Electronics and Information Technologies, 2026, 35, 161–174
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.056:004.7:621.3

METHODS FOR FAULT-TOLERANT AND CYBER-RESILIENT CLOUD-ORIENTED IoT PLATFORMS UNDER EMBEDDED DEVICE CONSTRAINTS

Volodymyr Voitovych**, *Roman Lys
Department of Sensor and Semiconductor Electronics,
Ivan Franko National University of Lviv,
107 Gen. Tarnavskoho Stryye, Lviv, 79017, UKRAINE

Voitovych, V., Lys, R. (2026). Methods for Fault-Tolerant and Cyber-Resilient Cloud-Oriented IoT Platforms under Embedded Device Constraints. *Electronics and Information Technologies*, 35, 161–174. https://doi.org/10.30970/eli.35.12

ABSTRACT

Background. Cloud-oriented Internet of Things (IoT) platforms integrate embedded sensing/actuation devices, edge nodes, and cloud services. Within these distributed cyber-physical architectures, cyber-attacks and operational faults frequently co-propagate, critically degrading availability, data integrity, and physical safety. Achieving fault tolerance and cyber resilience under severe embedded constraints remains a fundamental scientific challenge.

Materials and Methods. This review targets device–edge–cloud deployments constrained by limited energy, computational capacity, memory, and intermittent connectivity. It consolidates systems engineering guidance from ISO/IEC 30141, NIST SP 800-160, ETSI EN 303 645, NISTIR 8259A, and secure update frameworks (e.g., SUIT and TUF). We construct a joint taxonomy of attacks and failures, systematize resilience patterns for identity, observability, and recovery, and integrate current research on sensor degradation monitoring. The theoretical framework introduces formal mathematical metrics for resilience loss, Markovian state transitions, and dynamic evidence budgeting.

Results. The review synthesizes a generalized cyber-resilience model in which fault tolerance and cybersecurity are treated as a single adaptive observe-detect-decide-recover loop. The model is derived from the reviewed literature and combines hierarchical instrumentation, provenance-aware lifecycle control, evidence budgeting, risk-adaptive attestation, and latency-aware edge orchestration. Key gaps identified include the lack of unified metrics linking cybersecurity and dependability properties, fleet-scale attestation challenges, and resource-aware detection on constrained microcontroller endpoints**.**

Conclusion. Fault-tolerant cyber-resilience should be considered as an adaptive observe-detect-decide-recover control loop optimized for embedded resource bounds. The review summarizes a literature-based analytical model that can serve as a basis for future experimental validation of risk-adaptive remote attestation and locality-aware orchestration in constrained IoT environments**.**

Keywords: cyber resilience, fault tolerance, cloud-oriented IoT, edge computing, secure updates, self-healing.

INTRODUCTION

Cloud-oriented Internet of Things (IoT) platforms are increasingly deployed as complex, large-scale distributed cyber-physical systems (CPS). These sophisticated architectures systematically combine resource-constrained embedded devices, edge computing gateways, and highly scalable cloud infrastructure for continuous telemetry ingestion, edge-based analytics, and mission-critical remote actuation. Within such interconnected platforms,

© 2026 Volodymyr Voitovych & Roman Lys. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

classical hardware dependability and modern cybersecurity are functionally inseparable. Operational faults — such as network partitions, memory buffer overloads, or benign software defects — inevitably create exploitation windows for malicious actors. Conversely, active cyber-attacks, including credential abuse, malicious payload injections, and sensor data poisoning, predominantly manifest as availability degradation, anomalous resource consumption, and fundamentally unsafe control behavior [1].

The embedded layer imposes severe and uncompromising boundary conditions on the overall system design. Microcontroller-class endpoints, such as the widely adopted ESP32 (Tensilica Xtensa architecture) or STM32 (ARM Cortex-M series) architectures, operate under strict energy budgets. They frequently rely on limited-capacity batteries or intermittent energy harvesting mechanisms, possessing scarce compute capabilities (often sub-100 MHz clock speeds) and minimal RAM allocations (typically under 512 KB). Furthermore, these nodes are frequently deployed in harsh, inaccessible environmental conditions that induce gradual, stochastic physical hardware degradation. Phenomena such as sensor drift, signal noise amplification, thermal throttling, and connectivity intermittency complicate the differentiation between a cyber-attack and a hardware failure.

