



Secure Cyber-Physical Data Monitoring Architecture for Real-Time Fault Detection in U.S. Industrial Automation Systems

Md Shahid Nazir¹

[1]. Master of Science in Business Analytics, Grand Canyon University, Phoenix, AZ, USA;
Email: nazir.dwasa@gmail.com

Doi: [10.63125/exzc2653](https://doi.org/10.63125/exzc2653)

Received: 11 August 2026; Revised: 13 August 2026; Accepted: 10 October 2026; Published: 11 October 2026

Abstract

Plant data can be wrong because a sensor or actuator has failed or because someone has changed what the system sees. This article proposes a secure cyber-physical monitoring architecture that treats the data path as zones and conduits, places detectors where their inputs can be trusted, and combines physics-based residuals, statistical monitors and transport-integrity checks with an independently wired second measurement and a watermarked control input. It is tested on a transparent synthetic two-tank pumped process with PI control, eight instruments, five physical faults and five data-integrity or availability attacks with known onsets, comparing five architecture variants over 72 episodes per event. A baseline SCADA with limit alarms detected almost nothing, because closed-loop control holds the reported variables at their setpoints. Model-free statistics caught abrupt events in about 18 s and gradual ones in 12 to 34 minutes. Physics residuals caught leaks, frozen sensors, pump degradation and stiction in 12 to 104 s and a stealthy sensor ramp in about two minutes, but a slow drift only after 17 minutes. Authenticated telemetry with sequence numbers caught a man-in-the-middle change at once and a replay in about 20 s; the independent sensor cut the drift delay to about two minutes; the watermark gave a replay backstop at about 100 s at a cost in actuator activity. False alarms rose from about 2 to about 7 per day across the variants. Sweeps showed where each check fails: a clock offset of 15 s between the two measurement paths made the cross-source check unusable, packet loss turned a naive receiver into a source of false alarms, and a loss of pump gain confounded the watermark test. The data could not separate a drifting transmitter from a stealthy attack on it, which the attribution layer reports as an ambiguous class. All results describe a stylised testbed with three seeds; the roadmap and the mapping to NIST SP 800-82r3, IEC 62443, CISA, NERC CIP and TSA material are proposals and make no compliance claim.

Keywords

Industrial Control Systems, Cyber-Physical Security, Fault Detection, IEC 62443 Zones and Conduits, Replay Attack, Physical Watermarking, CUSUM, Operational Technology

Introduction

An industrial automation system watches itself through data. Level, flow, pressure and current readings travel from field devices to a programmable logic controller, from the controller to a supervisory system and a historian, and from there to the people and programs that decide whether the plant is healthy. Fault detection, in the sense of the control engineering literature, is the use of that data to notice that something physical has changed: a sensor drifts, a pump wears, a valve sticks, a tank leaks (Isermann, 2005; Frank, 1990; Willsky, 1976). Security monitoring, in the sense of the operational technology (OT) literature, is the use of the same data, and of data about the data, to notice that someone has changed what the system sees or does (Cárdenas et al., 2011; Giraldo et al., 2018; Humayed et al., 2017). The two problems have been studied by largely separate communities, and in a plant they land on the same operator, who sees an alarm and must decide whether to call maintenance or incident response.

The United States context sharpens the question. Guidance for OT security is extensive: the NIST guide to OT security (Stouffer et al., 2023), the Cybersecurity Framework 2.0 (National Institute of Standards and Technology, 2024), the cross-sector performance goals of the Cybersecurity and Infrastructure Security Agency (Cybersecurity and Infrastructure Security Agency, 2023a), and sector rules such as the NERC CIP standards for the bulk electric system (NERC, n.d.) and the Transportation Security Administration's pipeline directives (Transportation Security Administration, 2025), with incident reporting obligations arriving through the Cyber Incident Reporting for Critical Infrastructure Act (U.S. Congress, 2022) and federal attention raised by Executive Order 14028 (Executive Office of the President, 2021). Incidents have shown what the attack classes look like in practice: manipulation of controller logic with falsified process views in Stuxnet (Langner, 2011; Falliere et al., 2011), the targeting of a safety system in TRISIS (Dragos, 2017), an attempted change of a chemical setpoint at a water utility (Cybersecurity and Infrastructure Security Agency, 2021), the exploitation of exposed controllers at water facilities (Cybersecurity and Infrastructure Security Agency, 2023b), and a ransomware event that stopped a pipeline through its business systems (Beerman et al., 2023). Most of this guidance describes what to protect and monitor, and little of it says how a detector built on process data should be placed, calibrated and interpreted so that a fault and an attack are both caught in time and told apart where possible. My reading of the current texts is given in the discussion; they change, and the reader should verify them.

This article proposes a monitoring architecture that treats the data path as an engineered system with zones and conduits, places detectors where the data they consume can be trusted, and combines three kinds of evidence: the physics of the process, the statistics of the telemetry, and the integrity of the transport. It is expressed in engineering notation, a zone-and-conduit diagram, a piping and instrumentation diagram of the testbed, a control block diagram of the detection signal flow, a flowchart of the study, and a stage-gate roadmap, because the audience includes control engineers who read drawings before they read prose. The architecture is tested on a transparent synthetic testbed: a two-tank pumped process with two PI loops, eight instruments, a PLC, a telemetry path with loss and reordering, an independent measurement path, five physical faults and five data-integrity or availability attacks, all with known ground truth. Five architecture variants from a baseline SCADA with limit alarms to the full design are compared on detection delay, misses, false alarms per day, attribution and process impact, and the design is stressed with sweeps over attack size, transport quality, model error, noise and thresholds, with ablations of each security component, and with a Monte Carlo analysis over twelve uncertain factors.

The research questions are:

RQ1. Which architecture components contribute to detecting which classes of event, and at what cost in false alarms, when the baseline is a conventional SCADA with limit alarms?

RQ2. How do attack size and stealth, transport defects (loss, reordering, sampling, clock offset) and model and noise conditions change detection delay and false alarms, and where does each detector family fail?

RQ3. Under what conditions can the monitoring layer tell a physical fault from a data-integrity attack, and where is the distinction impossible from the data alone?

RQ4. How should the architecture be deployed and governed in the U.S. context, and what should an

owner expect from it?

The contribution is a framework and a quantified set of mechanisms, not a product or a measured result from a plant. All numbers describe a stylised testbed. What is intended to transfer is the structure: which checks catch which events, which checks confound one another, and what the honest limits of attribution are.

Background

Cyber-physical systems and the security of control

Cyber-physical systems join computation, networking and physical processes, and their design challenges differ from those of computing alone (Lee, 2008; Rajkumar et al., 2010). Security research on control systems has shown that attacks can be modelled as perturbations of sensing and actuation, that the physics constrains what an attacker can do without being noticed, and that detection should use the plant model as much as the network (Cárdenas et al., 2011; Mo et al., 2012; Pasqualetti et al., 2013; Sandberg et al., 2015). Surveys organise the attack space and the detection methods (Humayed et al., 2017; Giraldo et al., 2018; Ding et al., 2018; Dibaji et al., 2019). Two results from this literature shape the architecture here. First, a replay attack, in which recorded sensor data are played back while the plant is manipulated, defeats any residual test that compares sensors with a model of the same sensors, because the replayed data are physically consistent; a watermark, a small secret perturbation added to the control input, restores detectability at a cost in control performance (Mo & Sinopoli, 2009; Mo et al., 2015; Mo et al., 2014; Satchidanandan & Kumar, 2017). Second, an attacker who knows the model and the detector can inject data that stay below the detection threshold indefinitely, and the damage such an attack can do is bounded by the threshold (Teixeira et al., 2015; Bai et al., 2017; Urbina et al., 2016; Amin et al., 2013). Secure estimation under adversarial sensor corruption is possible only when enough sensors remain honest (Fawzi et al., 2014), which is the formal reason for an independent measurement path.

Fault detection and diagnosis

The fault-detection literature uses analytical redundancy: relations among measured variables that hold when the plant is healthy, so that their violation signals a fault (Willsky, 1976; Frank, 1990; Isermann, 2005). Model-based methods build observers and residuals; process-history methods learn statistical models of normal data, such as principal component analysis with squared prediction error, and have been compared on the Tennessee Eastman benchmark (Venkatasubramanian et al., 2003b; Venkatasubramanian et al., 2003a; Chiang et al., 2001; Qin, 2012; Yin et al., 2012; Downs & Vogel, 1993). The Kalman filter and its innovations are the classical residual generator (Kalman, 1960; Mehra & Peschon, 1971). These tools do not distinguish a fault from an attack; a biased sensor looks the same to a mass balance whether a transmitter has drifted or a wire has been tapped. The architecture here uses the fault-detection residuals as evidence and adds the integrity checks that can, sometimes, tell the two apart.

Change detection and alarm budgets

Residuals are turned into alarms by sequential tests. The CUSUM and EWMA charts have known properties, and the CUSUM is optimal in a minimax sense for detecting a shift in mean (Page, 1954; Roberts, 1959; Lucas & Saccucci, 1990; Lorden, 1971; Moustakides, 1986; Basseville & Nikiforov, 1993). On closed-loop process data the residuals are autocorrelated, and run lengths computed for independent data do not hold, so thresholds here are calibrated empirically to a false-alarm budget. The budget matters because alarms are consumed by people: studies of security operations report overwhelming false-positive rates, and the alarm-management standard for the process industries sets expectations for how many alarms an operator can handle (Alahmadi et al., 2022; Vielberth et al., 2020; International Society of Automation, 2016; Axelsson, 2000).

Industrial control systems: testbeds, datasets and process-aware detection

Research on ICS security has been served by physical testbeds and their datasets: the secure water treatment testbed and its data, the water distribution testbed, and the hardware-in-the-loop HAI dataset, among others, with surveys of testbeds and of attack trends (Mathur & Tippenhauer, 2016; Ahmed et al., 2017; Shin et al., 2020; Morris & Gao, 2014; Conti et al., 2021; Alladi et al., 2020; Makrakis et al., 2021; Goh et al., 2017). Detection methods range from recurrent and convolutional networks on sensor streams (Inoue et al., 2017; Kravchik & Shabtai, 2018) to process-aware and invariant-based

checks that read the PLC's view of the process ([Hadžiosmanović et al., 2014](#); [Lin et al., 2018](#); [Feng et al., 2019](#)), and to the use of sensor and process noise as a fingerprint that authenticates a sensor ([Ahmed et al., 2018](#)). The present study is synthetic rather than physical, which loses realism and gains ground truth for faults and attacks together, something the datasets rarely label.

Networks, protocols, time and architecture

Industrial networks and their protocols were designed for determinism, not authentication, and the surveys of control networks and of ICS security management describe the gap ([Galloway & Hancke, 2013](#); [Knowles et al., 2015](#); [Cherdantseva et al., 2016](#); [Yadav & Paul, 2021](#); [Bhamare et al., 2020](#)). Secure variants of Modbus have been designed ([Nai Fovino et al., 2009](#)), DNP3 includes secure authentication ([IEEE, 2012](#)), and TLS 1.3 is the general tool for authenticated transport ([Rescorla, 2018](#)); time protocols need protection of their own ([Mizrahi, 2014](#); [Mills, 1991](#); [IEEE, 2020](#)). The Purdue reference architecture and the IEC 62443 zone-and-conduit model structure where controls sit ([Williams, 1994](#); [International Electrotechnical Commission, 2020](#); [International Electrotechnical Commission, 2013b](#); [International Electrotechnical Commission, 2023](#); [International Electrotechnical Commission, 2013a](#)); zero-trust principles and resilience engineering add the ideas of continuous verification and of designing for degraded operation ([Rose et al., 2020](#); [Feng & Hu, 2023](#); [Syed et al., 2022](#); [Ross et al., 2021](#); [Linkov et al., 2013](#)). Edge computing and digital twins place analytics close to the process ([Shi et al., 2016b](#); [Eckhart & Ekelhart, 2019](#); [Tao et al., 2019](#)), and the machine-learning literature on intrusion detection warns about its own evaluation pitfalls ([Sommer & Paxson, 2010](#); [Buczak & Guven, 2016](#); [Ferrag et al., 2020](#)). Safety instrumented systems are a separate layer governed by functional-safety standards ([International Electrotechnical Commission, 2016](#)); the architecture here monitors but does not replace them.

Evaluating time-series detectors

How detection is scored changes what is learned. Real-time anomaly benchmarks reward early detection ([Lavin & Ahmad, 2015](#)), time-series precision and recall need their own definitions ([Tatbul et al., 2018](#)), and several widely used anomaly benchmarks have been shown to be flawed ([Wu & Keogh, 2023](#)). This study scores detection delay from a known onset, misses within fixed horizons, false alarms per day on held-out normal data, and attribution, and reports them together.

Summary

Table 1 lists the strands and the gap each leaves. No strand places physics-based, statistical and integrity checks in one architecture, tests them on the same events with ground truth for both faults and attacks, and reports where attribution fails.

Requirements and architecture

Requirements

Six requirements follow from the literature and from practice (Table 2). They concern where monitors sit, what the conduit and the content checks each cover, the alarm budget, attribution, graceful degradation, and governance.

Table 1: Literature strands, what they establish, representative sources and the gap addressed

Strand	What the literature establishes	Representative sources	Gap addressed here
Cyber-physical systems and their security	CPS couple computation and physics; attacks can be modelled as perturbations of sensing and actuation, and the physics itself gives detection power	Lee (2008); Rajkumar et al. (2010); Cárdenas et al. (2011); Mo et al. (2012); Humayed et al. (2017); Giraldo et al. (2018)	An architecture that places physics-based, statistical and integrity checks in specific zones and compares them on one testbed
Secure control, replay and watermarking	Replay and false-data attacks can be undetectable to residual tests; watermarked inputs and secure estimators restore detectability at a control cost	Mo & Sinopoli (2009); Mo et al. (2014, 2015); Satchidanandan & Kumar (2017); Teixeira et al. (2015); Fawzi et al. (2014); Bai et al. (2017)	Watermark delay and cost quantified against an independent sensor and sequence checks
Fault detection and diagnosis	Analytical redundancy, observers and change detectors detect and isolate faults; data-driven monitoring (PCA, SPE) complements models	Willsky (1976); Frank (1990); Isermann (2005); Venkatasubramanian et al. (2003a, 2003b); Chiang et al. (2001); Qin (2012)	Same residuals used for faults and for attacks, with the ambiguity made explicit
Change detection and control charts	CUSUM and EWMA schemes have known run-length properties; thresholds trade delay against false alarms	Page (1954); Roberts (1959); Lucas & Saccucci (1990); Lorden (1971); Moustakides (1986); Basseville & Nikiforov (1993)	Thresholds calibrated to an alarm budget on closed-loop data, where autocorrelation inflates run lengths
ICS security: surveys, testbeds, datasets	Water and process testbeds (SWaT, WADI, HAI) and datasets support detection research; surveys catalogue attacks and countermeasures	Mathur & Tippenhauer (2016); Ahmed et al. (2017); Shin et al. (2020); Morris & Gao (2014); Conti et al. (2021); Alladi et al. (2020); Makrakis et al. (2021)	Ground truth for both faults and attacks, which real datasets rarely label together
Process-aware and invariant-based detection	Semantic and invariant checks on process variables detect attacks that network monitoring misses; noise fingerprints authenticate sensors	Hadžiosmanović et al. (2014); Lin et al. (2018); Feng et al. (2019); Ahmed et al. (2018); Inoue et al. (2017); Kravchik & Shabtai (2018)	Cross-source consistency with an independently wired sensor as an invariant
Networks, protocols and time	ICS protocols lack authentication by default; secure variants and TLS add it; time protocols need their own protection	Galloway & Hancke (2013); Nai Fovino et al. (2009); IEEE (2012); Rescorla (2018); Mizrahi (2014); Mills (1991); IEEE (2020)	Loss, reordering, sampling and clock offset modelled as factors acting on detection
Architecture and zoning	Purdue levels and IEC 62443 zones and conduits structure where controls and monitors sit; zero trust and resilience engineering add principles	Williams (1994); International Electrotechnical Commission (2013b, 2020, 2023); Rose et al. (2020); Ross et al. (2021); Feng & Hu (2023)	Detector placement and the independent path expressed in zone-and-conduit terms
U.S. guidance, regulation and incidents	NIST SP 800-82r3, CSF 2.0, CISA goals and advisories, NERC CIP, TSA directives and incident reporting law frame expectations; incidents show the attack classes	Stouffer et al. (2023); National Institute of Standards and Technology (2024); Cybersecurity and Infrastructure Security Agency (2021, 2023a, 2023b); NERC (n.d.); Transportation Security Administration (2025); U.S. Congress (2022); Langner (2011); Dragos (2017); Beerman et al. (2023)	Mapping of architecture components to the guidance, without compliance claims
Evaluating detectors and alarms	Base rates, benchmark flaws and time-series scoring shape what detection numbers mean; alarm management limits what operators can absorb	Axelsson (2000); Sommer & Paxson (2010); Lavin & Ahmad (2015); Tatbul et al. (2018); Wu & Keogh (2023); Alahmadi et al. (2022); International Society of Automation (2016)	Delay, miss rate, false alarms per day and attribution reported together, with a stated alarm budget

