JOURNAL OF
Emerging Paradigms
in Computing Systems
JEPCS

Journal of Emerging Paradigms in Computing Systems (JEPCS) | Volume 02, Issue No. 02, 2026

ARTICLE TYPE: Research Article

Article ID: JEPCS-2026-0202-0026

A Security Framework for Cloud-Enabled IoT and Digital Twin Systems

Ali Ahmad[1,*], Nauman Irshad Ali Shah[1], Saqib Dar[2], Danish Ali[1]

[1] Department of Computer Science, University of Central Punjab, Lahore, Pakistan
[2] Independent Researcher, Riyadh, Saudi Arabia

*Corresponding author: aliahmaddev61@gmail.com

Contributing authors: aliahmaddev61@gmail.com, naumanirshadalishah@gmail.com, saqi2926@gmail.com, danish.ali02721@gmail.com

Received: February 15, 2026 **Accepted:** March 28, 2026

Abstract- Cloud powered IoT and digital-twin systems are linking physical operations with a centralized computing facility, but their security measures are often installed at single points be they a device, the network, the cloud or an application. This fragmentation allows compromised telemetry, stolen device identities, manipulated twin models, or virtual-machine attacks to propagate across the physical-virtual feedback loop. This paper proposes a layered framework that combines secure IoT onboarding, gateway policy enforcement, hybrid encryption, protected virtualization, digital-twin anomaly detection, continuous monitoring, and adaptive service-level agreements (SLAs). A normalized device-risk function $R_i(t)$ and a covariance-weighted twin-deviation score A_t drive a composite SLA state $S(t)$ and a graduated response ranging from re-authentication to command blocking. The control loop was implemented as a reproducible simulator of a water-process twin under replay, impersonation, false-data injection, flooding, model tampering, and key-release attacks. In that experiment, attested key release reduced virtualization attack success from 84.0% to 4.0%, dynamic SLA response at 900 concurrent users was 105 ms compared with 221 ms for a static SLA, and the proposed loop obtained higher normalized protection scores than traditional single-layer controls on the VM, encryption, IoT, twin, and SLA layers. The results demonstrate coordinated protection across physical, edge, cloud, and virtual-model layers and provide paper-ready Figures generated from the same executable notebook.

Keywords- Cloud security; Internet of Things; Digital twin; Hybrid encryption; Zero trust

1. Introduction

Cloud computing provides on-demand access to shared storage, processing, and networking resources. Its scalability has made it a central platform for the Internet of Things (IoT), in which sensors, actuators, vehicles, medical devices, and industrial equipment generate continuous data streams.Digital twins builds on this relationship by creating an ongoing virtual mirror of the physical asset or process. These virtual models can be used for the monitoring of the physical asset or process and the diagnosis, simulation, or decision-making required for operation. The practical importance of this problem has been growing steadily. The ENISA Threat Landscape 2025 analyses 4,875 cybersecurity incidents recorded between 1 July 2024 and 30 June 2025, highlighting the continuing evolution and convergence of cyber threats across the EU [9]. When considering a cloud-connected cyber-physical system, the issue might be: a production process is interrupted by a tampered sensor input, health of a patient is impacted by an unsafe command transmitted to a medical-device, or public infrastructure impacted by a compromised traffic-control state. The incorporation of cloud computing, IoT, and digital twins (DTs) further increase the attack surface. Many resource-constrained IoT devices employ weaker authentication, old firmware or use sparse cryptographic control mechanisms. Corrupted telemetry data can create erroneous twin status;

an unauthorized command issued by the twin could impact physical system. Across the cloud the attack vectors of VM escapes, hypervisor compromises and cross-tenant attacks are ever-present threats [1]-[3].Static service-level agreements (SLAs) further restrict response because fixed security requirements do not reflect changing device trust, twin confidence, or threat conditions. Existing research commonly treats device security, virtualization, data protection, digital-twin trust, and service management as separate problems. The 2024-2026 literature has begun to close part of that gap by using digital twins as active security instruments for residual-based detection, orchestration, adversary simulation, and privacy-preserving learning [18]-[30], [32]. Even so, those systems still leave cloud-workload isolation, hybrid encryption of twin state, and adaptive SLAs outside the same control loop. The objective of this paper is to define an integrated framework that protects both cloud information assets and the physical-virtual feedback loop. The paper makes four contributions:

- A cross-layer architecture that connects IoT onboarding, gateway risk scoring, hybrid encryption, attested virtualization, twin-residual analysis, and adaptive SLAs in one feedback process.
- Bounded scoring models for device risk $R_i(t)$, twin deviation A_t, and composite SLA state $S(t)$, together with a graduated response algorithm that defaults to fail-safe operation when confidence is low.
- A working code implementation of the control loop on a physics-based water-process twin, covering impersonation, replay, false-data injection, flooding, model tampering, and VM key-release attacks.
- Experimental Figures and tables produced by that notebook, showing that attested key release and adaptive SLA profiles improve protection relative to traditional isolated controls.

The paper is organized as follow: Section 2 is about related work. Section 3 develops the integrated framework. Section 4 and Section 5 discuss the tests and applications of the framework within an executable simulation, respectively, and finally Section 6 gives a conclusion and future work.

2. Related Work

Works on cloud-enabled cyber-physical systems is concerned with security of cloud, virtualisation layer, integrity of IoT identity and gateway and integrity of DT. Since 2025 a new field have seen the twin itself considered as an operational security measure.