These inherent physical and computational limitations heavily restrict the feasibility of deploying complex, heuristic-based endpoint instrumentation (such as machine learning-based intrusion detection systems) natively on the edge device. This necessitates the adoption of hierarchical, multi-tiered architectures where edge and cloud tiers actively facilitate the computational burden of anomaly detection, cryptographic verification, and state recovery processes [2].

International standards and robust engineering frameworks provide foundational guidelines for mitigating these distributed risks. ISO/IEC 30141 formalizes architectural decomposition, defining strict interfaces across the IoT domain [3]. Concurrently, ETSI EN 303 645 establishes baseline IoT product security requirements, particularly concerning hardware-backed identity management, secure boot sequences, and over-the-air (OTA) update protocols [4]. From a holistic lifecycle perspective, NIST SP 800-160 advocates for rigorous systems security engineering to verify trustworthy behavior continuously across the operational lifespan of the CPS [5]. Furthermore, remote attestation methodologies demonstrate how resource-rich verifiers can dynamically assess the cryptographic integrity of a device's memory state, effectively outsourcing the verification burden from constrained nodes to local edge gateways [6].

Despite this extensive standardization, existing literature predominantly lacks a cohesive, end-to-end analytical synthesis capable of connecting device-edge-cloud components into a unified fault-tolerant and cyber-resilient platform under embedded constraints. Current works tend to evaluate network security, cryptographic overhead, and hardware availability in isolation. However, operational resilience requires integrated reasoning regarding the computational, communicational, and energetic costs of threat detection, fault containment, and autonomous recovery, alongside their cascading physical safety implications [7].

This review is motivated by the need for a neutral synthesis of fault-tolerance and cyber-resilience methods applicable to cloud-oriented IoT platforms with heavily resource-limited embedded devices. Its objective is to summarize existing approaches, compare their applicability under energy and communication constraints, and formulate a generalized literature-based model of system cyber-resilience that may be further examined through experimental validation in device-edge-cloud environments.

MATERIALS AND METHODS

System model and state-space formalization

We conceptualize a three-tier operational platform (device–edge–cloud) featuring an explicit logical separation between the control plane (managing device identity, access

policies, configuration, and firmware updates) and the data plane (handling high-frequency telemetry routing and actuation commands). This decoupling is architecturally critical: control-plane incidents possess the potential to compromise entire fleets simultaneously via malicious updates or revoked certificates, whereas data-plane faults typically manifest as gradual, isolated signal quality degradation. This structural model aligns deeply with the established IoT reference architecture role decomposition [3].

Formally, the distributed system state at any continuous time t can be represented as a multidimensional vector space

$$\mathcal{S}(t) = \langle \mathcal{H}(t), \mathcal{N}(t), \mathcal{C}(t) \rangle. \quad (1)$$

Here, $\mathcal{H}(t)$ represents the physical hardware state (including instantaneous energy reserves $E_{bat}(t)$ and sensor calibration drift parameters), $\mathcal{N}(t)$ defines the network topology and dynamic connectivity matrices, and $\mathcal{C}(t)$ encapsulates the cyber state (cryptographic identity validity, firmware hash integrity, and active access control configuration).

Adverse events are mathematically represented by a joint set

$$E_{total} = E_{cyber} \cup E_{fault}, \quad (2)$$

encompassing both intentional cyber-attacks and stochastic operational faults. Fault tolerance is therefore interpreted as the systemic capability to maintain acceptable service continuity and physical safety under E_{total} via coordinated prevention, detection, containment, recovery, and adaptation mechanisms. Correspondingly, cyber resilience is defined as the system's capacity to sustain core mission objectives and execute autonomous recovery sequences post-disruption, requiring explicit cryptographic and physical verification of all self-healing actions within their cyber-physical context [5].