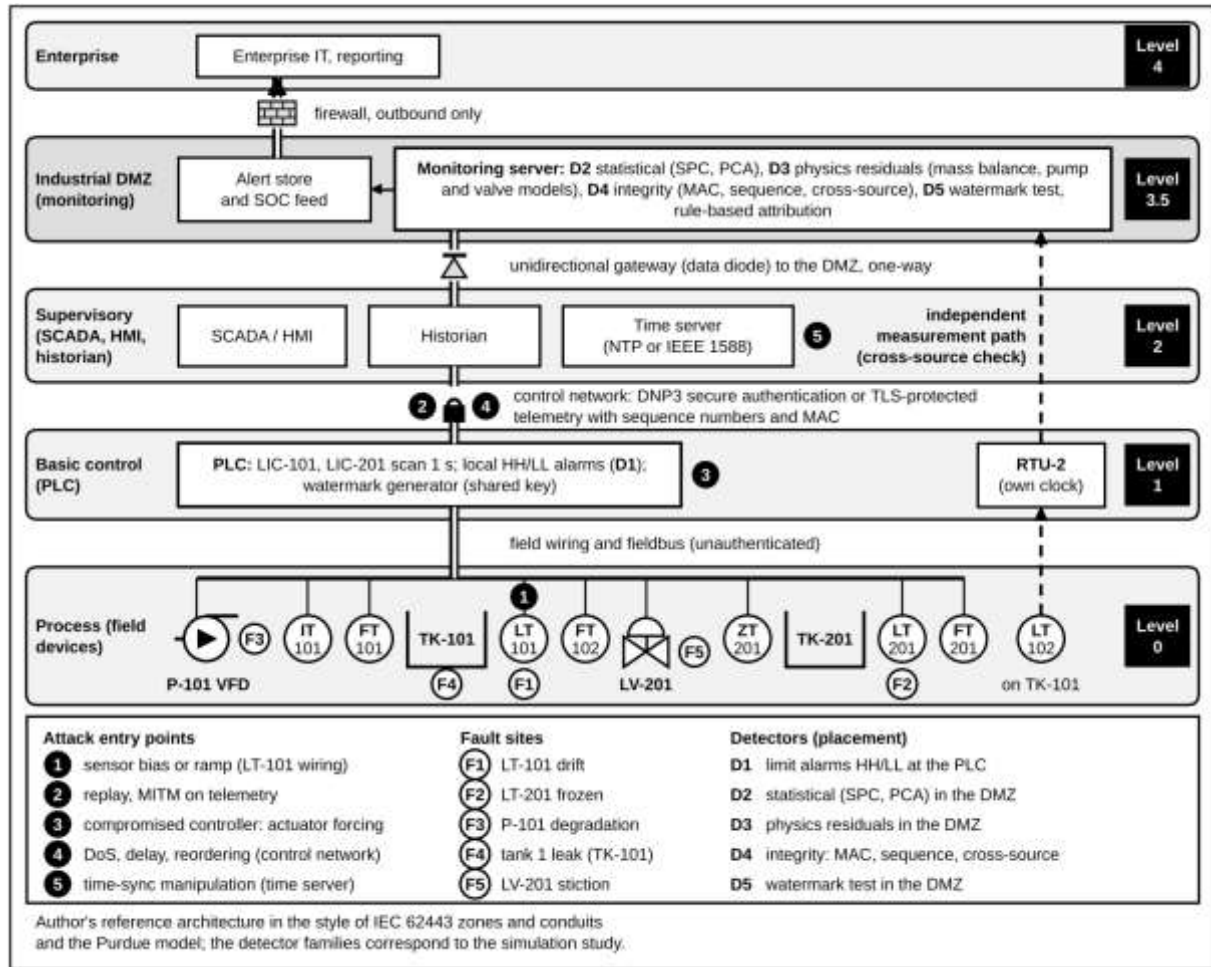
Table 2: Requirements for a secure monitoring architecture: rationale, operationalisation and where each is evidenced

Requirement	Why it matters	How it is operationalised here	Evidence in the testbed
R1 Place monitors where the data are trustworthy	A monitor that reads only the telemetry stream inherits every attack on that stream	Zone-and-conduit architecture; monitoring in the DMZ behind a unidirectional gateway; an independent measurement path	Figures 1 and 2, Table 5
R2 Protect the conduit, then check the content	Authentication stops forged packets but not a lying sensor or a compromised controller	MAC, sequence numbers and staleness checks (D4); physics residuals (D3) for what authentication cannot see	Tables 7 and 13, Figure 9d
R3 Detect in real time with a stated alarm budget	Operators act on alarms; too many are ignored	Thresholds calibrated on normal data to a target false-alarm rate; delay reported against the budget	Tables 10 and 11, Figure 9c
R4 Distinguish faults from attacks where the evidence allows	Response differs: maintenance versus incident response	Rule-based attribution from the detector pattern; explicit ambiguous class for gradual sensor integrity events	Table 14, Figure 6d
R5 Degrade gracefully under transport and model defects	Loss, reordering, clock offset and model error are normal in plants	Sequence-aware receiver; suppression of physics alarms while the stream is degraded; recalibration after changes	Tables 10 and 11, Figures 8 and 9
R6 Map to U.S. guidance and keep evidence	Owners must show what they monitor and why	Component-to-guidance mapping; roadmap with gates	Table 15, Figure 12

Zones, conduits and detector placement

Figure 1 shows the reference architecture. The process and its instruments sit at Level 0, the PLC at Level 1 with its control loops, local limit alarms and, in the full design, a watermark generator. Supervisory systems and the historian sit at Level 2. The monitoring server sits in an industrial DMZ at Level 3.5, fed through a unidirectional gateway, so that nothing in the monitoring layer can write back to control. The control-network conduit carries telemetry with sequence numbers and a message authentication code (or TLS), and the time source is treated as a protected asset. An independent measurement path carries a second level transmitter on the critical tank through its own remote terminal unit and its own conduit to the monitoring server, bypassing the PLC and the supervisory layer. Attack entry points (sensor wiring, the control network, the controller, the time source) and fault sites are marked, and the five detector families are placed where their inputs are available. The reference architecture is the author's synthesis in the style of the zone-and-conduit model and the Purdue levels; it makes no claim of compliance with any standard.

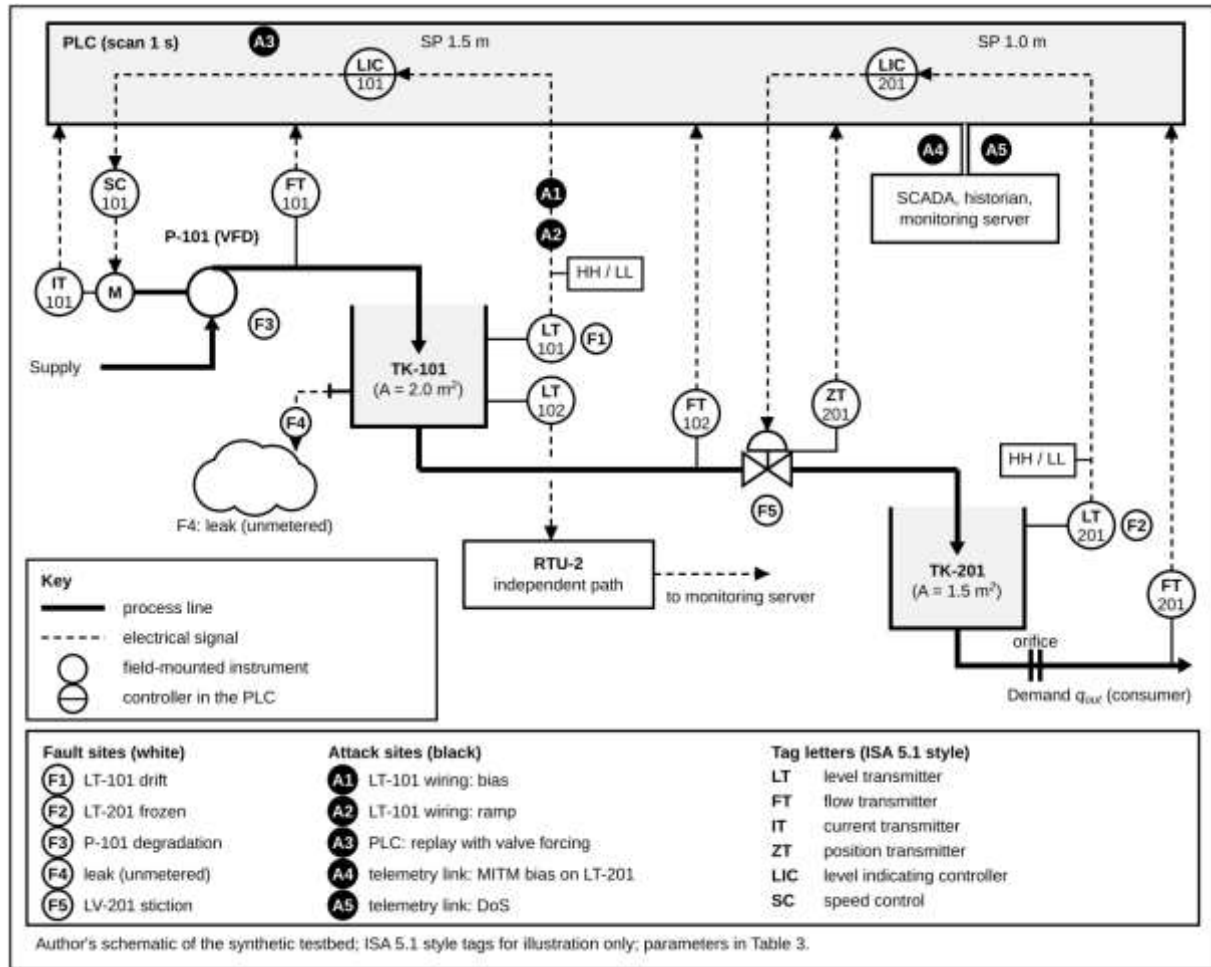
Figure 1: Secure monitoring reference architecture: zones and conduits from the field to the enterprise, authenticated telemetry, unidirectional gateway to the monitoring DMZ, independent measurement path, attack entry points, fault sites and detector placement



The testbed process

Figure 2 is the piping and instrumentation diagram of the testbed. A variable-speed pump P-101 fills tank TK-101; a control valve LV-201 drains it into tank TK-201, which supplies a consumer whose demand varies. Level controllers LIC-101 and LIC-201 in the PLC hold the two levels at their setpoints by the pump speed and the valve opening. Eight instruments are modelled: two level transmitters on tank 1 (LT-101 to the PLC, LT-102 on the independent path), one on tank 2, three flow transmitters, the pump motor current and the valve position. The fault sites (sensor drift, frozen sensor, pump degradation, leak, valve stiction) and the attack sites (sensor wiring, controller, telemetry link) are marked. The tags follow the ISA 5.1 style for illustration; the parameters are in Table 3 of the methods section.

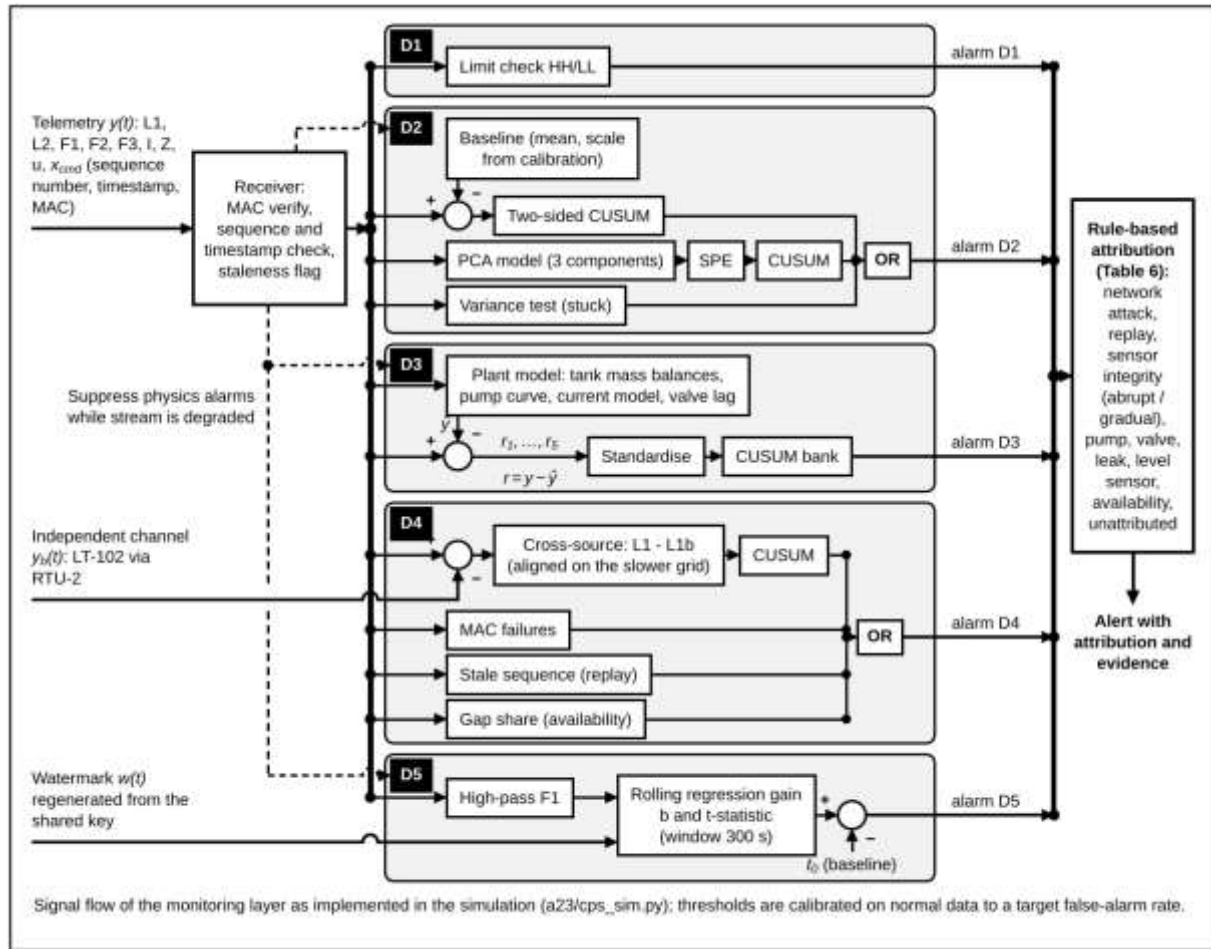
Figure 2: Piping and instrumentation diagram of the two-tank pumped process tested with ISA 5.1 style instrument bubbles, the two level loops, the independent LT-102 path, fault sites F1 to F5 and attack sites A1 to A5



Detection signal flow

Figure 3 shows the monitoring layer as a control block diagram. A receiver verifies the message authentication code, checks sequence numbers and timestamps and flags stale samples. Five detector lanes follow. D1 is the limit check. D2 is statistical: per-channel CUSUMs against a calibration baseline, a PCA model with its squared prediction error, and a variance test for stuck sensors. D3 is physics: residuals from the two tank mass balances, the pump curve, the motor-current model and the valve lag, each standardised and fed to a CUSUM bank. D4 is integrity: authentication failures, stale sequence numbers, the share of missing packets, and a cross-source consistency check between LT-101 and the independent LT-102. D5 is the watermark test: a rolling regression of the high-passed pump-flow reading on the watermark that the monitor regenerates from the shared key. A rule layer combines the alarms into an attribution with evidence, and a feedback path suppresses physics and statistical alarms while the receiver reports a degraded stream.

Figure 3: Detection signal flow in the monitoring layer: receiver checks, five detector lanes with residual generation and CUSUM tests, the watermark regression, suppression under a degraded stream, and rule-based attribution

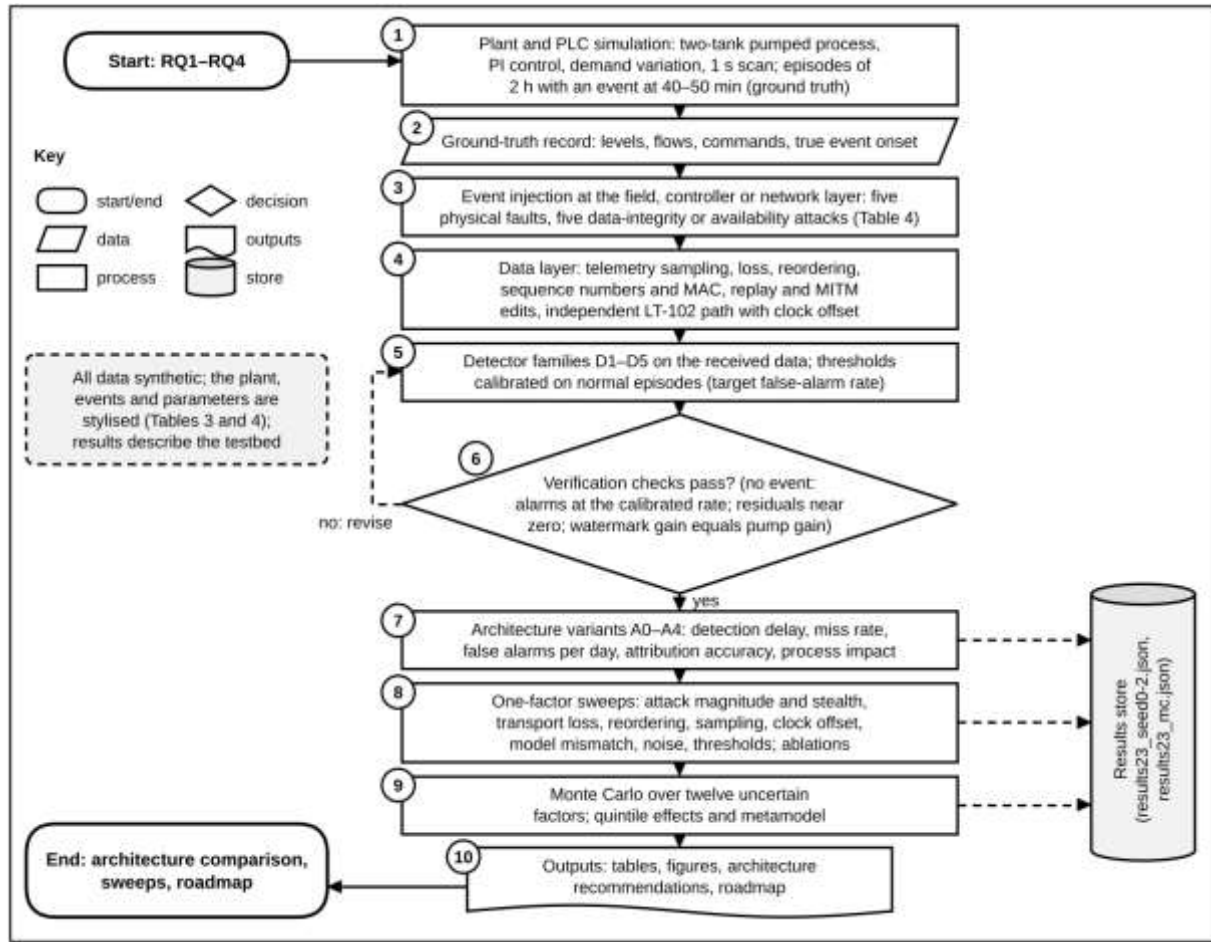


Methods

Study design

Figure 4 shows the study as a flowchart. A plant and PLC simulation produces the ground truth for episodes of two hours in which an event begins between 40 and 50 minutes. Events are injected at the field, controller or network layer. A data layer produces what the monitoring server receives: sampled telemetry with loss and reordering, sequence numbers and authentication flags, replay or man-in-the-middle edits, and the independent channel with its own clock. The detector families run on the received data with thresholds calibrated on normal episodes. Verification checks decide whether the pipeline is trusted. The five architecture variants are then compared, one-factor sweeps and ablations trace each factor, and a Monte Carlo analysis varies twelve factors jointly.