2.1. Cloud, IoT, and Twin Security

Zissis and Lekkas [1] classified (among other top cloud risks) most cloud risks per service models, while the NIST guidelines define that not all security tasks move entirely to the provider upon cloud deployment [3]. However, virtualization introduces a high reward "control plane". From the attack's perspective, a hypervisor takeover, a VM escape or a side channel attack from one VM to another may allow an intruder to transcend a single workload [19]. Secure boot, trusted execution environments (TEEs) and an attestation-gated release of private keys help mitigate this vulnerability [18], [19]. Security capabilities highlighted in the NIST IoT cybersecurity baseline include device identification, configuration, data protection, reduced interfaces, secured updates, and security-state awareness [4]. An enforcement point, such as the edge gateway, needs to be a feasible way to implement baseline security and it shouldn't turn into the single point of trust. Transition-model verification alongside the NIST Cybersecurity Framework can both test properties and structure governance output [12], [13]. Additionally, sector and regulatory specific constraints, like privacy under GDPR, HIPAA Security Rule, IEC 62443, and EU Cyber Resilience Act, make lifecycle support and industrial control assurance a required part of the design [14]-[17]. A DT is the digital counterpart of a physical thing which is synchronized in both directions [5]. Twin identity, model consistency, providence and freshness, and secure actuation are mandated for bi-directional telemetry and command flow [6]. Alcaraz and Lopez categorize DT threats across the functional hierarchy [18],Mun et al. Deduce common criteria requirement on DT architectures [19] and Qureshi et al. Survey security improving models and tools for DTs [20]. A twin model can recognize threats by validating authentication of observations against planned physical behavior [7]. Kampourakis et al. Bring industrial DT security in line with ISO 23247 [22], [31] and Alcaraz and Lopez analyze security trends of related standards [21], and a similar systematic review by [32] concluded the future role in with Industry 5.0 is DTs being an imminent security feature. The direction of the field is agreed upon, however, there seems to be a gap on a concrete cloud-to-device policy enforcement algorithm bounded to an auditable SLA.

2.2. Zero Trust, Encryption, and Recent Twin Frameworks

Security on identity, device, and requested resources must be assessed on the go by the Zero trust architecture [10]. The hybrid encryption method uses symmetric operations to protect bulk data and asymmetric key encapsulation to establish shared keying material [8]. The long-lived system should implement a crypto-agile solution to support cryptographic transitions, including migration to post-quantum algorithms [11]. Recent 2024-2026 literature represents a movement from taxonomies toward implemented twin-centric systems. Sinijoy et al. present a digital-twin-based multilevel attack detection and prevention scheme combining machine-learning and deep-learning techniques for industrial control security [23]. Sayghe uses a digital-twin-driven intrusion-detection framework that combines process simulation, physical residuals, and machine learning for industrial SCADA security [29]. Nguyen et al. Create incident playbooks from the

twin context, so-called SOAR4BC [24]. Baiardi et al. use a security twin with adversary simulation and Monte Carlo analysis for cyber-risk assessment [25]. Repetto develops a cybersecurity digital-twin blueprint with federation and zero-trust mechanisms for multi-ownership digital service chains [26], while Okegbile and Gambo combine adaptive AI, differential privacy, and zero-trust verification in IoT-enhanced digital-twin networks [27]. Issa et al. combine digital twins with blockchain-enabled federated learning to filter poisoned updates [28], while Salim et al. combine digital twins and federated learning for zero-day cyberthreat detection in IoT networks [30]. Although these designs improve audit ability, detection, trust, and privacy there is still research gaps on attested cloud isolation, hybrid encryption of twin states and translation device risk score and deviation score of twin state to agreed SLA profile. Table 1 positions representative approaches against the proposed scope.

Table 1: *Comparison of selected methods to protect your data.*

Approach	IoT	Twin	Cloud	Adaptive SLA	Cross-layer	Primary limitation
NISTIR 8259A [4]	Yes	No	Partial	No	No	Device baseline without being twin aware in it response
NIST IR 8356 [6]	Partial	Yes	Yes	No	Partial	Considerations of trust without algorithm implementation
Alcaraz and Lopez [18]	Partial	Yes	Partial	No	Partial	Taxonomy of threats, rather than a control loop
Sinijoy et al. [23]	Yes	Yes	No	No	Partial	DT-based multilevel detection without cloud isolation / SLA integration
Nguyen et al. [24]	Yes	Yes	Partial	No	Partial	SOAR playbook no TEE key to release/ SLA contract
Sayghe [29]	Yes	Yes	No	No	Combined	SCADA-focused DT-ID without cloud virtualization and SLA
Proposed framework	Yes	Yes	Yes	Yes	Yes	Simulator size validation; public ICS traces for a future work

Comparing showed which design constraint was carried over into the target methodology: that evidence from devices, physical models, cloud workload, and service performance will be fed into single policy enforcement mechanism. The section below then addresses that requirement. It specifies an architecture and define bounding scores and gateway-to-SLA reaction algorithm, instead of introducing yet another stand-alone detectors.

3. Materials and Methods / Methodology

The proposed framework assumes security to be a cohesive process starting with the device up through the edge gateway, the cloud, the DT, monitoring and SLA manager. Every layer generates some kind of evidence which is consumed by others.

3.1. Threat Model

The attacker may steal a device credential, resend or alter messages, inject bogus telemetry, exploit a VM that hosts the application logic or submits malicious model updates or flood a gateway with telemetry. Cryptographic primitive implementations are considered secure. Owning a correct credential does not imply physical correctness of the telemetry is known. Table 2 links the adversary actions with the required security property and control.

Table 2: *Threat model and required security properties.*

Adversary action	Targeted property	Required control
Device impersonation or replay	Authenticity and message freshness	Mutual authentication, sequence numbers, timestamp validation, and credential revocation
Plausible false-data injection	State integrity and physical safety	Twin-residual checks, provenance tracking, and human approval for high-impact commands
Undesired twin-model updates	Model trustworthiness and accountability	Signed model versions, approved transitions, revert capability and audit trails
VM escape/VM movement	Workload confidentiality and isolation	Secure boot, remote attestation, TEE backed key-release and network segregation
Gateway or service flooding	Availability and bounded degradation	Rate limiting, bounded queues, load monitoring, and an adaptive SLA profile

3.2. System Architecture

The control loop as implemented is summarized in Figure 1. Reading from top-to-bottom on the left-hand side, across to the right; the physical assets & IoT devices push telemetry and pull commands through the secure edge gateway. The gateway is the first enforcement point, where identity, message recency, integrity, and access policy are enforced, and where $R_i(t)$ from equation 1 is computed.Only then is data forwarded to the cloud virtualization and hybrid encryption layer, where workloads are isolated and where key release is tied to attested images and bulk telemetry is symmetrically wrapped for protection. The cloud's trusted state is exchanged with the digital-twin block, which predicts $\widehat{\mathbf{x}}t$, constructs At from eq 2, and passes the deviation score to the dynamic SLA manager. The SLA manager constructs $S(t)$ by aggregating risk, residual, vulnerability, and load (eq 4), and the policy that results-normal, elevated, restrictive or recovery-is returned to the gateway and to the monitor. Security monitoring sits over the top of the SLA manager, allowing alerts, queue depth, or attestation failure to cause $S(t)$'s rise more quickly; command flows also pass back down the same structure, where a higher $S(t)$ allows blocking of actuation at the gateway. This is the diagram that the implemented simulator executes at every timestep.