To model the device lifecycle under disruption, we employ a Continuous-Time Markov Chain (CTMC) defining the finite state space of an IoT endpoint:

$$\Phi = \{Normal, Degraded, Compromised, Quarantined, Recovering\}. \quad (3)$$

The transition rate matrix Λ governs the probability of moving between these states based on the injection of E_{total} and the efficacy of the implemented detection algorithms (P_d).

Evidence synthesis and hierarchical methodology

This review adopts a strict mechanism-oriented methodology. To systematically evaluate proposed resilience practices while acknowledging severe embedded limitations, every theoretical mechanism is analyzed along three primary dimensions:

1. *Propagation paths and containment boundaries:* Analyzing how an exploit or fault originating at the microcontroller propagates through edge message brokers to cloud data lakes.
2. *Embedded feasibility constraints:* Defining the strict mathematical upper bounds of continuous monitoring, CPU cycles, and radio transmission capabilities available for security overhead.
3. *Cryptographic evidence artefacts:* Identifying the specific logs, hashes, and nonces necessary for the rigorous verification of automated decisions without requiring human-in-the-loop intervention.

Operationally, the platform must treat these dimensions as configurable parameters within an adaptive observe–detect–decide–recover control loop. This continuous feedback mechanism ensures that monitoring intensity and recovery aggressiveness adapt

dynamically to real-time risk levels and available energy resources. We further integrate findings on digital twin modelling and edge orchestration (KubeEdge) as first-class inputs to these distributed resilience decisions [8, 9]. Secure software lifecycle control and SBOM vulnerability tracking are treated as complementary control-plane inputs [10, 11].

Formal metrics and cost-constraint inequalities

Rigorous evaluations of resilience mechanisms require foundational detection and recovery metrics. The detection probability (P_d) and false positive rate (FPR) are defined classically based on True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN):

$$P_d = \frac{TP}{TP + FN}, \tag{4}$$

$$FPR = \frac{FP}{FP + TN}. \tag{5}$$

System availability and global service continuity (S_{cont}) are derived from the Mean Time Between Failures ($MTBF$) and Mean Time To Recovery ($MTTR$):

$$S_{cont} = \frac{MTBF}{MTBF + MTTR}. \tag{6}$$

However, for deeply embedded systems, executing detection algorithms or transmitting cryptographic proofs must strictly adhere to embedded cost functions (C_{dev}). This encompasses absolute energy consumption (E_{cons} in Joules), CPU utilisation time (CPU_{util} in milliseconds), and memory allocation (RAM_{util} in Bytes). This is formulated as a weighted constraint inequality that must be satisfied at all times t:

$$C_{dev}(t) = w_e E_{cons}(t) + w_c CPU_{util}(t) + w_m RAM_{util}(t) \leq C_{max}, \tag{7}$$

where w_e, w_c, and w_m are hardware-specific configuration weights and C_{max} is the absolute physical limit of the microcontroller endpoint before brownout or hard fault occurs. Comparative claims in future experimental work must report statistical uncertainty across repeated trials and utilize action-aware evaluation paradigms that link detection latency directly to the overall recovery impact [7]. A compact classification of representative events across the device–edge–cloud continuum is summarized in **Table 1**.

Table 1. **Compact event taxonomy for device–edge–cloud IoT platforms**

Layer	Representative events	Main risk	Indicative controls
Device	Sensor drift; brownouts; firmware defect	Integrity loss; unsafe actuation	Signal-quality indicators; safe modes
Edge	Gateway overload; compromise; buffer loss	Latency spikes; data manipulation	Rate limiting; isolation; buffering
Cloud	Update compromise; IAM abuse; broker DoS	Fleet-wide compromise; cascading outages	Provenance; staged rollout; redundancy [14]

RESULTS AND DISCUSSION

Generalized cyber-resilience model synthesized from the literature

Based on the reviewed standards and studies, cyber-resilience of a cloud-oriented IoT platform can be summarized as a layered control model that links the device, edge, and cloud tiers through an observe-detect-decide-recover loop. At the device tier, the model relies on minimal telemetry, firmware integrity evidence, energy-state reporting, and safe local modes. At the edge tier, it aggregates evidence, performs correlation, controls traffic containment, and executes attestation or recovery decisions. At the cloud tier, it supports provenance tracking, update governance, SBOM-based vulnerability triage, long-term analytics, and policy coordination.