Figure 4: Study design: plant and PLC simulation, event injection, data layer, detectors and calibration, verification, architecture comparison, sweeps and ablations, Monte Carlo and outputs, drawn with standard flowchart symbols



All data are synthetic. No plant data, vendor data or confidential information were used. The parameters are stylised values chosen so that the process is controllable, the faults are plausible and the attacks are of the kinds documented in the literature and in incidents; they are not measurements of any real facility.

Plant, controllers and instruments

Table 3 gives the testbed. Tank 1 (area 2.0 m²) is filled by a variable-speed pump whose flow is proportional to speed and drained through a control valve into tank 2 (area 1.5 m²), which supplies a demand that varies as an Ornstein-Uhlenbeck process with a standard deviation of 15% and a correlation time of ten minutes. Two PI controllers in the PLC, scanned every second, hold the levels at 1.5 m and 1.0 m by the pump speed and the valve opening; the valve actuator has a five-second lag. The pump motor current is a linear function of flow. Sensor noise is white with the standard deviations in the table. Each episode is 7,200 s long; the first 1,200 s are a warm-up that is excluded from alarm counts; the event onset is uniform between 2,400 and 3,000 s, leaving at least 70 minutes of post-onset data.

Table 3: Testbed plant, controllers, instruments, episodes and calibration

Element	Model or value	Notes
Tank 1 (TK-101)	Area 2.0 m ² , level setpoint 1.5 m, alarm limits 0.3 and 2.6 m, physical limits 0 and 3.0 m	Filled by pump P-101, drained through LV-201
Tank 2 (TK-201)	Area 1.5 m ² , setpoint 1.0 m, alarm limits 0.2 and 1.7 m, physical limits 0 and 2.0 m	Supplies a varying demand
Pump P-101 (VFD)	$q_p = 0.020 u (1 - \text{deg}) \text{ m}^3/\text{s}$, speed u in $[0, 1]$; motor current $I = 2.0 + 400 q_p \text{ A}$	deg = degradation (fault F3)
Control valve LV-201	$q_{12} = 0.01633 \times \sqrt{h_1} \text{ m}^3/\text{s}$; first-order actuator, 5 s; stiction deadband in fault F5	Position feedback ZT-201
Demand	$q_{\text{out}} = 0.012 d(t) \sqrt{h_2}$; $d(t)$ Ornstein-Uhlenbeck around 1, s.d. 0.15, correlation time 600 s	Nominal flow 0.012 m ³ /s
Controllers (PLC, 1 s scan)	LIC-101: PI on h_1 to pump speed, gain 1.0 per m, reset 300 s; LIC-201: PI on h_2 to valve, gain 0.75, reset 300 s; anti-windup	Watermark added to the pump speed in A4
Sensors and noise (1 s.d.)	LT-101, LT-201 0.005 m; LT-102 0.008 m (independent path, 5 s); FT-101, FT-102, FT-201 0.25 L/s; IT-101 0.05 A; ZT-201 0.005	Gaussian white noise
Episodes	7,200 s each; event onset uniform in 2,400 to 3,000 s; first 1,200 s used only for warm-up	24 episodes per event and seed in the architecture run, 16 in sweeps
Calibration	30 normal episodes per seed learn baselines, the PCA model and thresholds at 0.5 false alarms per day per family	Separate normal episodes measure false alarms
Normal operation (seed 0 trace)	h_1 s.d. 0.044 m, h_2 s.d. 0.060 m, pump speed s.d. 0.081	From wmcost23.py

Events

Table 4 lists the ten events. The five physical faults are a slow drift of the tank 1 level transmitter, a frozen tank 2 level transmitter (the reading holds its value while the noise continues), a loss of pump capacity of 35% over 30 minutes, an unmetered leak from tank 1, and valve stiction with a deadband. The five attacks are a bias step on the tank 1 level signal before it reaches the PLC, a slow ramp of the same signal (the stealthy version, physically identical to the drift fault with the opposite sign), a replay of the twenty minutes of telemetry recorded before the onset while the valve is forced closed from a compromised controller, a man-in-the-middle change of the tank 2 level value on the telemetry link with the process untouched, and a denial of service on the telemetry link with 60% loss and ten seconds of jitter. The replay is spliced at the recorded sample closest to the current values of all channels, and a variant blends the first sixty seconds so that the seam is hidden; replayed packets carry their original sequence numbers unless stated. A separate factor sets whether the attacker holds the authentication key. These are fixed-form attacks; the attacker does not adapt to the detectors, which is a limitation taken up in the discussion.

Table 4: Event catalogue: five physical faults and five attacks, the layer at which each enters, its mechanism in the testbed, default size and sweep range

Event	Layer	Mechanism in the testbed	Default	Sweep
F1 LT-101 drift	Field sensor	Bias on the tank 1 level reading ramps up; the loop lowers the true level	0.3 m/h	0.1 to 1.0 m/h
F2 LT-201 frozen	Field sensor	Reading holds the value at onset (noise continues); the valve loop winds up	-	-
F3 Pump degradation	Actuator	Flow per unit speed falls by 35% over 30 min; current model unchanged	35%	-
F4 Tank 1 leak	Process	Unmetered outflow proportional to $\sqrt{h_1}$	4 L/s at 1.5 m	1 to 8 L/s
F5 Valve stiction	Actuator	Valve moves only when the command error exceeds a deadband, then slips	0.08	-
A1 Sensor bias step	Field wiring (before PLC)	LT-101 reads low by a constant; the loop overfills tank 1	-0.4 m	0.05 to 0.8 m
A2 Stealthy ramp	Field wiring	LT-101 bias ramps down slowly; physically identical to F1 with opposite sign	-0.5 m/h	0.1 to 2 m/h
A3 Replay with forcing	Controller and control network	Telemetry replays the 20 min before onset (spliced at the closest recorded sample, optional 60 s blend); the valve is forced	Valve 0.05	No forcing, 0.05 to 0.35; blended seam
A4 Telemetry MITM	Control network	LT-201 value to SCADA raised by a constant; the process is untouched; packets fail the MAC unless the attacker has the key	+0.5 m	Key ablation
A5 DoS on telemetry	Control network	60% packet loss and 10 s jitter from onset; control is local and unaffected	-	-
Transport and model factors	Network, monitor	Loss, jitter, sampling interval, clock offset of the independent path, model mismatch, noise, threshold multiplier	0, 0, 1 s, 0, 0, 1, 1	Tables 10 and 11 ranges

Data layer

Telemetry is sampled at one second (varied in a sweep) and carries the eight sensor values, the commands in effect when the samples were taken, a sequence number and, in the integrity-enabled variants, an authentication flag. Packet loss removes samples; a naive receiver holds the last value silently, a sequence-aware receiver knows from the sequence numbers that a sample is missing, interpolates across the gap for display and does not update its detectors on the missing samples; when more than 35% of the packets in the last five minutes are missing it declares the stream degraded, suppresses the physics, statistical and watermark alarms and raises an availability alert. Jitter is modelled as the swapping of adjacent samples with a probability proportional to the jitter divided by twice the sampling interval; a sequence-aware receiver undoes the swap. Latency adds to the detection delay directly and is not modelled further. The independent transmitter LT-102 is sampled every five seconds through its own path, and a clock offset between the two paths is a factor; the monitor compares LT-101 with the independent sample whose timestamp is nearest, on the slower grid. The watermark sequence is known to the monitor from the shared key and is not taken from the telemetry.

Architecture variants and detectors

Table 5 defines the variants. A0 has limit alarms only. A1 adds statistical monitoring without a plant model. A2 adds the physics residuals. A3 adds the integrity layer (authenticated telemetry with sequence numbers, the unidirectional gateway, the independent path and a sequence-aware receiver). A4 adds the watermark test and the attribution rules. Table 6 defines the statistics and the rules. All CUSUMs use a reference value of 0.5 standard deviations, two-sided. The D2 family standardises each

channel by a baseline mean and scale learned from the calibration episodes and uses a PCA model of nine channels with three components; the D3 family uses a monitoring-layer model whose parameters may differ from the plant's by a relative error (the mismatch factor); the cross-source statistic is the CUSUM of the difference between LT-101 and LT-102; the watermark statistic is the drop of the t-statistic of the regression gain from its calibration value, with the residual scale fixed at calibration so that transients do not inflate it.

Table 5: Architecture variants A0 to A4: conduit protection, data available to the monitor, detector families and receiver behaviour

Variant	Conduit protection	Data available to the monitor	Detector families	Receiver behaviour
A0 Baseline SCADA	None	Telemetry at 1 s	D1 limit alarms (HH/LL)	Naive: holds last value on loss, accepts reordered packets
A1 + statistical monitoring	None	Telemetry	D1, D2 (per-channel CUSUM on baseline, PCA squared prediction error, stuck test)	Naive
A2 + physics residuals	None	Telemetry and a plant model	D1, D2, D3 (mass balances, pump curve, current model, valve lag)	Naive
A3 + integrity layer	MAC and sequence numbers on telemetry; unidirectional gateway to the DMZ; time audit	Telemetry, plant model, independent LT-102 via RTU-2	D1, D2, D3, D4 (MAC failures, stale sequence, gap share, cross-source consistency)	Sequence-aware: interpolates across known gaps, does not update detectors on missing samples, suppresses physics alarms when the gap share exceeds 35%, raises an availability alert
A4 + watermark and attribution	As A3 plus a shared key for the watermark	As A3 plus the regenerated watermark	D1 to D4, D5 (watermark regression test), rule-based attribution	As A3

Table 6: Detector statistics, thresholds and the attribution rules of the full architecture

Detector or rule	Statistic	Threshold	Intended coverage
D1 Limit alarms	Level readings against HH and LL limits	Fixed limits	Gross excursions visible in the data as received
D2 Statistical	Two-sided CUSUM ($k = 0.5$) of each channel's standardised value against its calibration baseline; CUSUM of the PCA squared prediction error (3 components, 9 channels); variance-collapse test on levels	Calibrated	Shifts and relation breaks without a model
D3 Physics residuals	r1, r2 tank mass balances over 30 s windows; r3 flow versus pump curve; r4 current versus flow; r5 valve position versus lagged command; each standardised and fed to a CUSUM	Calibrated	Leaks, actuator faults, sensor faults and attacks that break the physics
D4 Integrity	MAC failure share over 10 s; stale sequence share (sequence older than the running maximum by more than 10); gap share over 300 s; CUSUM of LT-101 minus LT-102 on the 5 s grid	0.5; 5 of 10; 0.35; calibrated	Forged packets, replay, availability, sensor-path disagreement
D5 Watermark	Rolling regression (300 s) of the high-passed FT-101 reading on the regenerated watermark; t-statistic of the gain compared with its calibration value	Calibrated	Replay or substitution of the flow telemetry
Rule 1	MAC failures present	–	Network integrity attack
Rule 2	Gap share above 0.35 (no MAC failures)	–	Telemetry availability (DoS or link failure)
Rule 3	Stale sequence numbers	–	Replay attack
Rule 4	Pump-curve residual without current residual or cross-source disagreement	–	Pump fault
Rule 5	Cross-source disagreement with an abrupt step in LT-101	–	Sensor integrity: abrupt (attack likely)
Rule 6	Watermark test fires (pump not saturated) with cross-source disagreement or without pump residual	–	Replay attack
Rule 7	Cross-source disagreement, gradual	–	Sensor integrity: gradual (drift or stealthy attack): ambiguous
Rules 8 to 10	Valve residual; tank 1 balance; tank 2 balance or stuck test	–	Valve fault; leak; level sensor fault (tank 2)
Default	Any other alarm pattern	–	Unattributed anomaly

The attribution rules were written from the expected signatures of the simulated events; their order gives precedence to conduit evidence (authentication failures, stale sequence numbers, missing packets), then to the pump residual (so that a loss of actuator gain, which also weakens the watermark response, is not called a replay), then to the cross-source and watermark evidence, and finally to the physics residuals. One label is deliberately ambiguous: a gradual disagreement between the two tank

1 transmitters is reported as "sensor integrity: gradual (drift or stealthy attack)", because the data cannot say which it is.

Calibration and metrics

For each seed, thirty normal episodes (60 hours) are used to learn the baselines and the PCA model and to set each family's threshold so that it produces about 0.5 false alarms per day on those episodes, by a binary search on the alarm count with the alarm logic in place (an alarm fires when the statistic crosses the threshold and re-arms when it falls below half of it). Thresholds for limits, authentication failures and stale sequence numbers are fixed. Measured false alarms are counted on separate normal episodes, after the warm-up. Detection delay is the time from the event onset to the first alarm of the architecture (the OR of its enabled families), plus the transport latency; an event not alarmed before the episode ends is a miss, and the tables mark medians computed on fewer than half of the episodes. Attribution is scored at the first alarm from the detector pattern in the following five minutes. Process impact is the largest deviation of a true level from its setpoint after onset, the share of episodes in which a true level reaches an alarm limit and the time at which it does, and the detection margin, the time between detection and the limit. In sweeps that change the noise, the sampling interval, the model mismatch or the watermark amplitude the thresholds are recalibrated, because an owner would recommission the monitor after such a change; in sweeps of loss, jitter, clock offset and the threshold multiplier they are not, because those are disturbances that arrive after commissioning.

Sweeps, ablations and Monte Carlo

The architecture comparison uses 24 episodes per event and seed on three seeds (72 per event). The sweeps use 16 episodes per event and seed. One-factor sweeps vary the bias step, the ramp rate, the drift rate, the leak size, the replay variants, the watermark amplitude (with a blinding replay that forces nothing and blends its seam), packet loss, jitter, the sampling interval, the clock offset, the model mismatch, the noise multiplier and the threshold multiplier. Ablations remove the authentication code, the independent sensor or the watermark from A4, or give the attacker the key. The Monte Carlo analysis draws 200 combinations of twelve factors (noise multiplier, loss, jitter, sampling interval, clock offset, model mismatch, threshold multiplier, watermark amplitude, ramp rate, leak size, replay forcing and bias size) from uniform distributions over the sweep ranges, recalibrates for each draw on eight normal episodes, runs three episodes of each event with episodes shortened to 90 minutes, and records the mean delay (capped at 30 minutes), the miss rate at 30 minutes and the false alarms per day for A2 and A4. Because each draw is a small sample, only quintile means by factor are interpreted.

Verification

Four checks were applied before the experiments. With no event, each family's measured false-alarm rate on held-out episodes is reported against its target. With no event and no mismatch, the physics residuals have zero mean within their calibration scale. The watermark regression recovers the pump gain: the calibration t-statistic equals the value expected from the watermark amplitude and the flow noise. A bias step on LT-101 produces an immediate disagreement with LT-102 of the size of the step. These checks establish internal consistency, not fidelity to any plant.

Reproducibility

The simulation, the scenario runner, the Monte Carlo script, the aggregation and the result files are in a supplementary archive; fixed seeds reproduce every number. Three seeds for the sweeps and a Monte Carlo with three episodes per event and draw are small samples, and the study should be repeated with more seeds and with other plant models before any number is relied on.