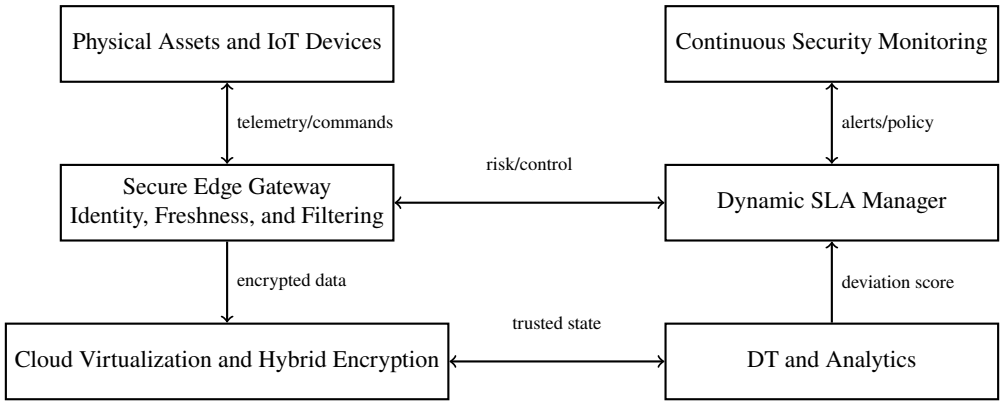

Figure 1: *Integrated security architecture for cloud-enabled IoT and digital twin systems.*

No layer makes an isolated trust decision in the proposed architecture: gateway and twin evidence converge at the SLA manager, whose policy can constrain both cloud processing and physical actuation. This feedback path is the structural basis for the scoring and response logic defined next, as shown in Figure 1.

3.3. Scoring Models Implemented in Code

Table 3 defines the symbols used both in the paper and in the executable code.

Table 3: *Principal notation used by the framework and the implemented simulator.*

Symbol	Definition
$z_{ik}(t)$	Normalized value of risk indicator k for device i at time t
α_k	Nonnegative weight of risk indicator k
m	Number of device-risk indicators
$R_i(t)$	Aggregate device-risk score
$\mathbf{x}_t$, $\widehat{\mathbf{x}}_t$	Observed state and twin prediction
$\mathbf{W}$, $\mathbf{\Sigma}$	Residual weighting matrix and benign-state covariance matrix
A_t, $\overline{A}_t$	Instantaneous and EWMA twin-deviation scores
λ	EWMA smoothing coefficient
$\overline{R}(t)$, $\overline{A}(t)$	Normalized aggregate device risk and twin deviation
$V(t)$, $L(t)$	Normalized vulnerability and service-load indicators
β_j	Nonnegative weight of SLA component j
$S(t)$	Composite SLA security state

The gateway computes a bounded monotone risk score

$$R_i(t) = \sum_{k=1}^{m} \alpha_k z_{ik}(t), \qquad \sum_{k=1}^{m} \alpha_k = 1, \qquad 0 \le R_i(t) \le 1. \tag{1}$$

Indicators include authentication failures, firmware age, malformed traffic, message rate, and prior quarantine. In the notebook these weights are $\boldsymbol{\alpha} = (0.30, 0.20, 0.15, 0.20, 0.15)$, and $m = 5$ is the number of indicators.

The twin residual is the squared Mahalanobis distance

$$A_t = \left(\mathbf{x}_t - \widehat{\mathbf{x}}_t\right)^T \mathbf{W} \left(\mathbf{x}_t - \widehat{\mathbf{x}}_t\right), \tag{2}$$

with $\mathbf{W} = \boldsymbol{\Sigma}^{-1}$ estimated on a benign calibration window, where $\boldsymbol{\Sigma}$ is the covariance matrix of benign residuals and $\mathbf{W}$ weights deviations by their expected variability. Isolated spikes are suppressed by

$$\overline{A}_t = \lambda A_t + (1 - \lambda)\overline{A}_{t-1}, \qquad \lambda = 0.25. \tag{3}$$

where $\lambda \in (0, 1]$ is the EWMA smoothing coefficient and $\overline{A}_{t-1}$ is the preceding smoothed deviation score.

The SLA manager forms

$$S(t) = \beta_1 \overline{R}(t) + \beta_2 \overline{A}(t) + \beta_3 V(t) + \beta_4 L(t), \tag{4}$$

Here $\overline{R}(t)$ and $\overline{A}(t)$ are normalized aggregate device risk and twin deviation, while $V(t)$ and $L(t)$ are normalized vulnerability and service-load indicators. The nonnegative component weights sum to one and are set to $\boldsymbol{\beta} = (0.35, 0.35, 0.15, 0.15)$. Thresholds on $S(t)$ select normal, elevated, restrictive, or recovery profiles (Table 4). Unlike SOAR playbooks generated after an alert [24], $S(t)$ is a signed service profile chosen before the incident.

Table 4: *Adaptive SLA states and technical actions.*

State	Normalized range	Representative actions
Normal	$0 \le S < \theta_1$	Standard logging, automated low-impact commands
Elevated	$\theta_1 \le S < \theta_2$	Step-up authentication, extra monitoring
Restrictive	$\theta_2 \le S < \theta_3$	Telemetry-only mode, command approval, isolation
Recovery	$S \ge \theta_3$	Fail-safe control, re-attestation, key rotation

Algorithm 1 converts deterministic gateway checks and continuous risk evidence into a containment decision and an updated SLA state. It is implemented directly in the code function `respond()` and is executed at every simulator timestep.