This model should be understood as an analytical synthesis derived from the reviewed literature, not as a fully validated experimental algorithm. Its main value is the integration of cyber-resilience and fault-tolerance elements that are often considered separately: resilience loss metrics, Markovian state transitions, evidence budgeting, remote attestation, secure update mechanisms, and edge queuing constraints. The synthesized model can therefore serve as a structured basis for future experimental verification rather than as a standalone empirical result.

Threat and fault surfaces in device–edge–cloud architectures

To properly engineer resilience, the topology and attack vectors must be explicitly defined.

As illustrated in **Fig. 1**, threat and fault surfaces within cloud-oriented IoT platforms can be categorically classified into four primary domains:

1. Endpoint compromise and physical device tampering (e.g., JTAG interface extraction, malicious flash overwrites, targeted battery exhaustion attacks).
2. Protocol abuse and message broker overload (e.g., MQTT connection storms, CoAP amplification).
3. Edge gateway compromise and container breakout (e.g., privilege escalation within a Docker/containerd runtime at the edge).
4. Cloud control-plane attacks (e.g., update registry poisoning, Identity and Access Management (IAM) credential theft).

Crucially, an identical observable surface may represent both a reliability deficit and a severe security risk. For instance, a sudden spike in packet loss can obscure intentional network-level data poisoning, while a benign firmware defect triggering cyclic hardware watchdogs (reboots) may generate anomalous traffic patterns indistinguishable from a distributed denial-of-service (DDoS) payload [1]. Because architectural boundaries define how failures propagate [3, 5], practical resilience requires dynamic mathematical propagation modelling. A compromised node inflating connection requests can rapidly saturate an edge broker's memory pool, subsequently triggering unwarranted, costly cloud autoscaling — a systemic positive feedback loop that must be severed through explicit cross-tier evidence capture and rigid traffic containment rules.

Resilience patterns for identity, access control, and key lifecycle

In highly distributed architectures, identity and access control serve not merely as preventive perimeters, but as primary, highly effective recovery primitives. Upon the detection of anomalous behavior at the edge tier, rapid cryptographic key rotation and token revocation can halt fault propagation laterally across the network before a definitive, time-consuming root-cause analysis is completed. Standards strongly advocate against universal, hard-coded credentials to ensure that fleet-wide containment and granular, device-specific remediation remain computationally viable [4].

Managing the key lifecycle requires a delicate balance with embedded feasibility constraints. Resource-constrained devices should avoid frequent heavyweight asymmetric