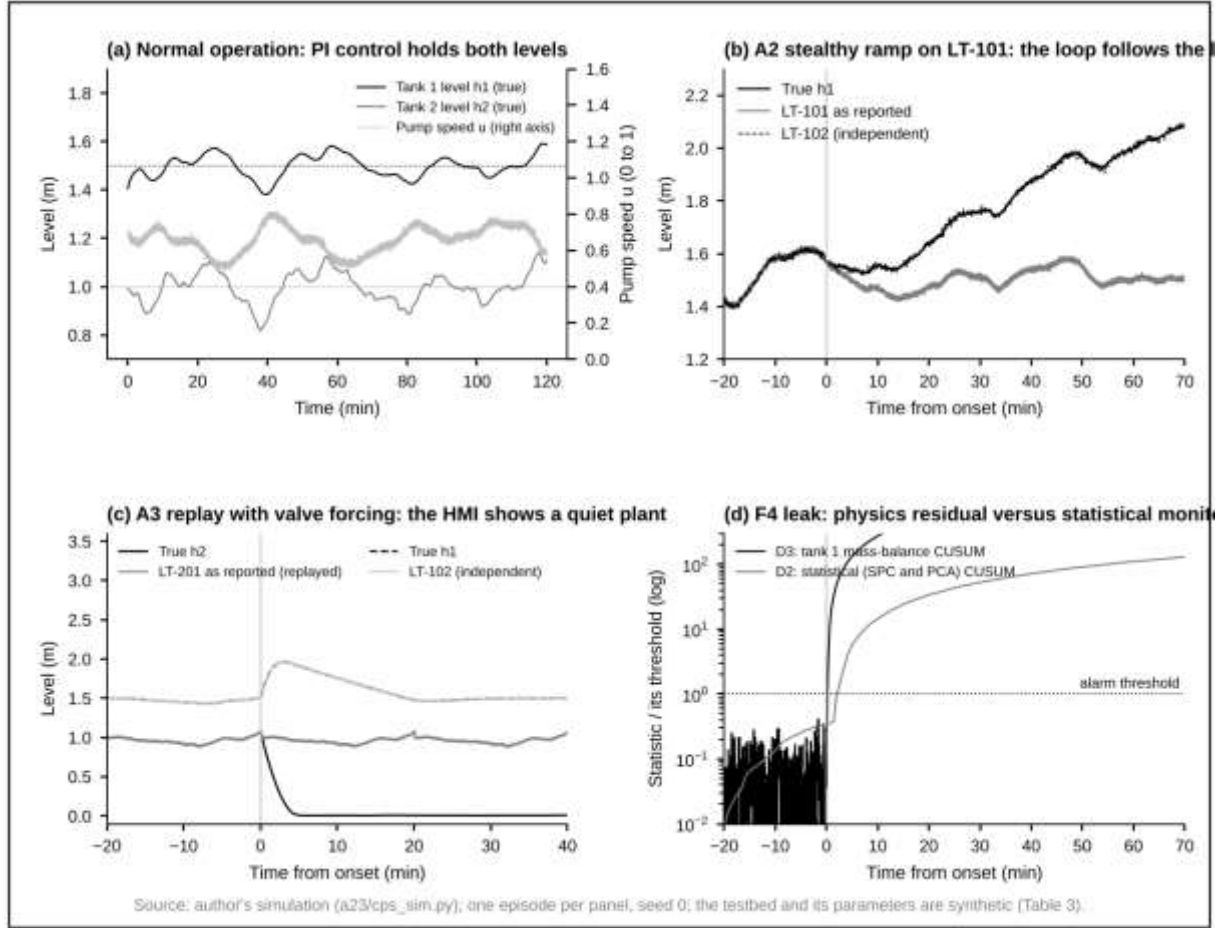
Results

Normal operation and the alarm budget

Figure 5a shows two hours of normal operation: the controllers hold the tank 1 level within about 0.04 m of its setpoint and the tank 2 level within about 0.06 m while the demand wanders, and the pump speed moves between 0.4 and 0.8 to follow it. The calibrated thresholds and the false alarms they produced on held-out normal episodes are summarised in Figure 6c. With a target of 0.5 alarms per day per family, the measured rates on held-out data were higher: 1.8 per day for the statistical family, 2.6 for the physics family, 1.6 for the cross-source check and 0.8 for the watermark test, giving 1.8 per day for A1, 4.4 for A2, 6.0 for A3 and 6.8 for A4, with a standard deviation over seeds of about 3.7 per day for A4. The excess over the target is the cost of calibrating on 60 hours of data with a maximum-

based search: the thresholds are fitted to the calibration run. The limit alarms, the authentication check and the stale-sequence check produced no false alarms. Each added family therefore buys its detection power with roughly one to two alarms per day in this testbed, a budget an owner must accept explicitly.

Figure 5: Testbed behaviour: normal operation, the stealthy ramp on LT-101, the replay with valve forcing, and the leak as seen by the physics and statistical detectors (one seed 0 episode per panel)

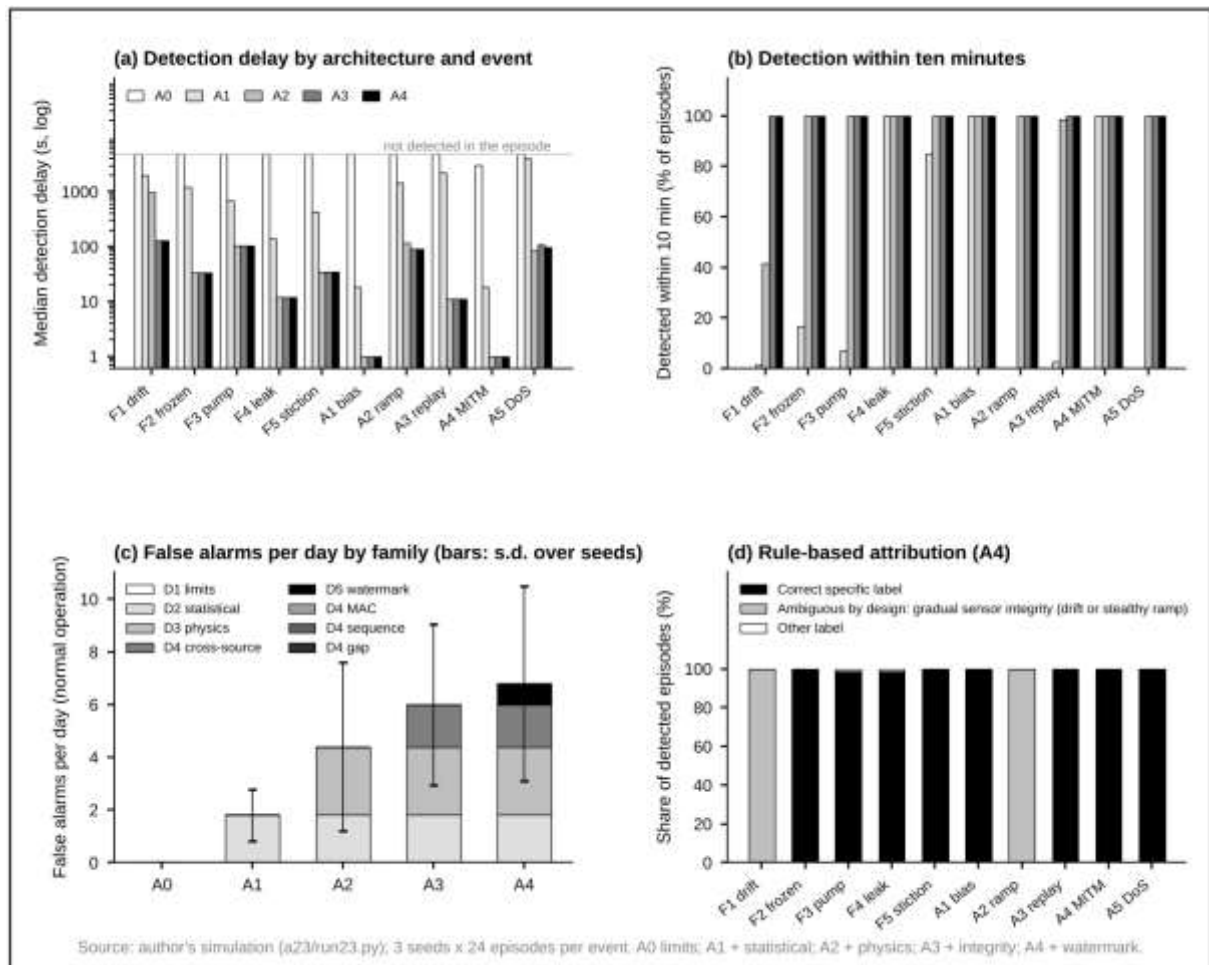


Architecture comparison

Table 7 and Figure 6 give the median detection delay and the share detected within ten minutes for each architecture and event (72 episodes per event). The baseline A0, with limit alarms only, detected nothing within the episode except a single late MITM episode: the controllers hold the reported levels inside the limits whatever the sensors say, and the events that do drive the true level to a limit (the frozen sensor and the replay) are invisible to limit alarms because the reported level is the frozen or replayed one. A0 is not a straw man; it is the posture of a plant whose only monitor is its HMI.

Table 7: Median detection delay in seconds and share detected within ten minutes, by architecture and event (72 episodes per event; an asterisk marks medians computed on fewer than half of the episodes; n.d. = not detected)

Event	A0 limits only	A1 + statistical	A2 + physics	A3 + integrity	A4 + watermark
F1 LT-101 drift	n.d. (0)	2028* (1)	992 (42)	131 (100)	131 (100)
F2 LT-201 frozen	n.d. (0)	1204 (17)	34 (100)	34 (100)	34 (100)
F3 pump degradation	n.d. (0)	692 (7)	104 (100)	104 (100)	104 (100)
F4 tank 1 leak	n.d. (0)	138 (100)	12 (100)	12 (100)	12 (100)
F5 valve stiction	n.d. (0)	425 (85)	35 (100)	35 (100)	35 (100)
A1 sensor bias step	n.d. (0)	18 (100)	0 (100)	0 (100)	0 (100)
A2 stealthy ramp	n.d. (0)	1436 (0)	117 (100)	91 (100)	91 (100)
A3 replay with forcing	n.d. (0)	2217* (3)	11 (99)	11 (100)	11 (100)
A4 telemetry MITM	3045* (0)	18 (100)	0 (100)	0 (100)	0 (100)
A5 DoS on telemetry	n.d. (0)	4118* (0)	84 (100)	108 (100)	98 (100)

Figure 6: Architecture comparison: median detection delay and detection within ten minutes by architecture and event, false alarms per day by detector family, and rule-based attribution in A4

A1 adds model-free statistics and detects every event eventually, but slowly for the gradual ones: 2,028 s for the drift (24% within 30 minutes), 1,204 s for the frozen sensor, 692 s for the pump degradation and 1,436 s for the stealthy ramp, against 18 s for the bias step and the MITM change, 138 s for the leak

and 425 s for stiction. The mechanism is that closed-loop control hides a sensor fault from the channel it affects (the reported level stays at the setpoint) and moves the evidence into the actuator, where it appears as a slow shift of pump speed that a CUSUM on autocorrelated data needs a long time to accumulate; the PCA squared prediction error helps because the relation between level, flow and speed breaks, but slowly. A1 missed most replays (8% within 30 minutes), as expected, since replayed data are statistically normal.

A2 adds the physics residuals and changes the picture for the faults: 34 s for the frozen sensor (the tank 2 balance stops closing), 104 s for the pump degradation (flow falls below the pump curve while the current still matches the flow), 12 s for the leak and 35 s for stiction, with every episode detected within ten minutes. For the drift, A2 reached 992 s with 42% within ten minutes: a drift of 0.3 m per hour contributes only 0.17 L/s to the tank 1 balance, a third of the flow noise, and the CUSUM barely accumulates it. For the stealthy ramp of 0.5 m per hour, A2 detected in 117 s, because the loop's response to the ramp (rising pump speed and flow with a falling reported level) violates the balance more strongly. A2 also detected the replay in 11 s, but only because of the seam: the replay is spliced at the recorded sample closest to the current values, and the residual over a 30 s window sees the discontinuity in the level derivative. This is an optimistic result for the defender and is examined below. A3 adds the integrity layer. The MITM change was detected at once by the authentication check (0 s) and the replay by stale sequence numbers (18 s) and by the independent sensor (28 s in the pure-replay variant), and the DoS was reported as an availability event by the gap share at 98 s rather than as a process alarm, whereas A1 and A2, with no sequence numbers, saw it as a process anomaly (A2 raised a physics alarm at 84 s that the rules of A4 would have called a fault). The independent sensor also cut the drift delay from 992 s to 131 s and the ramp from 117 s to 91 s, because a disagreement between two transmitters on the same tank accumulates directly, without passing through the plant. A4 adds the watermark and the attribution rules; its delays equal A3's at the defaults because the sequence check and the independent sensor fire before the watermark test (102 s in the pure replay), and its contribution is the attribution (Figure 6d) and robustness when the other checks are removed (the ablations below).

Process impact and detection margin

Table 8 gives the largest deviation of a true level from its setpoint after onset, the share of episodes in which a true level reached an alarm limit and the time at which it did, and for the events that reached a limit the share of episodes detected before it. Most events did not drive the process to a limit within the episode: the bias step moved tank 1 to 0.51 m above its setpoint (reported as on setpoint), the ramp to 0.63 m in 70 minutes, the drift to 0.39 m below, and the pump, leak and stiction faults were absorbed by the controllers. Two events did. The frozen sensor reached an alarm limit in 78% of episodes, at 1,551 s on average, and a physical limit (an empty or overflowing tank 2) in 67%, because the valve loop integrates a constant error on the frozen reading until the valve is fully open or closed; A2 and A4 detected it 1,269 s before the limit in the median episode, A1 only 204 s before, and A1 missed the limit in 12% of episodes. The replay with the valve forced closed emptied tank 2 in every episode within 170 s; A2 and A4 detected it about 150 s before, A1 almost never before (in the median episode 1,956 s after). The margin is the quantity an owner should ask for: a detector that fires after the tank is empty has told the operators what they already know.

Attack size and stealth

Table 9 and Figure 7 give the sweeps over attack size. A bias step is detected by A2 and A4 within two seconds at every size from 0.05 m upward, because the step appears in the level derivative and, for A4, in the disagreement with the independent transmitter at once; A1 needs 2,462 s for a 0.05 m step, 1,492 s for 0.1 m and 42 s for 0.2 m. A stealthy ramp divides the architectures more sharply. At 0.1 m per hour (3 cm in 20 minutes) A1 and A2 effectively failed within the episode (3% and 6% within ten minutes), while A4 detected in 296 s through the independent sensor; at 0.25 m per hour A2 detected in 2,153 s and A4 in 158 s; at 1 and 2 m per hour all three detected within 770, 40 and 40 s and within 372, 25 and 25 s, and the true level reached its alarm limit at 3,934 and 1,980 s, so every architecture detected before harm at the fast rates and only A4 did so at the slow ones. The drift fault, which is the same physics with the opposite sign, gave the same pattern: 3,774 s for A2 and 296 s for A4 at 0.1 m per hour, converging to 37 s for both at 1 m per hour. Leak size behaves as expected for a mass balance: A2 detected a 1 L/s leak (8% of the nominal flow) in 26 s and an 8 L/s leak in 8 s, while A1 needed 776 s

and 81 s.

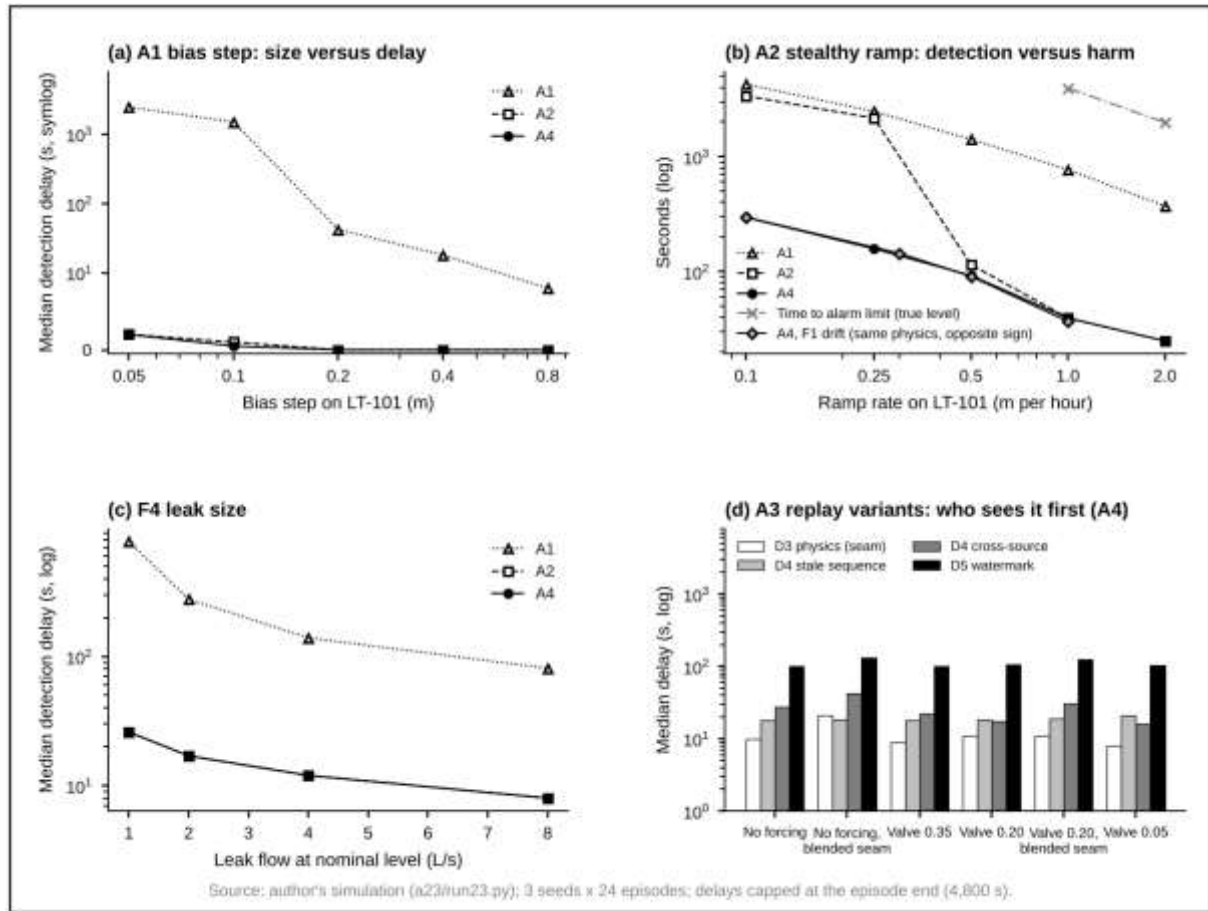
Table 8: Process impact after onset and the detection margin: largest level deviation, episodes reaching an alarm limit, time to the limit, episodes reaching a physical limit, and the share of episodes detected before the limit (median margin) for A1, A2 and A4

Event	Max. level deviation from setpoint (m)	Episodes reaching an alarm limit (%)	Mean time to alarm limit (s)	Episodes reaching a physical limit (%)	A1: detected before the limit (%) (median margin, s)	A2: same	A4: same
F1 LT-101 drift	0.39	0	–	0	–	–	–
F2 LT-201 frozen	0.90	78	1551	67	88 (204)	100 (1269)	100 (1269)
F3 pump degradation	0.30	0	–	0	–	–	–
F4 tank 1 leak	0.16	0	–	0	–	–	–
F5 valve stiction	0.19	0	–	0	–	–	–
A1 sensor bias step	0.51	0	–	0	–	–	–
A2 stealthy ramp	0.63	0	–	0	–	–	–
A3 replay with forcing	1.00	100	170	100	0 (-1956)	99 (152)	100 (152)
A4 telemetry MITM	0.14	0	–	0	–	–	–
A5 DoS on telemetry	0.14	0	–	0	–	–	–