Encrypted virtualization only provides decryption keys when attested. Public-key encryption provides wrap over session.HMAC-SHA-256 over larger batches is provided within Hybrid Eneryption. The results delivered by these techniques are testable: layer decisions (gateway choice, device risk, double residuals, key-release result, service state, response latency) in each layer, so we map those in section 4 in to an attack model to define a range of metrics. This allows evaluating the framework as a control loop not only as architecture.

4. Experimental Setup

The evaluation uses the executable code provided through the https://github.com/aliahmad-gif/A-Security-Framework-for-Cloud-Enabled-IoT-and-Digital-Twin-Systems. It requires only NumPy, pandas, and Matplotlib. No GPU and no private dataset are used. A physics-based water tank is the physical plant and a process model is the DT. This evaluation asks whether gateway and twin evidence can distinguish attacks from non-malicious execution and do attestation and hybrid protection improve the reduction of exposure on cloud layer. Also does the adaptive SLA sustain its response time under escalating workload. They are measured by detection rate, false-positive rate, success rate, normalized protection rate, response time and cost for wrapping, respectively.

4.1. Simulator and Attack Windows

Each timestep the plant updates tank level, inflow, and outflow. The twin predicts the next state. The gateway then runs Algorithm 1. A 4000-step run uses a calibration window $[0, 800)$ to estimate $\mathbf{W}$. Attacks are injected in the windows of Table 5.

Algorithm 1 Gateway decision and SLA update process

Require: Device i, message M_t, predicted state $\widehat{\mathbf{x}}_t$, policy thresholds τ_R, τ_A, and τ_C
Ensure: Accepted telemetry or a documented containment action
Verify credential, authorization, integrity, sequence, and freshness
if any deterministic verification fails **then**
    Reject M_t, increase $R_i(t)$, and record the failed control
    return containment decision
end if
Decode observation $\mathbf{x}_t$ and compute $R_i(t)$ using Equation (1)
Compute A_t using Equation (2)
Update $\overline{A}_t$ using Equation (3)
if $R_i(t) \geq \tau_C$ or $\overline{A}_t \geq \tau_C$ **then**
    Isolate the source, block high-impact commands, and require human review
else if $R_i(t) \geq \tau_R$ or $\overline{A}_t \geq \tau_A$ **then**
    Re-authenticate the device and restrict it to telemetry-only operation
else
    Accept the state update and continue monitoring
end if
Update the SLA security state $S(t)$ and append an auditable event
return selected decision

Table 5: *Implemented attack scenarios, code hooks, and evaluation measures.*

Attack (code name)	Window	What the code does
Replay	900-1100	Reuses the last observation and clears the freshness flag
Impersonation	1300-1450	Fails authentication and inflates the observed level
False-data injection	1700-2050	Adds a physically inconsistent level drift to authenticated telemetry
Flood	2300-2550	Raises message-rate and malformed-traffic indicators and load $L(t)$
Model tamper	2800-3000	Distorts the twin state used as $\widehat{\mathbf{x}}_t$
VM key-release	3300-3600	Attempts decryption-key access; protected mode admits an 8% false-accept rate

Detection rate and false-positive rate follow

$$\mathrm{DR} = \frac{TP}{TP + FN}, \qquad \mathrm{FPR} = \frac{FP}{FP + TN}. \tag{5}$$

Here TP and FN are attacked timesteps correctly detected and missed, respectively, while FP and TN are benign timesteps incorrectly flagged and correctly accepted. Thus, DR measures attack coverage and FPR measures disruption of benign operation. Ablation configurations disable twin residual, gateway risk, attestation, or adaptive SLA one at a time.Conventional controller permits all the messages and does not guarantee any key release.

4.2. Figure Generation

In the following below are all result plots generated by the notebook, Figs 2-6. The plots as well as the tables then came from exactly the same executed notebook. Now that the scenarios and the measures have been established, this section considers if the evidence collected answers the three evaluation questions. We track control loop as whole in time, then detail layer protection, VM key release, SLA latency and cost of encryption individually.

5. Results and Discussion

All numbers in this section have been extracted from the code used to run the included plots. These are numbers specific to the control loop as implemented on the water-process twin, not measurements taken on the real cloud or on SWaT/WADI.

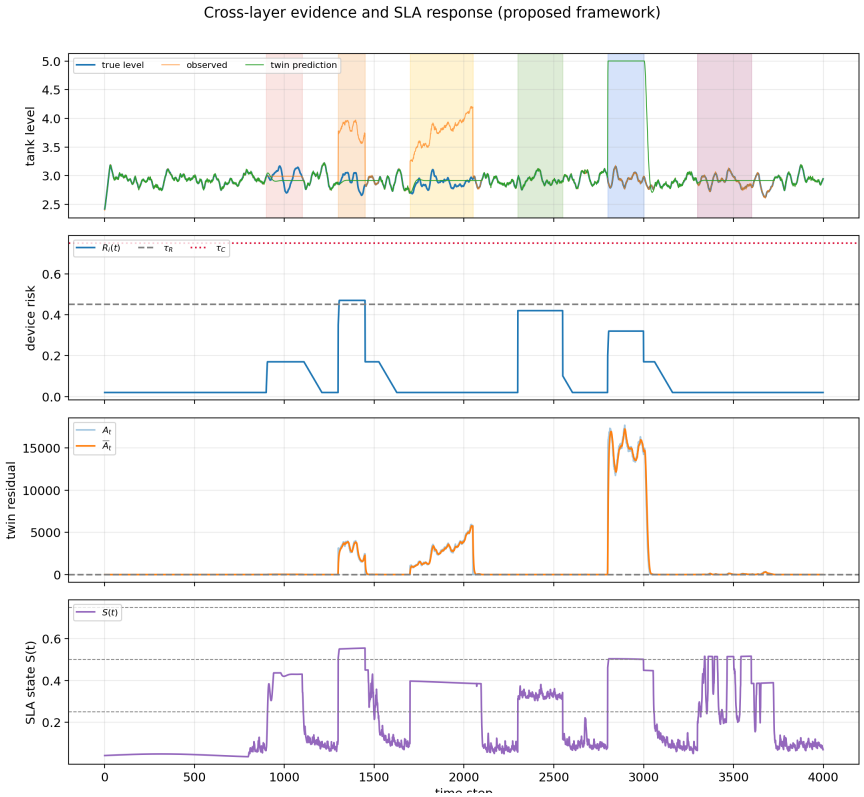

Figure 2: *Cross layer trace of evidence and SLA reaction on the simulated system. Dark bands denote attack periods. Top-down order: tank level (true, observed and twin), device risk $Ri(t)$ along with the two values of τ_R and τ_C. Twin residual At along with $\overline{A}t$, and the state $S(t)$ corresponding to the SLA constraint.*

