and Drug Administration Staff, Tech. Rep., 2016. [Online]. Available: https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf.

[12] D Takahashi, "Insulin pump hacker says vendor Medtronic is ignoring security risk," *Venturebeat*, 2011. [Online]. Available: https://venturebeat.com/2011/08/25/insulin-pump-hacker-says-vendor-medtronic-is-ignoring-security-risk/.

References IV

[13] D Clery, "Could your pacemaker be hackable?" *Science*, vol. 347, no. 6221, pp. 499–499, 2015.

[14] J. Sametinger, J. Rozenblit, R. Lysecky, and P. Ott, "Security challenges for medical devices," *Commun. ACM*, vol. 58, no. 4, pp. 74–82, Mar. 2015. [Online]. Available: http://doi.acm.org/10.1145/2667218.

[15] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: Non-invasive security for implantable medical devices," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 2–13, Aug. 2011. [Online]. Available: http://doi.acm.org/10.1145/2043164.2018438.

[16] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09, Chicago, Illinois, USA: ACM, 2009, pp. 410–419. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653712.

References V

[17] M. Ivezic. (2017), Cyber-kinetic attacks, [Online]. Available: http://ivezic.com/cyber-kinetic-book/.

[18] J. Hunker, "Cyber war and cyber power," *Issues for NATO doctrine.*, vol. 62, 2010.

[19] S. Pinisetty, P. S. Roop, S. Smyth, N. Allen, S. Tripakis, and R. V. Hanxleden, "Runtime enforcement of cyber-physical systems," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 5s, 178:1–178:25, Sep. 2017. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/3126500.

[20] University of Oxford. (Jan. 2019), EBMC, [Online]. Available: http://www.cprover.org/ebmc.

[21] S. Inc. (2017), Heartbleed bug,

[22] *TimeQuest timing analyzer: Quick start tutorial*, UG-TMQSTANZR-1.1, Altera, Dec. 2009.

[23] *PowerPlay early power estimator user guide*, UG-01070, Intel, Feb. 2017.