Table 9: Attack size and stealth: median detection delay for A1, A2 and A4, largest level deviation and episodes reaching an alarm limit, for the bias step, the stealthy ramp, the drift fault, the leak size and the replay variants

Event and level	A1 median delay (s)	A2 median delay (s)	A4 median delay (s)	Max. level deviation (m)	Episodes reaching an alarm limit (%)
A1 bias step (m): 0.05	2462*	2	2	0.14	0
0.10	1492	1	0	0.19	0
0.20	42	0	0	0.30	0
0.40	18	0	0	0.51	0
0.80	8	0	0	1.00	3
A2 ramp (m/h): 0.10	4250*	3372*	296	0.17	0
0.25	2466*	2153*	158	0.34	0
0.50	1436	117	91	0.63	0
1.00	770	40	40	1.26	100
2.00	372	25	25	1.50	100
F1 drift (m/h): 0.1	4009*	3774*	296	0.17	0
0.3	2028*	992	131	0.39	0
0.5	1308	104	91	0.64	0
1.0	716	37	37	1.20	56
F4 leak (m ³ /s): 0.001	776	26	26	0.14	0
0.002	278	17	17	0.14	0
0.004	138	12	12	0.16	0
0.008	81	8	8	0.45	0
A3 replay, no forcing	2482*	10	10	0.14	0
A3 replay, no forcing, blended seam	1743*	21	15	0.14	0
A3 replay, valve 0.35	2643*	9	9	0.76	17
A3 replay, valve 0.20	2813*	11	11	0.93	100
A3 replay, valve 0.20, blended seam	23	11	11	0.93	100
A3 replay, valve 0.05 (default)	2217*	11	11	1.00	100

Figure 7: Attack size and stealth: bias step size, stealthy ramp rate against the time to harm, leak size, and the replay variants by detector



The replay variants (Figure 7d) show which check sees a replay first. With no actuator forcing (a pure blinding of the operators), the seam was seen by the physics residual in 10 s, the stale sequence numbers in 18 s, the independent sensor in 28 s and the watermark test in 102 s; blending the seam over 60 s pushed the physics residual to 21 s and the independent sensor to 43 s and left the sequence check unchanged. Forcing the valve to 0.35, 0.2 or 0.05 brought the independent sensor forward (22, 18 and 18 s) because tank 1 rises as soon as the valve closes, and the time to an alarm limit fell from 1,878 s to 351 s and 170 s. Two conclusions follow. The physics residual's replay detection depends on an imperfect seam and should not be relied on; a careful attacker with a longer recording or a model of the plant could remove it. The checks that do not depend on the seam, sequence numbers, the independent sensor and the watermark, work because each uses something the attacker could not record in advance: a counter, a second measurement, or a secret perturbation. Their delays (18, 28 and 102 s) are the real cost of a replay in this architecture, against 170 s to an empty tank in the worst variant.

Transport defects

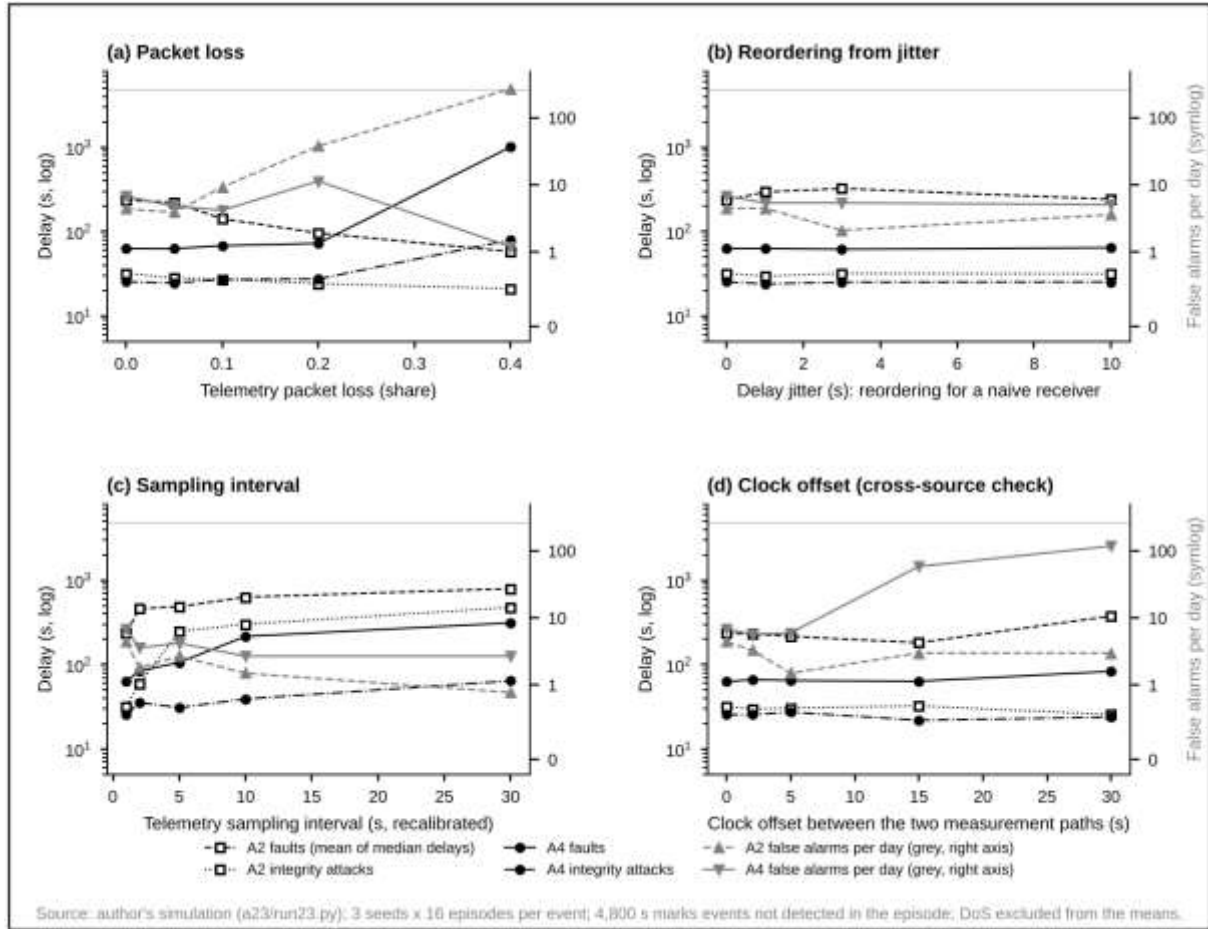
Table 10 and Figure 8 give the transport sweeps, with the mean of the median delays over the five faults and over the four integrity attacks (the DoS event is excluded from the means because it is itself a transport event), and the false alarms per day, for A2 with its naive receiver and A4 with its sequence-aware receiver. Packet loss separates the two receivers. The naive receiver holds the last value silently, so a lost sample looks like a flat spot and the physics residual sees a derivative that is not there: A2's false alarms rose from 4.4 per day with no loss to 9.3 at 10% loss, 38 at 20% and 270 at 40%, and its detection delays fell only because false alarms are counted as detections when they happen to follow the onset. The sequence-aware receiver knows which samples are missing, interpolates across them for display and does not update its detectors on them: A4's false alarms stayed at 4.2 per day at 10% loss and 11 at 20% (the watermark test contributes most of the rise, since fewer received samples carry the

watermark), with fault and attack delays within about 20 s of the lossless case. At 40% loss the share of missing packets hovered around the degraded-stream threshold of 0.35, so A4 alternated between suppressing its process detectors and running them: false alarms fell to 1.2 per day, the replay and MITM were still caught by sequence numbers and authentication (20 and 0 s), but the drift and pump degradation took 2,074 and 1,336 s and the DoS, which raises the loss further, was not reported as a new event because the stream was already flagged as degraded before the onset. A plant that loses a third of its telemetry has a transport problem to fix before it has a monitoring problem, and the architecture says so rather than guessing. Reordering from jitter had little effect on either receiver at the levels tested, because an adjacent swap changes a derivative for one sample and the CUSUMs do not accumulate it.

Table 10: Transport sweeps: mean of the median delays over the faults and over the integrity attacks, false alarms per day and the share of attacks detected within ten minutes, for A2 and A4, against packet loss, jitter, sampling interval and clock offset

Factor and level	A2 faults: mean of median delays (s)	A2 attacks (s)	A2 false alarms per day	A4 faults (s)	A4 attacks (s)	A4 false alarms per day	A4 attacks detected in 10 min (%)
Packet loss (share): 0	236	32	4.4	63	26	6.8	100
0.05	222	28	3.9	63	25	4.8	100
0.1	142	27	9.3	68	27	4.2	100
0.2	96	24	37.8	73	28	11.4	100
0.4	58	21	270.0	1004	80	1.2	54
Jitter (s): 0	236	32	4.4	63	26	6.8	100
1	295	30	4.5	63	24	5.4	100
3	324	32	2.1	62	25	5.4	100
10	242	32	3.6	65	25	5.1	100
Sampling interval (s):							
1	236	32	4.4	63	26	6.8	100
2	460	59	1.8	84	36	3.6	100
5	486	248	2.7	106	31	4.2	100
10	631	301	1.5	216	39	2.7	100
30	795	476	0.9	312	64	2.7	100
Clock offset (s): 0	236	32	4.4	63	26	6.8	100
2	230	30	3.3	66	26	5.7	100
5	216	31	1.5	65	27	6.0	100
15	182	32	3.0	64	22	58.5	100
30	376	26	3.0	83	24	117.6	100

Figure 8: Transport defects: packet loss, reordering from jitter, sampling interval and clock offset against detection delay and false alarms per day for A2 (naive receiver) and A4 (sequence-aware receiver)



The sampling interval acts on detection as it does in signal processing: the detectors were recalibrated at each interval, and delays grew roughly in proportion. At 5 s A4's fault mean rose from 63 to 106 s and the attack mean from 26 to 31 s; at 10 s to 216 and 39 s; at 30 s the drift and ramp were detected in 257 and 201 s and the leak in 49 s, against 131, 91 and 12 s at 1 s sampling. A2 suffered more, because its slow events depend on the physics residuals alone: its drift delay went from 992 to 2,630 s and its ramp delay from 117 to 1,818 s at 30 s sampling. Replay and MITM were almost unaffected (28 and 12 s at 30 s sampling), because sequence numbers and authentication do not depend on the sampling rate. The 30 s window of the mass balance contains a single sample at 30 s sampling, so the physics residuals lose their smoothing; an owner who can only sample slowly should lengthen the windows and expect proportionally longer delays.

The clock offset between the primary telemetry and the independent path is the one transport factor that broke a detector outright. The cross-source check compares the two transmitters at the same nominal time; with an offset, it compares them at different true times, and the natural movement of the level over the offset (about 2 mm per second during transients) becomes a disagreement. At 2 and 5 s the cross-source false alarms were 1.8 and 4.5 per day; at 15 s they were 55 per day and at 30 s 115 per day, which would make the check unusable. Detection delays were almost unchanged. The remedy is in the architecture rather than the detector: the two paths must share a disciplined time source, and the monitor should estimate and compensate the residual offset from the data (a cross-correlation of the two level signals during normal operation) and alarm on the time source when the offset drifts (Mizrahi, 2014; IEEE, 2020).

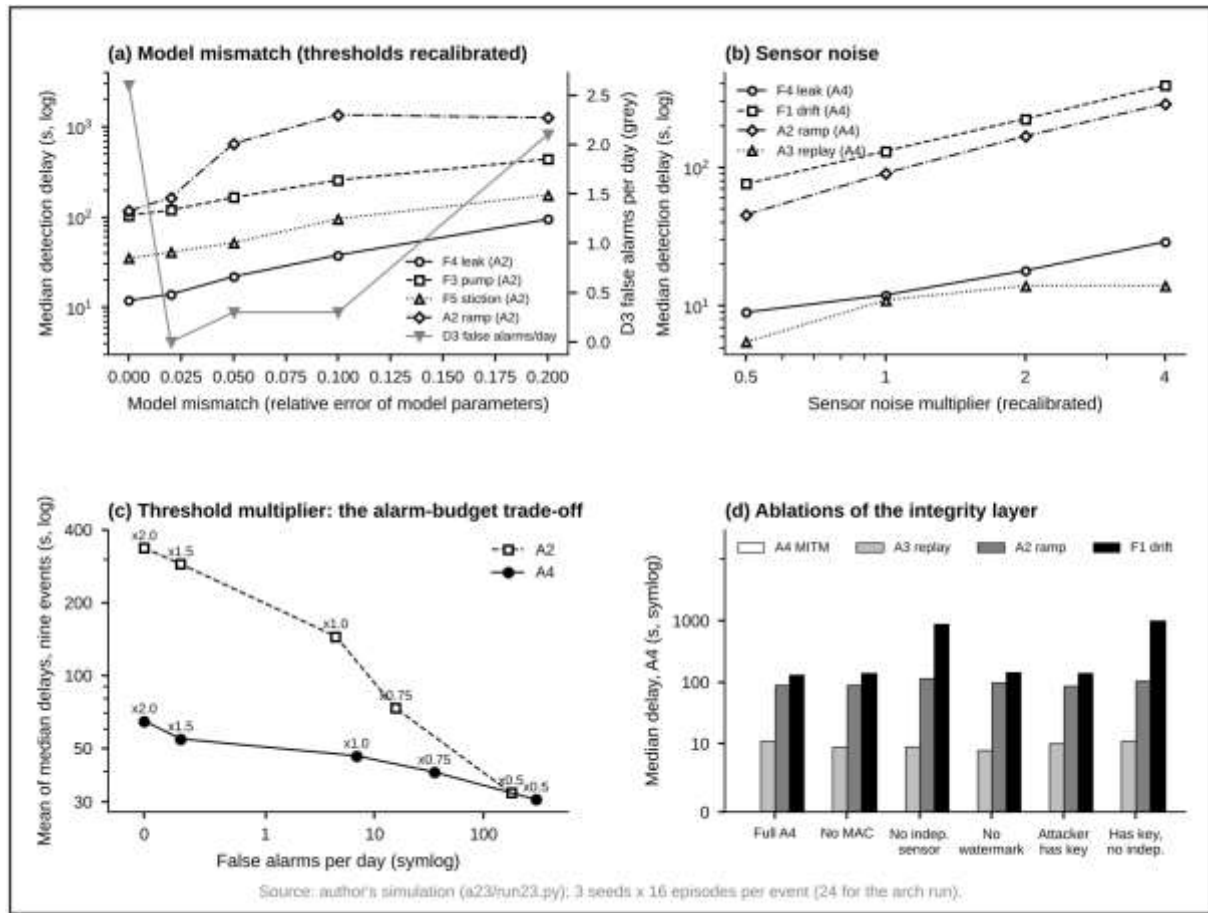
Model mismatch, noise and the threshold multiplier

Table 11 and Figure 9 give the model and threshold sweeps. Model mismatch degrades the physics residuals gradually once recalibrated: with 5% parameter error A2 detected the leak in 22 s against 12 s with a perfect model, with 10% in 38 s and with 20% in 94 s, while the pump-degradation delay went from 104 to 440 s. The stealthy ramp suffered most on A2, from 117 s to 1,360 s at 10% and 1,263 s at 20%, because the ramp's signature in the balance is small and the mismatch, which varies with the operating point, adds a structured residual that recalibration can only partly absorb. A4 was shielded by the independent sensor: its ramp and drift delays stayed near 105 s throughout. Sensor noise acts as expected: halving the noise cut A4's drift delay to 76 s and the ramp to 46 s; doubling it raised them to 224 and 168 s, and quadrupling to 391 and 288 s, with the leak at 29 s.