5.1. Cross-Layer Evidence Over Time

The simulator has written one record per time step: tank level, gateway choice, risk, residual and SLA state. These are plotted in Figure 2 in four stacked panels over the same x-axis of simulation steps 0..4000. The top panel is the plant: true level, IoT report level and twin prediction $\widehat{\mathbf{x}}_t$. Six attack intervals given in Table 5 are depicted by colored vertical bands. The three curves cluster at nearly level 3 during healthy period. The impersonation attack causes rapid fluctuations on the reported value while true value and twin trace match, because gateway discards the unauthenticated packet. False data injection starts slower by pushing up the reported value, while true state tracks physical system and grows. The twin, hence, gets separated from the actual plant. As opposite, model tampering drives the plant toward the actuator saturation value against physics of water plant while the predicted states track on its pre-attack trajectory which is precisely what A_t monitors.

The second panel shows the gateway risk $Ri(t)$ in Eq. (1). The horizontal lines correspond to policy thresholds on policy thresholds $\tau_R = 0.45$ (restrict) and $\tau_C = 0.75$ (contain). The risk stays close to 0 for calibration and clear runs, and shoots up once authentication has failed, or freshness buffer has been emptied, or a piece of traffic has malformed content. Note that impersonation or flood attacks raise $R_i(t)$ without any need of any physics residuals. The third panel shows the twin residuals-instantaneous Mahalanobis score At andEWMAof the Mahalanobis score $\overline{A}t$ from Eqs. (2) and (3). Peaks correspond to false data injection and model attacks which constitute complementary detections of the gateway-a correctly signed-but-impossible observation will elevate A_t still. Finally, the fourth panel shows the combined SLA condition $S(t)$ as in Eq. (4) between a restricted and contained band as shown as dotted lines in Table 4. It stays below during calibration; rises into restricted and contained conditions based on an escalating of either risk or residual signals. Together, the reader of four panels depicts the functional interaction of the loop: an anomaly first shows up on either $Ri(t)$ or At and $S(t)$ instead of statically fixed on the SLA state changes the service policy.

Deterministic gateway evidence and physics-based residual evidence respond to their corresponding attacks much earlier before converging into the SLA state (Figure 2). Such complementing nature favors the utility of employing both scores in the cross-layer decision loop.

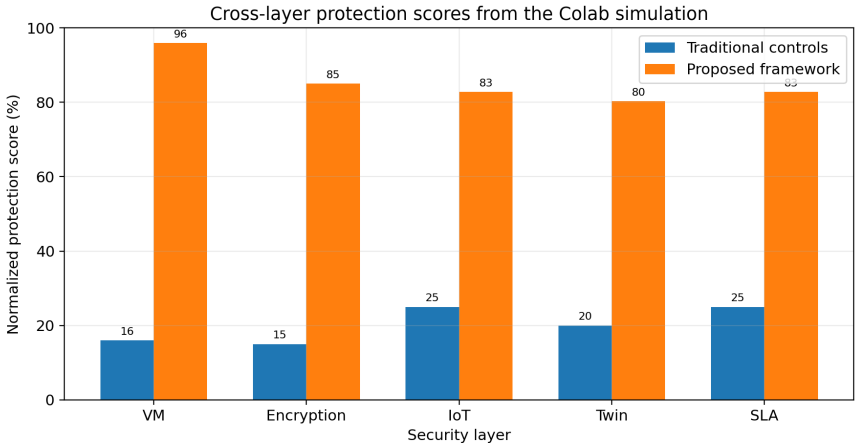

Figure 3: *The normalized protection scores derived from ablation. Control compared to our loop on VM, encryption, IoT, twin, and SLA layers.*

5.2. Layer Comparison

This comparison, summarized in Fig. 3 in terms of scaled protection evidence (from the same 4000-step run, and all normalized the same way), examines VM, encryption, IoT, twin, and SLA layers respectively. The notebook that generates Fig. 3 simply keeps counts of times each appropriate control firing during attack and silence during each benign window, and then scales each count to be a percentage out of 100. The summed bars do not represent a new experimental run: they confirm whether each layer of Fig. 1 is actually working. The standard controls (left bar in each pair) simply allow each message, do not prove the control release of the keys, and do not maintain an adaptive SLA; as a result they performed 16% for VM, 15% for encryption, 25% for IoT, 20% for twin and 25% for SLA. Each corresponding step of the proposed loop(right bar in each pair) performs 96%, 85%, 83%, 80% and 83%, respectively. All layer's performed substantially better by their increased performance at proving control release or more efficient adaptive system policies. The VM layer's larger increase is caused by attestable keyed release lacking from the controls; VM (and IoT, SLA and twin) benefit more by more efficiently combining R i(t), A t , S(t). This therefore demonstrates, on a layer by layer basis whether components of the proposed architecture operate, although it does not prove each corresponding bar independently is an independent system test. The proposed loop performs with better normalized protection on each of the 5 layers, with best comparison over VM layer (Figure 3). This is significant that the positive differences are performed over the entire architecture not just a certain layer.

5.3. Virtualization Attack Success

The virtualization experiment isolates a single aspect of the architecture-release of decryption key from cloud workload, as shown in Figure 4. In the experiment as implemented, a compromised guest repeatedly requests the session key that's used to unwrap telemetry. The baseline bar is the non-protected, traditional scenario: where knowing of the existence of a running VM is considered enough authorization, so that 84.0% of these attempts work. The protected bar is the attested one: the hypervisor first verifies the image's integrity and then released the key. That percent drops to 4.0%; because the simulator allows for a 8.0% rate of false positives when the image is already comprised, that is not evidence of a perfect oracle of attestation, rather of an imperfect one. The y-axis measures the level of success, so the shorter the bar-the better the results. The point of the Figure is simply two points determined on a binary choice-to release or not-during the VM key-release window described in Table 5. Attestation-gated key release decreases Simulated attack success by 84% to 4%(Figure 4). Workload integrity therefore alone stands as the cause that an compromised guest can no longer gain material needed for decryption simply by remaining up.