Table 11: Model mismatch, sensor noise and the threshold multiplier: selected detection delays and false alarms per day for A2 and A4 (thresholds recalibrated for mismatch and noise, not for the multiplier)

Factor and level	F4 leak, A2 (s)	F3 pump, A2 (s)	F1 drift, A4 (s)	A2 ramp, A4 (s)	A3 replay, A4 (s)	D3 false alarms per day (A2)	A4 false alarms per day
Model mismatch: 0	12	104	131	91	11	2.6	6.8
0.02	14	120	156	108	11	0.0	0.6
0.05	22	166	146	104	14	0.3	1.8
0.10	38	258	150	107	15	0.3	2.1
0.20	94	440	158	103	14	2.1	3.9
Noise multiplier: 0.5	9	66	76	46	6	1.2	3.6
1	12	104	131	91	11	2.6	6.8
2	18	164	224	168	14	0.6	1.5
4	29	282	391	288	14	0.0	2.7
Threshold multiplier: 0.5	9	82	80	47	8	171.0	300.9
0.75	10	94	112	70	8	13.8	35.4
1	12	104	131	91	11	2.6	6.8
1.5	14	107	164	118	12	0.0	0.3
2	16	124	186	131	12	0.0	0.0

Figure 9: Model mismatch, sensor noise, the threshold multiplier trade-off between false alarms and delay, and the ablations of the integrity layer



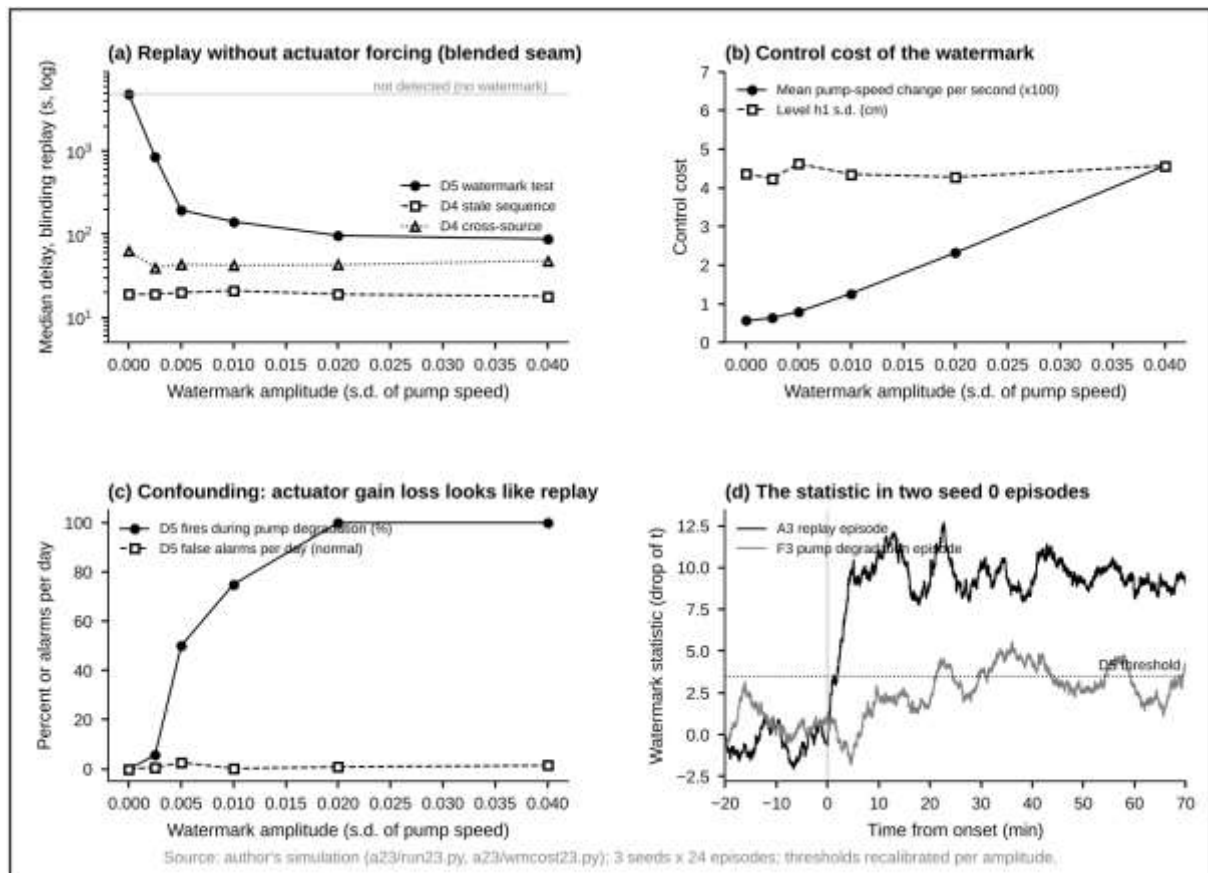
The threshold multiplier traces the alarm-budget trade-off (Figure 9c). Multiplying every calibrated threshold by 0.5 raised A4's false alarms to 301 per day and cut the mean of its median delays over the nine process and integrity events from 46 to 31 s; multiplying by 2 cut false alarms to zero in the held-out episodes and raised the mean delay to 65 s, with the slow events (drift, pump degradation, ramp) carrying most of the increase and the abrupt ones (bias, MITM, replay) almost unchanged. For A2 the same multipliers moved false alarms from 178 per day to zero and the mean delay from 33 to 336 s. The curve is the owner's choice to make, and the architecture cannot make it for them; what the results add is that the integrity checks (authentication, sequence, cross-source) sit outside the trade-off, since their thresholds are not statistical.

The watermark and the ablations

Table 12 and Figure 10 examine the watermark. Against a blinding replay with a blended seam (no actuator forcing, so that only the data lie), the watermark test detected in 142 s at the default amplitude (1% of pump speed, a signal-to-noise ratio of 0.8 in the flow sensor), 195 s at 0.5%, and 854 s with only 82% of episodes detected at 0.25%; at 2% and 4% the delay fell to 96 and 88 s. The sequence check (about 19 s) and the independent sensor (40 to 60 s) were faster at every amplitude, so the watermark is a backstop for the case in which the attacker rewrites sequence numbers and no independent sensor exists, not the first line. Its control cost was small in level terms (the tank 1 standard deviation stayed at about 4.3 cm at all amplitudes, since the level integrates the perturbation) but visible in actuator activity: the mean change of pump speed per second rose from 0.57% without a watermark to 1.26% at 1% and 4.6% at 4%, which is wear on a variable-speed drive and a cost the control engineer must accept. The watermark test also has a confound: a loss of pump gain weakens the flow's response to the watermark, so during the pump-degradation fault the test fired in 75% of episodes at the default amplitude and in 100% at 4%; the attribution rules place the pump-curve residual ahead of the watermark test for this reason, and no pump-degradation episode was labelled a replay.

Table 12: Watermark amplitude: detection of a blinding replay by the watermark, sequence and cross-source checks, confounding by pump degradation, false alarms and control cost

Setting (blinding replay, blended seam)	Watermark SNR in FT-101	D5 median delay (s)	D4 stale sequence (s)	D4 cross-source (s)	D5 fires under pump degradation (% in 30 min)	D5 false alarms per day	Mean pump-speed change per second (x100)	h1 s.d. (cm)
Watermark amplitude 0	0.0	n.d.	19	63	0	0.0	0.57	4.4
Watermark amplitude 0.0025	0.2	854	19	40	6	0.6	0.63	4.2
Watermark amplitude 0.005	0.4	195	20	44	50	2.6	0.80	4.6
Watermark amplitude 0.01	0.8	142	21	42	75	0.2	1.26	4.4
Watermark amplitude 0.02	1.6	96	19	43	100	0.8	2.32	4.3
Watermark amplitude 0.04	3.2	88	18	48	100	1.4	4.56	4.6

Figure 10: The watermark: replay detection against amplitude, control cost, confounding by pump degradation, and delay against the watermark signal-to-noise ratio

The ablations (Table 13, Figure 9d) show what each integrity component carries. Removing the authentication code left the MITM change detectable only by the statistical and physics families (0 s, as a step in the reported level) and removed its attribution as a network attack. Removing the independent sensor returned the drift and the ramp to the physics residual's delays (885 and 116 s) and left the replay to the sequence check and the watermark. Removing the watermark changed nothing at the defaults, since the sequence check and the independent sensor fire first. Giving the attacker the key removed the MITM's authentication evidence, and the independent sensor could not help because the MITM altered tank 2's reading while the independent transmitter is on tank 1; the change was then caught as a step

by the statistical and physics families and labelled a tank 2 level-sensor fault in 94% of episodes. With the key and no independent sensor, the replay was still caught by the sequence check and the watermark, and the MITM by the step. The lesson of the ablations is that no single component covers the integrity events, and that the independent sensor and the authentication code cover different attacks: one protects the conduit, the other provides a second witness to the physics.

Table 13: Ablations of the integrity layer in A4: median detection delay for the integrity attacks and the drift, false alarms per day, and the labels given to the MITM episodes

Ablation (A4)	A4 MITM (s)	A3 replay (s)	A1 bias (s)	A2 ramp (s)	F1 drift (s)	False alarms per day	Most frequent label for A4 MITM episodes
Full A4	0	11	0	91	131	6.8	network integrity attack (100%)
No MAC on telemetry	0	10	0	90	142	4.2	level sensor fault (tank 2) (100%)
No independent sensor	0	10	0	116	885	3.0	network integrity attack (100%)
No watermark	0	9	0	100	145	0.9	network integrity attack (100%)
Attacker holds the MAC key	0	10	0	88	140	3.3	level sensor fault (tank 2) (94%)
Attacker holds the key, no independent sensor	0	11	0	106	1000	2.7	level sensor fault (tank 2) (100%)

Table 14: Rule-based attribution in the architecture run and the Monte Carlo summary

Event or quantity	Labels given (share of detected episodes) or value	Specific label correct	Fault versus attack class correct
F1 LT-101 drift	sensor integrity: gradual (drift or stealthy attack) (100%)	ambiguous	100
F2 LT-201 frozen	level sensor fault (tank 2) (100%)	100	100
F3 pump degradation	pump fault (99%); sensor integrity: gradual (drift or stealthy attack) (1%)	99	99
F4 tank 1 leak	leak (99%); replay attack (1%)	99	99
F5 valve stiction	valve fault (100%)	100	100
A1 sensor bias step	sensor integrity: abrupt (attack likely) (100%)	100	100
A2 stealthy ramp	sensor integrity: gradual (drift or stealthy attack) (100%)	ambiguous	100
A3 replay with forcing	replay attack (100%)	100	100
A4 telemetry MITM	network integrity attack (100%)	100	100
A5 DoS on telemetry	telemetry availability (100%)	100	100
Monte Carlo draws; factors	200; 12 (ranges of Tables 10 and 11 plus attack sizes)	-	-
A2: mean delay capped at 30 min, miss rate, false alarms per day (median over draws)	539 s; 17%; 3.4	-	-
A4: same	260 s; 3%; 10.3	-	-
Draws where A4 misses more than 20% of events	3%	-	-
A4 attribution across the Monte Carlo draws (correct or designed-ambiguous label, % of detected episodes)	F1 51; F2 79; F3 87; F4 68; F5 85; A1 45; A2 51; A3 93; A4 100; A5 100	-	-

Attribution and the Monte Carlo analysis

Table 14 gives the labels assigned by the rule layer in the architecture run and the Monte Carlo summary. The rules labelled the MITM, the replay, the bias step, the frozen sensor, the leak, the stiction and the DoS correctly in almost every episode, and the pump degradation in 99% of episodes; the drift and the stealthy ramp were both labelled "sensor integrity: gradual (drift or stealthy attack)" in every episode, which is the designed ambiguous class. The fault-versus-attack class was therefore correct wherever the data allow it to be, and undecidable for the two gradual sensor events, which is the honest answer: a transmitter that disagrees slowly and steadily with its twin could be drifting or could be lied to, and the data cannot say which. The rules were written with knowledge of the simulated signatures, so these accuracies are an upper bound.

The Monte Carlo analysis varied twelve factors jointly over 200 draws (Figure 11). The median over draws of A4's mean delay, capped at 30 minutes, was 260 s (10th to 90th percentile 111 to 497 s) against 539 s (254 to 805 s) for A2, with miss rates at 30 minutes of 3% (0 to 17%) and 17% (3 to 30%), and false alarms of 10 per day (0 to 79) for A4 and 3.4 per day (0 to 24) for A2. A4's higher false-alarm rate is the price of its extra families under adverse draws, chiefly the cross-source check at clock offsets that the one-factor sweep had already shown to be fatal above about 10 s. In 3% of draws A4 missed more than a fifth of the events; these were draws that combined a threshold multiplier above 1.7 or a sampling interval of 2 to 4 s with high noise or loss of 15 to 30%. The events A4 missed most often were the drift (27% of episodes across draws, because slow drifts at low ramp rates with high noise take longer than 30 minutes), the DoS (14%, in draws whose baseline loss already flagged the stream) and the frozen sensor (9%); A2 missed the drift in 76% and the ramp in 22%. By the range of quintile means, the factors that moved A4's mean delay most were the threshold multiplier, the sampling interval, the model mismatch and packet loss, and the factors that moved its miss rate most were the sampling interval, packet loss and the threshold multiplier; the attack sizes (ramp rate, bias, leak) mattered for A2 and much less for A4, which is the independent sensor at work again. These are rankings from small samples (three episodes per event and draw) and should be read as orderings, not as effect sizes.

The attribution rules also degraded under the varied conditions. Across the draws, A4's label was correct (or the designed ambiguous label for the drift and the ramp) in 100% of the MITM and DoS episodes, 93% of replays, 87% of pump degradations and 85% of stiction episodes, but only 79% of frozen sensors, 68% of leaks, 51% of drifts and ramps and 45% of bias steps; under high noise, large clock offsets or loss the first alarm often came from a family other than the one the rules expect, and the abrupt-step test for the bias was defeated by coarse sampling. Rules that are exact at the commissioning conditions are therefore only a starting point, and the roadmap's shadow-mode phase exists to tune them.

Discussion

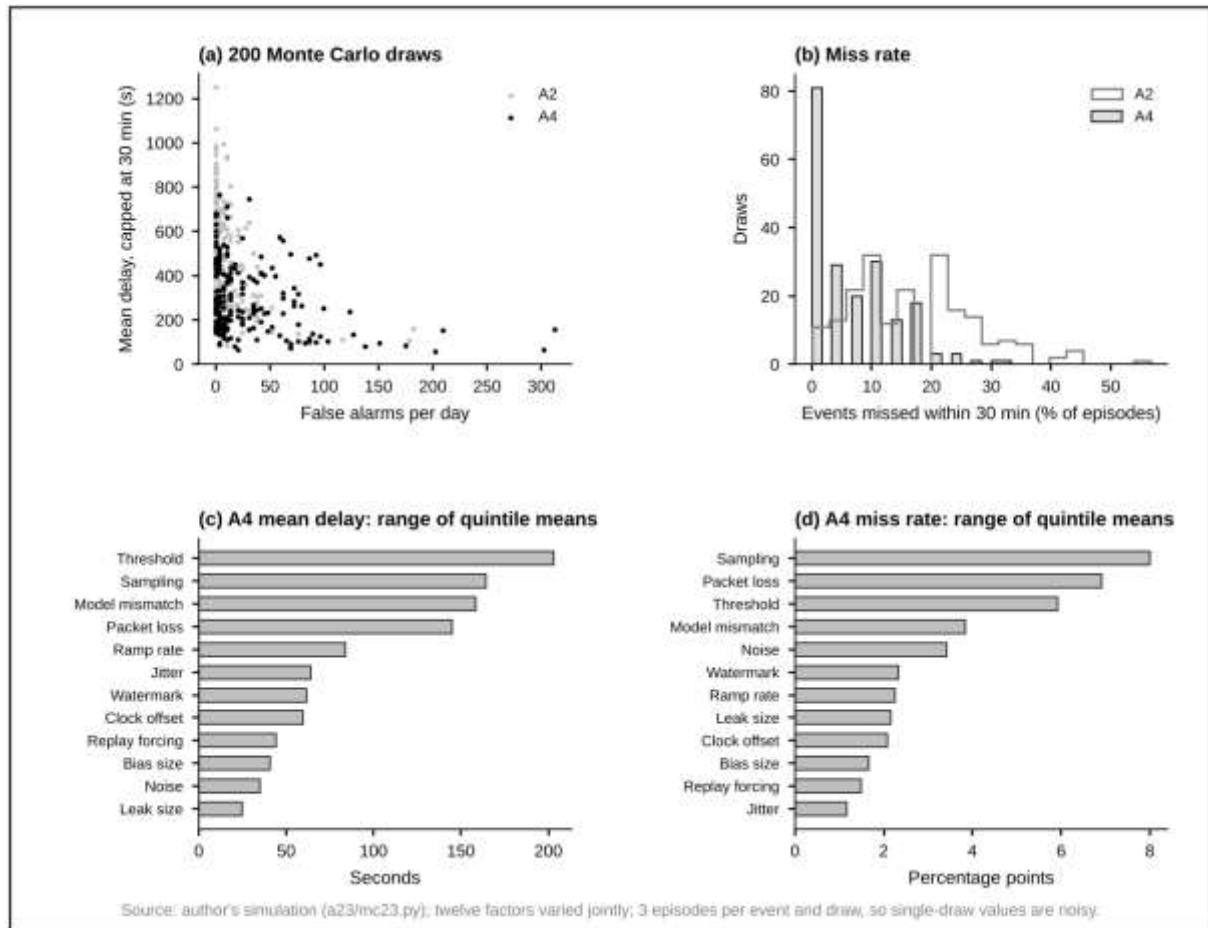
Answers to the research questions

RQ1. Each component covers a class of event that the others do not, and each costs about one to two false alarms per day at the calibration used. Limit alarms alone saw almost nothing, because closed-loop control holds the reported variables at their setpoints. Model-free statistics caught abrupt events (bias step, MITM) within about 20 s and gradual ones (drift, frozen sensor, pump degradation, stealthy ramp) in 12 to 34 minutes. Physics residuals caught the faults in 12 to 104 s and the stealthy ramp in about two minutes, and a slow drift only after 17 minutes. The integrity layer caught the MITM change at once by authentication, the replay within about 20 s by stale sequence numbers, the DoS as an availability event, and cut the drift and ramp delays to about two minutes through the independent sensor. The watermark added a replay backstop at 100 s. False alarms rose from zero (A0) through about 2 (A1), 4 (A2) and 6 (A3) to about 7 per day (A4).

RQ2. Attack size and stealth set the delay in the expected way, with the slow ramp as the discriminating case: at 0.1 m per hour only the independent sensor detected within the episode. Transport defects acted through the receiver: a naive receiver turned packet loss into physics false alarms, a sequence-aware receiver interpolated across gaps and was unaffected up to the point where it declared the stream degraded; reordering had little effect; slower sampling lengthened delays in proportion; and a clock offset between the primary and independent paths above a few seconds made the cross-source check unusable. Model mismatch and noise degraded the physics residuals gradually once recalibrated, with

the stealthy ramp the most sensitive; the threshold multiplier traced a steep trade-off in which halving the thresholds multiplied false alarms by an order of magnitude for a modest gain in delay.

Figure 11: Monte Carlo analysis over twelve factors: false alarms against mean delay for A2 and A4, miss rates, and the range of quintile means of A4's delay and miss rate by factor



RQ3. The data separate a fault from an attack when the attack leaves integrity evidence: a forged packet fails authentication, a replayed packet carries an old sequence number or ignores the watermark, a lost stream shows as gaps. They separate them partly when the physics shows a signature that no plausible fault produces: an abrupt step in a level reading, which the rules call "attack likely". They do not separate them when the attack imitates a fault exactly: a slow bias on a transmitter is drift or an attack, and the only honest label is the ambiguous one, to be resolved by inspecting the transmitter and the conduits. Attribution was correct wherever the evidence existed, but the rules were written with knowledge of the signatures, so this is an upper bound.