5.4. Adaptive SLA Latency

The latency experiment is a test to see if a security-aware SLA will be capable of binding the response time within its constraints under increasing load which is shown in Figure 5. The code implemented here does not communicate to a commercial cloud but simulate the queueing model where the arrival rate is the amount of parallel users and the service rate is dictated by the SLA. The static profile uses a fixed capacity for any security state causing the wait to increase proportionally. The dynamic profile gets the value from equation 4 and can increase the number of working nodes in the system when the current state is aggravated or constricted as the relationship in table 4. Both are tested for 100, 300, 500, 700, 900 parallel users. The static SLA is at 30 ms and dynamic at 21 ms for 100 parallel users. For 500 it is 125 ms and 61 ms while for 900 it is 221 ms and 105 ms correspondingly. The working of this Figure should then be read as: x-axis is the load on the server while y-axis shows simulated response time. Here one can see the space between the

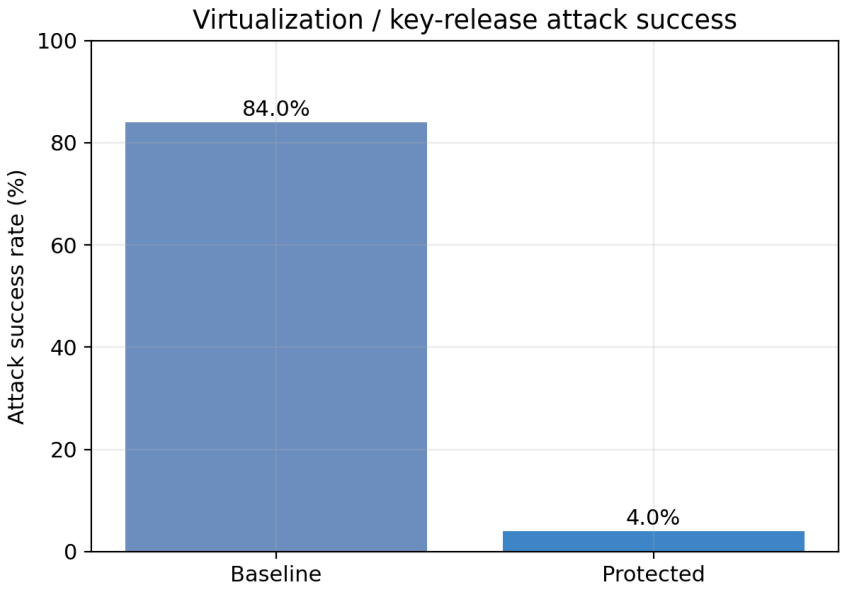

Figure 4: *Virtualization and the rate of success for the key-release attack (implicated:implementation and measured using): baseline rate:* 84.0% *and attested key release rate:* 4.0%.

two lines being the gain in using dynamic workload management instead of static contract as SLA. it proved the bounded performance degradation by the overload which means it is not a simulation run on a live service that would autoscale.

5.5. Hybrid Encryption Micro-Benchmark

The cost of wrapping is measured under the same code execution time to give the encryption layer in Figure 1 some concrete timing, rather than just a design point. Each bar represents the average time it takes to protect a single 64KiB batch of telemetry data. Symmetric-only authenticates the entire 64KiB batch using HMAC-SHA-256, takes 0.59ms. Asymmetric-only takes a short, mod-pow placeholder that represents its public key usage to authenticate the entire 64 KiB batch, and takes 0.25ms; it is cheaper because it uses a hash of only a small length in this micro-benchmark. Hybrid only needs to run both schemes and it is almost exactly the same as purely symmetric, and 0.63ms; because the cost of the slow public key encryption is amortized by the enormous size of the batch it is barely greater than fully symmetric. Thus, the functioning of the graph is a test of the three types of intended work-bulk integrity done by the fast symmetric scheme and id and key-establishment done by a relatively slow public key scheme- not any claim about any production cipher suite and no claim at all on network round-trip delays. During this micro-benchmark, the cost ofhybrid wrapping is hardly greater than the cost of the symmetric full-batch operation(Figure 6), which seems to lend support to separating bulk protection from the session key establishment, without doing the asymmetric processing for all telemetry data.

5.6. Interpretation

Figure 2 to 7 prove together that the executed loop follows the indications from section 3. The deterministic check of the gateway triggers $Ri(t)$ against spoofing or replay, the twin residual triggers At against faultiness/tampering with model; the attestation reduces success to key release and $S(t)$ switches policy under workload. The limitation of any form of check is being unable to find impossible, yet authentic value; and of the twin being unable, by itself, to safe ward a stolen hypervisor, on its own, but as demonstrated: that co-ordination is the achieved value is measured. Currently there is not yet a support for public ICS traces, measured on motes energy or multi-cloud attestation portability. These will be the next steps of validation [23], [29], [30]. These experimental results represent the internal behavior of the developed framework and demarcate the bound of the evidence: The speedups can be replicated within the simulator, yet still need to be validated on public traces and within the cloud. As such, the conclusion separates the evidence which has been delivered, and that which needs to be proved-outside the simulator.

6. Conclusion and Future Work

A layered security framework for IoT with DT embedded is then presented and simulated. The device risk $Ri(t)$, twin remainder At, attested virtualization, hybrid-encryption wrap and SLA adaptability status $S(t)$ works as an entire security control loop. It is shown in the simulation that by using attested key release, the rate of successful virtualization attack reduces from 84.0% to 4.0%. In the dynamic SLA policy, it took 105 ms at 900 simultaneous users for recovery time to respond whereas static SLA approach took 221 ms at 900 simultaneous users. The protection score also increases linearly

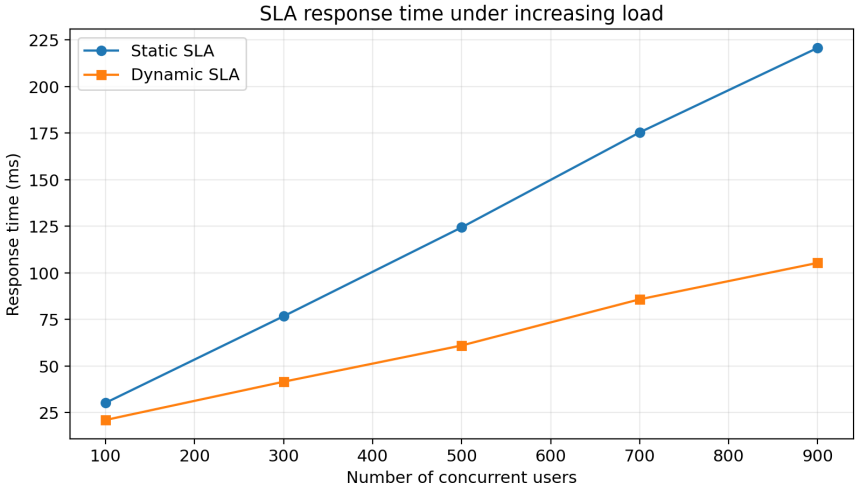

Figure 5: *SLA response time under varying load during actual implemented test execution. Static SLA vs. Dynamic SLA at 100, 300, 500, 700 and 900 parallel users.*

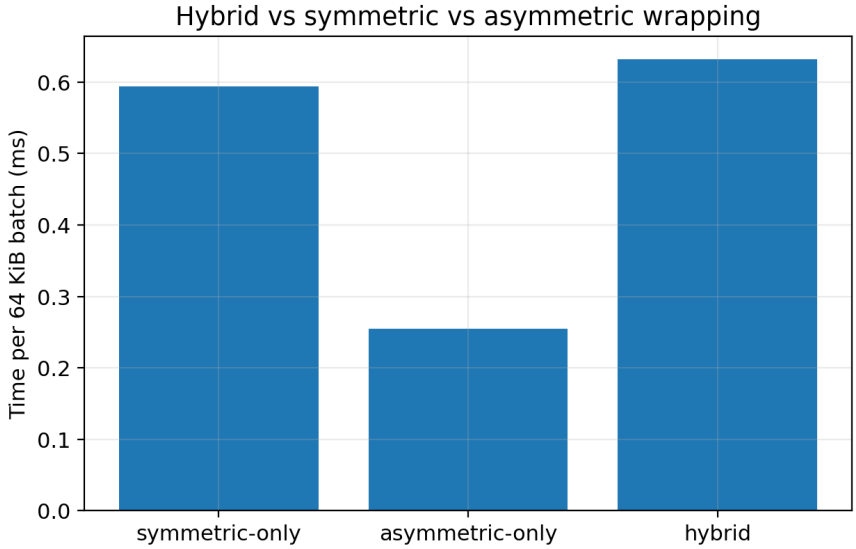

Figure 6: *Time for 64 KiB batch on symmetric-only, asymmetric-only and hybrid wrapping on implemented micro-benchmark.*

from device to cloud and so do the advantages over traditional methods.Four direct practical implications can be drawn from the results of this paper. (1) Device's telemetry must not be solely validated based on knowing the credentials. (2) Release of sensitive key should rely on the workload integrity, as attested; (3) Critical operations required by a certain workload should have a stricter authority level compare with read-only telemetry; (4) SLAs must define concrete security indications and corresponding performance indicators such that dynamic security mechanism can respond accordingly.

Further implementation could involve testing this architecture with actual industrial and network security datasets, like SWaT traces, WADI traces and CIC-DDoS traces. Additional testing should run multiple times with different seeds, report the confidence intervals, and justify the statistical relevance of results presented here. We also suggest investigating policy transfer between two different Cloud providers for this architecture with SLA definitions and attestation controls. Test across multiple-cloud and real-world operational environment will further validate this work.

Author Declarations

Author Contributions (CRediT): Ali Ahmad: Conceptualization, Methodology, System Design, Investigation, Visualization, and Writing - Original Draft. Nauman Irshad Ali Shah: Software, Code implementation, Writing - Original Draft, Writing - Review & Editing. Saqib Dar: Supervision, Testing, Writing - Review & Editing. Danish Ali: Testing, data analysis, validation and investigation. All authors have read and approved the submitted version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The executable code is publicly available in the authors' GitHub repository: aliahmad-gif/A-Security-Framework-for-Cloud-Enabled-IoT-and-Digital-Twin-Systems.

Acknowledgments: The authors would like to thank the Center for Emerging Networks & Technologies (CENT) Lab at the University of Central Punjab (UCP), Lahore, Pakistan, for technical support and constructive feedback.

Conflicts of Interest: The authors declare no conflict of interest.

Ethics Statement: Not applicable.

Declaration of Generative AI: During the preparation of this work, the authors used generative AI tools only for language editing and proofreading, where applicable. The authors reviewed and edited the content and take full responsibility for the published paper.

References

[1] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583-592, 2012, doi: 10.1016/j.future.2010.12.006.

[2] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1-11, 2011, doi: 10.1016/j.jnca.2010.07.006.

[3] W. Jansen and T. Grance, *Guidelines on Security and Privacy in Public Cloud Computing*, NIST Special Publication 800-144, National Institute of Standards and Technology, 2011, doi: 10.6028/NIST.SP.800-144.

[4] M. Fagan, K. Megas, K. Scarfone, and M. Smith, *IoT Device Cybersecurity Capability Core Baseline*, NISTIR 8259A, National Institute of Standards and Technology, 2020, doi: 10.6028/NIST.IR.8259A.

[5] G. Shao and M. Helu, "Framework for a digital twin in manufacturing: Scope and requirements," *Manufacturing Letters*, vol. 24, pp. 105-107, 2020, doi: 10.1016/j.mfglet.2020.04.004.

[6] J. Voas, P. Mell, P. Laplante, and V. Piroumian, *Security and Trust Considerations for Digital Twin Technology*, NIST IR 8356, National Institute of Standards and Technology, 2025, doi: 10.6028/NIST.IR.8356.