RQ4. The architecture gives technical content to guidance that mostly states outcomes: zones and conduits with an authenticated control-network conduit and a one-way link to the monitoring layer, an independent measurement path for the critical loop, detectors placed in the DMZ with calibrated thresholds and a stated alarm budget, attribution with evidence, and recalibration as a governed activity. The roadmap and the mapping to NIST, IEC, CISA, NERC and TSA material are proposals, and the owner should expect a monitor that catches the events studied here within seconds to minutes at a cost of several alarms per day, that must be recommissioned after changes, and that will sometimes be unable to say whether an alarm is a fault or an attack.

Why the architecture behaves as it does

Five mechanisms explain most of the results. First, closed-loop control hides sensor faults from the channel they affect and moves the evidence into the actuator: a biased level transmitter reads its setpoint while the pump speed shifts, so a per-channel chart on the level sees nothing and a chart on

the pump speed sees a slow drift that an autocorrelated CUSUM accumulates slowly. This is why the model-free family was slow on every gradual event and why the independent sensor, which does not pass through the loop, was fast. Second, authentication protects the conduit and not the content: the MAC caught every forged packet and could do nothing about a sensor that lied before the PLC, which is the Stuxnet pattern ([Langner, 2011](#)), nor about a controller that lied with the attacker's key. Third, a replay is exposed by anything the attacker could not record in advance: a sequence counter, a second measurement, or a secret perturbation; the physics residual's detection of the seam is a gift from an imperfect attacker and should not be counted on ([Mo & Sinopoli, 2009](#); [Mo et al., 2015](#)). Fourth, the watermark test and the pump-curve residual share a cause, since a loss of actuator gain weakens the response to the watermark as it weakens the response to the controller; the rules resolve the confound by precedence, and a design with a watermark on an actuator that cannot lose gain would avoid it. Fifth, every check that compares two sources needs them on one clock: the cross-source check failed at offsets of 15 s and above, which is well inside what unsynchronised controllers show, and the time source is therefore part of the detector and must be protected and audited ([Mizrahi, 2014](#); [Mills, 1991](#); [IEEE, 2020](#)).

A sixth point is about what the numbers mean. The false-alarm rates measured on held-out data exceeded the calibration target by a factor of two to five, and the standard deviation over seeds was large, because 60 hours of normal data fix a maximum-based threshold poorly. This matches the base-rate warning of the intrusion-detection literature ([Axelsson, 2000](#)) and the experience of security operations ([Alahmadi et al., 2022](#)): the alarm budget, not the detection power, is the binding constraint on a monitor, and it should be measured on the longest normal record available before the monitor is trusted.

Governance and deployment in the U.S. context

Figure 12 turns the findings into a four-phase roadmap with three gates, and Table 15 maps the architecture components to the guidance an owner is likely to be measured against. The first phase scopes and zones: an asset inventory and a P&ID review, a zone-and-conduit model in the manner of IEC 62443-3-2 ([International Electrotechnical Commission, 2020](#)), a consequence-driven list of the faults and attacks that matter for each loop, target security levels and detection goals per zone ([International Electrotechnical Commission, 2013b](#)), and a month of normal telemetry captured as the calibration baseline. The second phase secures the conduits: authenticated telemetry on the control-network conduit (sequence numbers and a message authentication code, DNP3 secure authentication or TLS where the devices support it ([IEEE, 2012](#); [Rescorla, 2018](#); [Nai Fovino et al., 2009](#))), a unidirectional gateway to the monitoring DMZ, hardening of the time source with an audit of clock offsets ([Mizrahi, 2014](#)), and the independent measurement path for the critical loop. The third phase instruments and models: plant models for the residuals verified on data, statistical baselines and the PCA model, a watermark on an actuator where the control cost is acceptable, thresholds calibrated to an alarm budget and the attribution rules written with operations. The fourth phase operates in shadow mode, reviews every alert with the operators, tunes thresholds and suppression against the false-alarm budget, recalibrates after any change of process, instrument or model, and keeps the evidence. The timings are indicative and were not derived from the study.

The mapping in Table 15 is the author's reading of the texts as they stood at the time of writing and makes no compliance claim. The NIST guide to OT security covers architecture, segmentation, monitoring and incident response for OT in general ([Stouffer et al., 2023](#)), and the Cybersecurity Framework 2.0 organises the same activities under its functions ([National Institute of Standards and Technology, 2024](#)); the CISA cross-sector performance goals give a baseline that includes segmentation, logging and detection ([Cybersecurity and Infrastructure Security Agency, 2023a](#)). The NERC CIP standards apply to the bulk electric system and include electronic security perimeters and system security management, and newer requirements on internal network security monitoring have been under development; an owner in that sector should check the current enforceable versions ([NERC, n.d.](#)). The TSA pipeline security directive, revised periodically, requires segmentation, access control, monitoring and detection, and incident reporting for covered pipeline owners ([Transportation Security Administration, 2025](#)); the version cited here is the one found at the time of writing, and later versions may differ.

Figure 12: Deployment roadmap for the secure monitoring architecture: four phases with stage gates, governance and enablers, and the recalibration loop

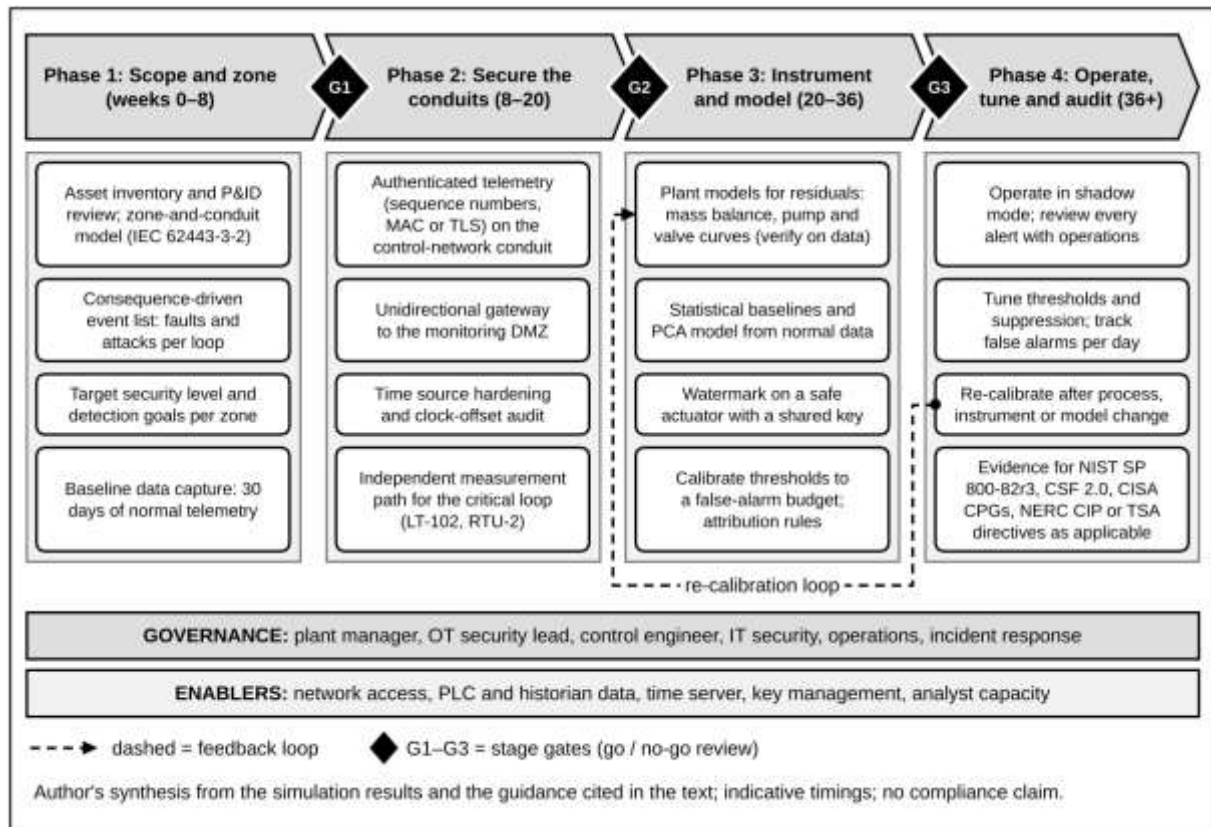


Table 15: Mapping of the architecture components to U.S. guidance and standards (author's reading; no compliance claim)

Architecture component	NIST SP 800-82r3 and CSF 2.0	IEC 62443 (3-2, 3-3)	CISA CPGs, NERC CIP, TSA directive, CIRCIA	Caveat
Zone-and-conduit model with a monitoring DMZ and unidirectional gateway	Network architecture and segmentation guidance; Protect and Detect functions	Zone and conduit risk assessment (3-2); network segmentation requirements (3-3)	Network segmentation goals; electronic security perimeters (CIP-005); segmentation measures in the pipeline directive	Mapping is the author's reading; the texts should be consulted for the current wording
Authenticated telemetry (sequence numbers, MAC or TLS) and time-source hardening	Communication protection; integrity of transmitted data	Data integrity and communication integrity requirements (3-3)	Secure connectivity goals; CIP-005 and CIP-007 controls	Many legacy protocols cannot be authenticated without gateways
Independent measurement path for the critical loop	Defence in depth; diversity	Resilience measures at the system level	Not addressed explicitly	Adds cost and a second asset to maintain
Physics-based, statistical and watermark detectors with calibrated thresholds	Continuous monitoring; anomaly detection (Detect)	Security monitoring requirements (3-3)	Detection and response goals; internal network security monitoring expectations in CIP (newer standards should be checked)	Thresholds and models need recalibration after changes
Rule-based attribution and alert evidence	Incident response and reporting (Respond)	Event logging and audit	CIRCIA reporting once rules are final; TSA incident reporting	The ambiguity between drift and stealthy attack must be carried into the response process
Roadmap with gates and recalibration	Continuous improvement (Govern)	Security program maturity	Cross-sector goals as a baseline	No compliance claim is made

CIRCI creates reporting obligations whose implementing rules were being finalised when this article was prepared (U.S. Congress, 2022), and Executive Order 14028 set federal direction on logging, zero trust and incident response (Executive Office of the President, 2021; Rose et al., 2020). For a manufacturing plant outside these sectors, the NIST and CISA material is voluntary guidance, and the architecture here is a way of giving it technical content. Incident reporting and response also depend on sound incident handling and on alert quality (Cichonski et al., 2012; ISO/IEC, 2023b; Vielberth et al., 2020), which is why the architecture reports attribution with its evidence rather than a bare alarm.

Implications for analytics and engineering practice

Several practical points follow. First, decide what each detector is for and place it accordingly: limit alarms protect against gross excursions that honest sensors show; statistical monitors catch relation breaks without a model but are slow on closed-loop data; physics residuals catch faults and the attacks that break the physics; integrity checks catch what happens to the data in transit; the independent sensor and the watermark catch what the others cannot. Second, calibrate thresholds to an alarm budget on real normal data and expect the measured rate to exceed the target; report false alarms per day as a first-class metric, as the alarm-management literature advises (International Society of Automation, 2016; Alahmadi et al., 2022). Third, make the receiver sequence-aware and let it tell the detectors when the stream is degraded, or transport defects will be reported as process events. Fourth, write the attribution rules with operations and keep an explicit ambiguous class: a gradual disagreement between a transmitter and its independent twin is either drift or an attack, and the response procedure, not the detector, must resolve it (a maintenance check of the transmitter is the natural first step, and the integrity evidence on the conduits decides whether the security team is called). Fifth, recalibrate after any change of process, instrument, model or network, because every threshold in this study was tied to a baseline. Sixth, keep the monitoring layer unable to write to control; the architecture's value comes partly from the fact that a compromise of the monitor cannot become a compromise of the plant.

Limitations

Table 16: Assumptions and limitations of the study, likely effect on the findings and what would reduce each

Assumption or limitation	Likely effect on the findings	What would reduce it
All data are synthetic; the plant, faults and attacks are stylised	Absolute delays and rates describe the testbed; orderings and mechanisms are the findings	Replay on a physical testbed (SWaT, WADI or HAI) and on plant data with labelled events
One two-tank process with two PI loops	Multivariable, batch or discrete processes may behave differently; the physics residuals are specific to this plant	Other plant models; model-free variants of D3
Attacks are of fixed form; the attacker does not adapt to the detectors	A stealthier attacker could keep residuals below thresholds; the replay seam is detectable here but need not be in practice	Optimal-attack analyses in the secure-control literature; adaptive attackers
The rules of the attribution layer were written with knowledge of the simulated signatures	Attribution accuracy is an upper bound; the ambiguous class is the honest part	Blind evaluation on events not used to design the rules
Thresholds are calibrated on 30 normal episodes per seed	Measured false-alarm rates exceed the target and vary between seeds	Longer baselines; sequential calibration
Transport effects are modelled simply (loss as hold, jitter as adjacent reordering, constant latency)	Burst losses and protocol-specific behaviour are not represented	Network emulation with real protocol stacks
Three seeds for sweeps; Monte Carlo with 3 episodes per event and draw	Single-draw values are noisy; only quintile effects are interpreted	More seeds and draws
Standards and regulations are summarised from the author's reading as of writing	Wording and dates may have changed	Verify against the current texts before use

Table 16 lists the main limitations. All data are synthetic; the plant is a two-tank process with two PI loops; the attacks are of fixed form and the attacker does not adapt; the attribution rules were written with knowledge of the simulated signatures, so their accuracy is an upper bound; thresholds are calibrated on 60 hours of normal data per seed; the transport model is simple; three seeds and a small Monte Carlo are modest samples; and the regulatory summary reflects the author's reading at the time of writing. The findings I regard as most likely to transfer are the mechanisms: that closed-loop control hides sensor faults from single-channel statistics and moves the evidence into the actuator; that authentication protects the conduit and not the content; that a replay is exposed by anything the attacker cannot record in advance, whether a watermark, an independent sensor or a sequence number; that an actuator gain loss confounds a watermark test; and that drift and a stealthy ramp are indistinguishable from the process data alone.

Further research

Four lines of work follow. The first is to replay the architecture on a physical testbed and its datasets (Mathur & Tippenhauer, 2016; Ahmed et al., 2017; Shin et al., 2020) and on plant data with labelled maintenance and security events, so that the delays and false-alarm rates can be calibrated against reality. The second is adaptive attackers: optimal stealthy attacks against the calibrated thresholds, in the manner of the secure-control literature (Teixeira et al., 2015; Bai et al., 2017), and attackers who learn the watermark. The third is richer plants: multivariable processes, batch operations, and discrete manufacturing, where the physics residuals take different forms, and the use of digital twins as the model source (Eckhart & Ekelhart, 2019; Tao et al., 2019). The fourth is the human side: how operators act on attributed alerts, how the ambiguous class is resolved in practice, and what alarm budget a control room can carry (Alahmadi et al., 2022).

Conclusion

A monitoring architecture for an industrial automation system should be designed like the plant it watches: with zones and conduits, with instruments placed where they can see what matters, and with a stated budget for alarms. On a synthetic two-tank process with five faults and five attacks, a baseline SCADA with limit alarms saw almost nothing, because closed-loop control keeps the reported levels at their setpoints whatever the sensors say. Model-free statistics caught abrupt events in seconds and gradual ones in half an hour or more. Physics residuals caught the faults in tens of seconds and the stealthy ramp in about two minutes, but could be blinded by a well-made replay and were slow on a slow drift. The integrity layer, authenticated telemetry with sequence numbers and an independently wired second transmitter, caught the telemetry attacks at once and the replay within about twenty seconds, cut the drift and ramp delays to about two minutes through the cross-source check, and reported a denial of service as what it was. The watermark added a backstop for replay at a cost in actuator activity and with a confound against loss of pump gain. False alarms rose from about two per day with statistics alone to about seven per day with the full design, and every added family bought its detection with roughly one to two alarms per day.

The architecture's weak points are specific. The cross-source check needs the two paths within a few seconds of a common time; packet loss above a third of the stream turns the monitor into an availability alert; thresholds are tied to a baseline that must be recalibrated after any change; and the data alone cannot separate a drifting transmitter from a stealthy attack on it, which the attribution layer reports as an ambiguous class rather than a guess. For a U.S. owner the architecture gives technical content to guidance that mostly says what to do and not how; the roadmap and the mapping to that guidance are proposals, the numbers describe a stylised testbed with three seeds, and both should be calibrated on a real plant before they are relied on.