[7] E. C. Balta, M. Pease, J. Moyne, K. Barton, and D. M. Tilbury, "Digital twin-based cyber-attack detection framework for cyber-physical manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1695-1712, 2024, doi: 10.1109/TASE.2023.3243147.

[8] R. Barnes, K. Bhargavan, B. Lipp, and C. Wood, "Hybrid Public Key Encryption," RFC 9180, Internet Research Task Force, February 2022, doi: 10.17487/RFC9180.

[9] European Union Agency for Cybersecurity, *ENISA Threat Landscape 2025*, October 2025, doi: 10.2824/1946374.

[10] S. Rose, O. Borchert, A. Kerman, M. Souppaya, et al., *Implementing a Zero Trust Architecture: High-Level Document*, NIST Special Publication 1800-35, National Institute of Standards and Technology, 2025, doi: 10.6028/NIST.SP.1800-35.

[11] National Institute of Standards and Technology, *Considerations for Achieving Crypto Agility: Strategies and Practices*, NIST Cybersecurity White Paper 39, December 2025, doi: 10.6028/NIST.CSWP.39.

[12] V. C. Hu, *Security Property Verification by Transition Model*, NIST IR 8539, National Institute of Standards and Technology, January 2025, doi: 10.6028/NIST.IR.8539.

[13] National Institute of Standards and Technology, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST CSWP 29, February 2024, doi: 10.6028/NIST.CSWP.29.

[14] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data," *Official Journal of the European Union*, 2016.

[15] U.S. Department of Health and Human Services, "The HIPAA Security Rule," HHS.gov. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/security/index.html.

[16] International Electrotechnical Commission, *IEC 62443-3-3:2013, Industrial Communication Networks - Network and System Security - Part 3-3: System Security Requirements and Security Levels*, IEC, Geneva, Switzerland, 2013.

[17] European Parliament and Council of the European Union, "Regulation (EU) 2024/2847 on horizontal cybersecurity requirements for products with digital elements (Cyber Resilience Act)," *Official Journal of the European Union*, November 20, 2024.

[18] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1475-1503, 2022, doi: 10.1109/COMST.2022.3171465.

[19] H. Mun, K. Han, E. Damiani, H. K. Yeun, T.-Y. Kim, L. Martino, and C. Y. Yeun, "A comprehensive survey on digital twin: Focusing on security threats and requirements," *IEEE Access*, vol. 13, pp. 73362-73390, 2025, doi: 10.1109/ACCESS.2025.3563621.

[20] A. R. Qureshi, A. Asensio, M. Imran, J. Garcia, and X. Masip-Bruin, "A survey on security enhancing Digital Twins: Models, applications and tools," *Computer Communications*, vol. 238, art. 108158, 2025, doi: 10.1016/j.comcom.2025.108158.

[21] C. Alcaraz and J. Lopez, "Digital twin security: A perspective on efforts from standardization bodies," *IEEE Security & Privacy*, vol. 23, no. 1, pp. 83-90, 2025, doi: 10.1109/MSEC.2024.3504193.

[22] K. E. Kampourakis, V. Gkioulos, and S. Katsikas, "Cybersecurity digital twins for industrial systems: From literature synthesis to framework design," *Information*, vol. 17, no. 3, art. 286, 2026, doi: 10.3390/info17030286.

[23] P. J. Sinijoy, M. Bhasi, and V. R. Renjith, "Digital twin-based multilevel attack detection and automotive preventing scheme using machine learning and deep learning approaches," *Expert Systems with Applications*, vol. 302, art. 130647, 2026, doi: 10.1016/j.eswa.2025.130647.

[24] P. Nguyen, A. Rauniyar, J. Bartel, J. Laufer, C. Dalamagkas, and K. Pohl, "AI-driven digital twin-based security orchestration, automation and response for critical infrastructures," *Automated Software Engineering*, vol. 33, art. 61, 2026, doi: 10.1007/s10515-026-00612-1.

[25] F. Baiardi, S. Ruggieri, and V. Sammartino, "A Security Twin to Defeat Intrusions in Cyber Physical Systems," in *Proceedings of the 35th European Safety and Reliability Conference (ESREL 2025) and the 33rd Society for Risk Analysis Europe Conference (SRA-E 2025)*, Stavanger, Norway, 2025, pp. 643-650, doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P7829-cd.

[26] M. Repetto, "Cybersecurity Digital Twins: Concept, blueprint, and challenges for multi-ownership digital service chains," *Journal of Information Security and Applications*, vol. 96, art. 104299, 2026, doi: 10.1016/j.jisa.2025.104299.

[27] S. D. Okegbile and I. P. Gambo, "Artificial intelligence-driven security framework for internet of things-enhanced digital twin networks," *Internet of Things*, vol. 31, art. 101564, 2025, doi: 10.1016/j.iot.2025.101564.

[28] W. Issa, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "DT-BFL: Digital Twins for Blockchain-enabled Federated Learning in Internet of Things networks," *Ad Hoc Networks*, vol. 178, art. 103934, 2025, doi: 10.1016/j.adhoc.2025.103934.

[29] A. Sayghe, "Digital Twin-Driven Intrusion Detection for Industrial SCADA: A Cyber-Physical Case Study," *Sensors*, vol. 25, no. 16, art. 4963, 2025, doi: 10.3390/s25164963.

[30] M. M. Salim, D. Camacho, and J. H. Park, "Digital Twin and federated learning enabled cyberthreat detection system for IoT networks," *Future Generation Computer Systems*, vol. 161, pp. 701-713, 2024, doi: 10.1016/j.future.2024.07.017.

[31] International Organization for Standardization, *ISO 23247-1:2021, Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 1: Overview and General Principles*, ISO, Geneva, Switzerland, 2021.

[32] M. Whaiduzzaman, N. T. Monalisa, S. T. Himi, S. Sultana, T. Jan, and A. Barros, "Enhancing IIoT security using digital twins in Industry 5.0: A systematic literature review," *Information*, vol. 17, no. 2, art. 209, 2026, doi: 10.3390/info17020209.