REFERENCES

- [1]. Ahmed, C. M., Palleti, V. R., & Mathur, A. P. (2017). WADI: A water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWATER '17)* (pp. 25-28). ACM. <https://doi.org/10.1145/3055366.3055375>
- [2]. Ahmed, C. M., Zhou, J., & Mathur, A. P. (2018). Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In *Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC '18)* (pp. 566-581). ACM. <https://doi.org/10.1145/3274694.3274748>
- [3]. Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of the 31st USENIX Security Symposium* (pp. 2783-2800). USENIX Association. <https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi>

- [4]. Alladi, T., Chamola, V., & Zeadally, S. (2020). Industrial control systems: Cyberattack trends and countermeasures. *Computer Communications*, 155, 1–8. <https://doi.org/10.1016/j.comcom.2020.03.007>
- [5]. Amin, S., Litrico, X., Sastry, S., & Bayen, A. M. (2013). Cyber security of water SCADA systems: Part I: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5), 1963–1970. <https://doi.org/10.1109/TCST.2012.2211873>
- [6]. Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security*, 3(3), 186–205. <https://doi.org/10.1145/357830.357849>
- [7]. Bai, C.-Z., Pasqualetti, F., & Gupta, V. (2017). Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. *Automatica*, 82, 251–260. <https://doi.org/10.1016/j.automatica.2017.04.047>
- [8]. Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application*. Prentice Hall. <https://people.irisa.fr/Michele.Basseville/kniga/>
- [9]. Beerman, J., Berent, D., Falter, Z., & Bhunia, S. (2023). A review of Colonial Pipeline ransomware attack. 2023 *IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, 8–15. <https://doi.org/10.1109/CCGridW59191.2023.00017>
- [10]. Bhamare, D., Zolanvari, M., Erbad, A., Jain, R., Khan, K., & Meskin, N. (2020). Cybersecurity for industrial control systems: A survey. *Computers & Security*, 89, 101677. <https://doi.org/10.1016/j.cose.2019.101677>
- [11]. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>
- [12]. Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., & Sastry, S. (2011). Attacks against process control systems: Risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (pp. 355–366). ACM. <https://doi.org/10.1145/1966913.1966959>
- [13]. Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., & Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. *Computers & Security*, 56, 1–27. <https://doi.org/10.1016/j.cose.2015.09.009>
- [14]. Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. Springer. <https://doi.org/10.1007/978-1-4471-0347-9>
- [15]. Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *Computer security incident handling guide* (NIST Special Publication 800-61 Rev. 2). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-61r2>
- [16]. Conti, M., Donadel, D., & Turrin, F. (2021). A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials*, 23(4), 2248–2294. <https://doi.org/10.1109/COMST.2021.3094360>
- [17]. Cybersecurity and Infrastructure Security Agency. (2021). *Compromise of U.S. water treatment facility* (Alert AA21-042A). CISA. <https://www.cisa.gov/news-events/alerts/2021/02/11/compromise-us-water-treatment-facility>
- [18]. Cybersecurity and Infrastructure Security Agency. (2023a). *Cross-sector cybersecurity performance goals*. Cybersecurity and Infrastructure Security Agency. <https://www.cisa.gov/cross-sector-cybersecurity-performance-goals>
- [19]. Cybersecurity and Infrastructure Security Agency. (2023b). *IRGC-affiliated cyber actors exploit PLCs in multiple sectors, including U.S. water and wastewater systems facilities* (Advisory AA23-335A). CISA. <https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-335a>
- [20]. Dibaji, S. M., Pirani, M., Flamholz, D. B., Annaswamy, A. M., Johansson, K. H., & Chakraborty, A. (2019). A systems and control perspective of CPS security. *Annual Reviews in Control*, 47, 394–411. <https://doi.org/10.1016/j.arcontrol.2019.04.011>
- [21]. Ding, D., Han, Q.-L., Xiang, Y., Ge, X., & Zhang, X.-M. (2018). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275, 1674–1683. <https://doi.org/10.1016/j.neucom.2017.10.009>
- [22]. Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245–255. [https://doi.org/10.1016/0098-1354\(93\)80018-I](https://doi.org/10.1016/0098-1354(93)80018-I)
- [23]. Dragos. (2017). *TRISIS malware: Analysis of safety system targeted malware*. Dragos. <https://www.energy.senate.gov/services/files/40B2ED59-D34E-47C3-B9E2-1E8D030C5748>
- [24]. Eckhart, M., & Ekelhart, A. (2019). Digital twins for cyber-physical systems security: State of the art and outlook. *Security and Quality in Cyber-Physical Systems Engineering*, 383–412. https://doi.org/10.1007/978-3-030-25312-7_14
- [25]. Executive Office of the President. (2021). *Executive Order 14028: Improving the nation's cybersecurity*. Federal Register, 86 FR 26633. <https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity>
- [26]. Falliere, N., Murchu, L. O., & Chien, E. (2011). *W32.Stuxnet dossier (Version 1.4)*. Symantec Corporation. <https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en>
- [27]. Fawzi, H., Tabuada, P., & Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6), 1454–1467. <https://doi.org/10.1109/TAC.2014.2303233>
- [28]. Feng, C., Palleti, V. R., Mathur, A. P., & Chana, D. (2019). A systematic framework to generate invariants for anomaly detection in industrial control systems. In *Proceedings 2019 Network and Distributed System Security Symposium (NDSS)*. Internet Society. <https://doi.org/10.14722/ndss.2019.23265>
- [29]. Feng, X., & Hu, S. (2023). Cyber-physical zero trust architecture for industrial cyber-physical systems. *IEEE Transactions on Industrial Cyber-Physical Systems*, 1, 394–405. <https://doi.org/10.1109/TICPS.2023.3333850>
- [30]. Ferrag, M. A., Maglaras, L., Moschogiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, 102419. <https://doi.org/10.1016/j.jisa.2019.102419>

- [31]. Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3), 459–474. [https://doi.org/10.1016/0005-1098\(90\)90018-D](https://doi.org/10.1016/0005-1098(90)90018-D)
- [32]. Galloway, B., & Hancke, G. P. (2013). Introduction to industrial control networks. *IEEE Communications Surveys & Tutorials*, 15(2), 860–880. <https://doi.org/10.1109/SURV.2012.071812.00124>
- [33]. Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., Tippenhauer, N. O., Sandberg, H., & Candell, R. (2018). A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys*, 51(4), Article 76. <https://doi.org/10.1145/3203245>
- [34]. Goh, J., Adepu, S., Tan, M., & Lee, Z. S. (2017). Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)* (pp. 140–145). IEEE. <https://doi.org/10.1109/HASE.2017.36>
- [35]. Hadžiosmanović, D., Sommer, R., Zambon, E., & Hartel, P. H. (2014). Through the eye of the PLC: Semantic security monitoring for industrial processes. In *Proceedings of the 30th Annual Computer Security Applications Conference* (pp. 126–135). ACM. <https://doi.org/10.1145/2664243.2664277>
- [36]. Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security: A survey. *IEEE Internet of Things Journal*, 4(6), 1802–1831. <https://doi.org/10.1109/JIOT.2017.2703172>
- [37]. IEEE. (2012). *IEEE standard for electric power systems communications: Distributed network protocol (DNP3)* (IEEE Std 1815-2012). IEEE. <https://doi.org/10.1109/IEEESTD.2012.6327578>
- [38]. IEEE. (2020). *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems* (IEEE Std 1588-2019). IEEE. <https://doi.org/10.1109/IEEESTD.2020.9120376>
- [39]. Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., & Sun, J. (2017). Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 1058–1065). IEEE. <https://doi.org/10.1109/ICDMW.2017.149>
- [40]. International Electrotechnical Commission. (2013a). IEC 62264-1:2013 *Enterprise-control system integration: Part 1: Models and terminology*. IEC. <https://webstore.iec.ch/en/publication/6675>
- [41]. International Electrotechnical Commission. (2013b). IEC 62443-3-3:2013 *Industrial communication networks: Network and system security: Part 3-3: System security requirements and security levels*. IEC. <https://webstore.iec.ch/en/publication/7033>
- [42]. International Electrotechnical Commission. (2016). IEC 61511-1:2016 *Functional safety: Safety instrumented systems for the process industry sector: Part 1: Framework, definitions, system, hardware and application programming requirements*. IEC. <https://webstore.iec.ch/en/publication/24241>
- [43]. International Electrotechnical Commission. (2020). IEC 62443-3-2:2020 *Security for industrial automation and control systems: Part 3-2: Security risk assessment for system design*. IEC. <https://webstore.iec.ch/en/publication/30727>
- [44]. International Electrotechnical Commission. (2023). *Understanding IEC 62443*. International Electrotechnical Commission. <https://www.iec.ch/blog/understanding-iec-62443>
- [45]. International Society of Automation. (2016). ANSI/ISA-18.2-2016 *Management of alarm systems for the process industries*. ISA. <https://www.isa.org/products/ansi-isa-18-2-2016-management-of-alarm-systems-for>
- [46]. Isermann, R. (2005). Model-based fault-detection and diagnosis: Status and applications. *Annual Reviews in Control*, 29(1), 71–85. <https://doi.org/10.1016/j.arcontrol.2004.12.002>
- [47]. ISO/IEC. (2023b). ISO/IEC 27035-1:2023 *Information technology: Information security incident management: Part 1: Principles and process*. ISO. <https://www.iso.org/standard/78973.html>
- [48]. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45. <https://doi.org/10.1115/1.3662552>
- [49]. Knowles, W., Prince, D., Hutchison, D., Disso, J. F. P., & Jones, K. (2015). A survey of cyber security management in industrial control systems. *International Journal of Critical Infrastructure Protection*, 9, 52–80. <https://doi.org/10.1016/j.ijcip.2015.02.002>
- [50]. Kravchik, M., & Shabtai, A. (2018). Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy* (pp. 72–83). ACM. <https://doi.org/10.1145/3264888.3264896>
- [51]. Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3), 49–51. <https://doi.org/10.1109/MSP.2011.67>
- [52]. Lavin, A., & Ahmad, S. (2015). Evaluating real-time anomaly detection algorithms: The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 38–44). IEEE. <https://doi.org/10.1109/ICMLA.2015.141>
- [53]. Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (pp. 363–369). IEEE. <https://doi.org/10.1109/ISORC.2008.25>
- [54]. Lin, Q., Adepu, S., Verwer, S., & Mathur, A. P. (2018). TABOR: A graphical model-based approach for anomaly detection in industrial control systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS '18)* (pp. 525–536). ACM. <https://doi.org/10.1145/3196494.3196546>
- [55]. Linkov, I., Eisenberg, D. A., Plourde, K., Seager, T. P., Allen, J., & Kott, A. (2013). Resilience metrics for cyber systems. *Environment Systems and Decisions*, 33(4), 471–476. <https://doi.org/10.1007/s10669-013-9485-y>
- [56]. Lorden, G. (1971). Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6), 1897–1908. <https://doi.org/10.1214/aoms/1177693055>
- [57]. Lucas, J. M., & Saccucci, M. S. (1990). Exponentially weighted moving average control schemes: Properties and enhancements. *Technometrics*, 32(1), 1–12. <https://doi.org/10.1080/00401706.1990.10484583>

- [58]. Makrakis, G. M., Koliass, C., Kambourakis, G., Rieger, C., & Benjamin, J. (2021). Industrial and critical infrastructure security: Technical analysis of real-life security incidents. *IEEE Access*, 9, 165295–165325. <https://doi.org/10.1109/ACCESS.2021.3133348>
- [59]. Mathur, A. P., & Tippenhauer, N. O. (2016). SWaT: A water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)* (pp. 31–36). IEEE. <https://doi.org/10.1109/CySWater.2016.7469060>
- [60]. Mehra, R. K., & Peschon, J. (1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5), 637–640. [https://doi.org/10.1016/0005-1098\(71\)90028-8](https://doi.org/10.1016/0005-1098(71)90028-8)
- [61]. Mills, D. L. (1991). Internet time synchronization: The Network Time Protocol. *IEEE Transactions on Communications*, 39(10), 1482–1493. <https://doi.org/10.1109/26.103043>
- [62]. Mizrahi, T. (2014). *Security requirements of time protocols in packet switched networks (RFC 7384)*. Internet Engineering Task Force. <https://doi.org/10.17487/RFC7384>
- [63]. Mo, Y., & Sinopoli, B. (2009). Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 911–918). IEEE. <https://doi.org/10.1109/ALLERTON.2009.5394956>
- [64]. Mo, Y., Chabukwar, R., & Sinopoli, B. (2014). Detecting integrity attacks on SCADA systems. *IEEE Transactions on Control Systems Technology*, 22(4), 1396–1407. <https://doi.org/10.1109/TCST.2013.2280899>
- [65]. Mo, Y., Kim, T. H.-J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., & Sinopoli, B. (2012). Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE*, 100(1), 195–209. <https://doi.org/10.1109/JPROC.2011.2161428>
- [66]. Mo, Y., Weerakkody, S., & Sinopoli, B. (2015). Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine*, 35(1), 93–109. <https://doi.org/10.1109/MCS.2014.2364724>
- [67]. Morris, T. H., & Gao, W. (2014). Industrial control system traffic data sets for intrusion detection research. In *Critical Infrastructure Protection VIII (IFIP Advances in Information and Communication Technology, Vol. 441)* (pp. 65–78). Springer. https://doi.org/10.1007/978-3-662-45355-1_5
- [68]. Moustakides, G. V. (1986). Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4), 1379–1387. <https://doi.org/10.1214/aos/1176350164>
- [69]. Nai Fovino, I., Carcano, A., Masera, M., & Trombetta, A. (2009). Design and implementation of a secure Modbus protocol. In *Critical Infrastructure Protection III (IFIP Advances in Information and Communication Technology, Vol. 311)* (pp. 83–96). Springer. https://doi.org/10.1007/978-3-642-04798-5_6
- [70]. National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0 (NIST CSWP 29)*. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.CSWP.29>
- [71]. North American Electric Reliability Corporation. (n.d.). *CIP (Critical Infrastructure Protection) reliability standards*. NERC. <https://www.nerc.com/standards/reliability-standards/cip>
- [72]. Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2), 100–115. <https://doi.org/10.1093/biomet/41.1-2.100>
- [73]. Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11), 2715–2729. <https://doi.org/10.1109/TAC.2013.2266831>
- [74]. Qin, S. J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2), 220–234. <https://doi.org/10.1016/j.arcontrol.2012.09.004>
- [75]. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. A. (2010). Cyber-physical systems: The next computing revolution. In *Proceedings of the 47th Design Automation Conference (DAC '10)* (pp. 731–736). ACM. <https://doi.org/10.1145/1837274.1837461>
- [76]. Rescorla, E. (2018). *The Transport Layer Security (TLS) protocol version 1.3 (RFC 8446)*. Internet Engineering Task Force. <https://doi.org/10.17487/RFC8446>
- [77]. Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics*, 1(3), 239–250. <https://doi.org/10.1080/00401706.1959.10489860>
- [78]. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture (NIST Special Publication 800-207)*. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-207>
- [79]. Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2021). *Developing cyber-resilient systems: A systems security engineering approach (NIST Special Publication 800-160, Vol. 2, Rev. 1)*. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-160v2r1>
- [80]. Sandberg, H., Amin, S., & Johansson, K. H. (2015). Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems Magazine*, 35(1), 20–23. <https://doi.org/10.1109/MCS.2014.2364708>
- [81]. Satchidanandan, B., & Kumar, P. R. (2017). Dynamic watermarking: Active defense of networked cyber-physical systems. *Proceedings of the IEEE*, 105(2), 219–240. <https://doi.org/10.1109/JPROC.2016.2575064>
- [82]. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016b). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>
- [83]. Shin, H.-K., Lee, W., Yun, J.-H., & Kim, H. (2020). HAI 1.0: HIL-based augmented ICS security dataset. In *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association. <https://www.usenix.org/conference/cset20/presentation/shin>
- [84]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. <https://doi.org/10.1109/SP.2010.25>

- [85]. Stouffer, K., Pease, M., Tang, C., Zimmerman, T., Pillitteri, V., Lightman, S., Hahn, A., Saravia, S., Sherule, A., & Thompson, M. (2023). *Guide to operational technology (OT) security* (NIST Special Publication 800-82 Rev. 3). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-82r3>
- [86]. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE Access*, 10, 57143–57179. <https://doi.org/10.1109/ACCESS.2022.3174679>
- [87]. Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital twins and cyber-physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison. *Engineering*, 5(4), 653–661. <https://doi.org/10.1016/j.eng.2019.01.014>
- [88]. Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and recall for time series. In *Advances in Neural Information Processing Systems* 31 (NeurIPS 2018). Curran Associates. <https://papers.nips.cc/paper/2018/hash/8f468c873a32bb0619eae2050ba45d1-Abstract.html>
- [89]. Teixeira, A., Shames, I., Sandberg, H., & Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, 51, 135–148. <https://doi.org/10.1016/j.automatica.2014.10.067>
- [90]. Transportation Security Administration. (2025). *Security Directive Pipeline-2021-02F: Pipeline cybersecurity mitigation actions, contingency planning, and testing*. TSA. <https://www.tsa.gov/sd-and-ea>
- [91]. U.S. Congress. (2022). *Cyber Incident Reporting for Critical Infrastructure Act of 2022 (Division Y of the Consolidated Appropriations Act, 2022, Public Law 117-103)*. U.S. Government Publishing Office. <https://www.govinfo.gov/app/details/PLAW-117publ103>
- [92]. Urbina, D. I., Giraldo, J. A., Cardenas, A. A., Tippenhauer, N. O., Valente, J., Faisal, M., Ruths, J., Candell, R., & Sandberg, H. (2016). Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1092–1105). ACM. <https://doi.org/10.1145/2976749.2978388>
- [93]. Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., & Yin, K. (2003a). A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3), 327–346. [https://doi.org/10.1016/S0098-1354\(02\)00162-X](https://doi.org/10.1016/S0098-1354(02)00162-X)
- [94]. Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003b). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3), 293–311. [https://doi.org/10.1016/S0098-1354\(02\)00160-6](https://doi.org/10.1016/S0098-1354(02)00160-6)
- [95]. Vielberth, M., Böhm, F., Fichtinger, I., & Pernul, G. (2020). Security operations center: A systematic study and open challenges. *IEEE Access*, 8, 227756–227779. <https://doi.org/10.1109/ACCESS.2020.3045514>
- [96]. Williams, T. J. (1994). The Purdue enterprise reference architecture. *Computers in Industry*, 24(2-3), 141–158. [https://doi.org/10.1016/0166-3615\(94\)90017-5](https://doi.org/10.1016/0166-3615(94)90017-5)
- [97]. Willsky, A. S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6), 601–611. [https://doi.org/10.1016/0005-1098\(76\)90041-8](https://doi.org/10.1016/0005-1098(76)90041-8)
- [98]. Wu, R., & Keogh, E. J. (2023). Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 35, 2421–2429. <https://doi.org/10.1109/TKDE.2021.3112126>
- [99]. Yadav, G., & Paul, K. (2021). Architecture and security of SCADA systems: A review. *International Journal of Critical Infrastructure Protection*, 34, 100433. <https://doi.org/10.1016/j.ijcip.2021.100433>
- [100]. Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22(9), 1567–1581. <https://doi.org/10.1016/j.jprocont.2012.06.009>