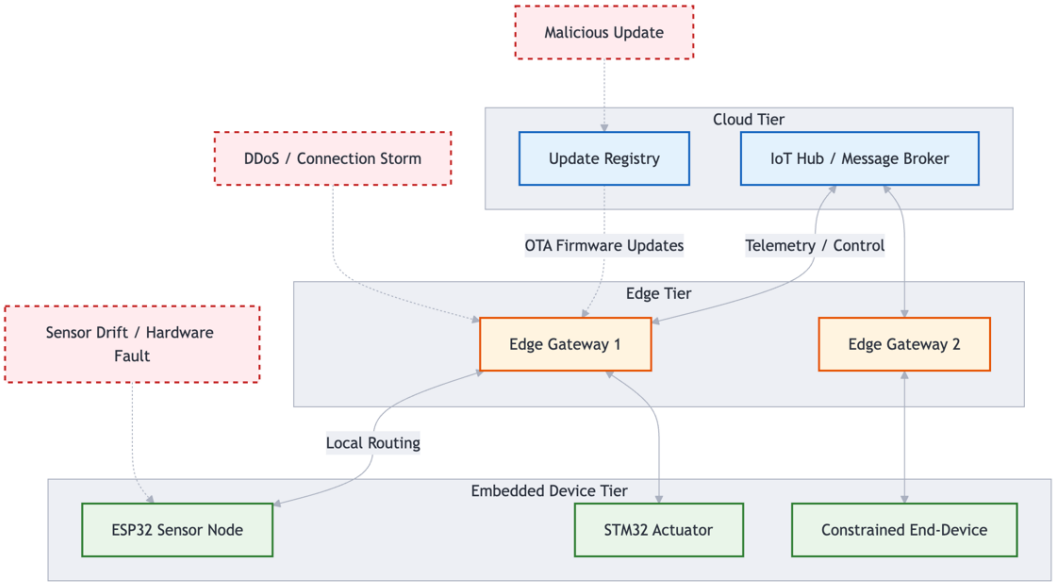


Fig. 1. Three-tier device–edge–cloud architecture illustrating the co-propagation of operational faults (e.g., sensor drift, battery depletion) and cyber-attacks (e.g., firmware modification, protocol injection) across the distributed continuum

cryptographic operations, while edge gateways can perform more complex identity negotiation and verification tasks. Microcontrollers are therefore better suited to minimal secure channels based on optimized symmetric cryptography, whereas event-driven signing, attestation, and re-enrollment are reserved for updates, anomalies, or scheduled audits. Cohort-based policies — grouping devices dynamically by operational criticality, topological proximity, and OTA update phase — further refine identity-driven containment, ensuring that critical recovery actions remain executable even under conditions of severe partial connectivity loss [4, 5, 6].

Update-centric resilience: TUF, SUIT, and Uptane patterns

Firmware update infrastructure constitutes a central point of systemic risk due to its unparalleled capacity to alter fleet-wide behavior simultaneously [12]. A robust design must operate under the pessimistic assumption that the cloud update repository itself may eventually be compromised by an Advanced Persistent Threat (APT). Frameworks such as TUF (The Update Framework) pioneered a compromise-resilience philosophy by demonstrating how cryptographically decoupled update authenticity, metadata signatures, and timestamp roles strictly limit the blast radius of exposed cryptographic keys [13].

In highly constrained environments, SUIT (Software Updates for Internet of Things) introduces compact binary metadata manifests (often CBOR-encoded) and explicit execution rules [14]. This permits microcontrollers to independently verify update integrity, validate dependencies, and assess local hardware conditions (such as ensuring sufficient battery level and memory availability before installation) before committing potentially destructive payloads to non-volatile flash memory. Uptane extends these principles directly into safety-critical operational environments (like automotive CPS), detailing procedures for surviving total repository infiltration [15]. Operationally, secure updates and fault tolerance exhibit deep interdependencies. A flawed but benign update can induce fleet-wide outages strictly analogous to a coordinated botnet attack. Consequently, OTA update rollouts must be orchestrated as tightly controlled, observable mathematical experiments tied directly to measurable telemetry stability indicators and cryptographically verified attestation logs [6, 13].

Observability engineering and dynamic evidence budgeting

Attaining sufficient observability on constrained devices necessitates a strict triage of telemetry signals due to the high energetic cost of continuous wireless transmission, which may dominate the total energy budget of battery-powered IoT nodes. Microcontrollers must therefore aggregate a minimal, high-value evidence set, typically comprising periodic heartbeats, firmware version hashes, energy state vectors, reboot counters, and statistical signal-quality features (variance, skewness). Edge nodes subsequently enrich this sparse data by appending queue latency metrics and executing cross-device temporal correlation, while the cloud tier trains models on long-horizon trends [2, 9]. This adaptive allocation mechanism is illustrated in **Fig. 2**, where evidence collection dynamically shifts between baseline telemetry and anomaly-driven diagnostics under strict energy constraints.

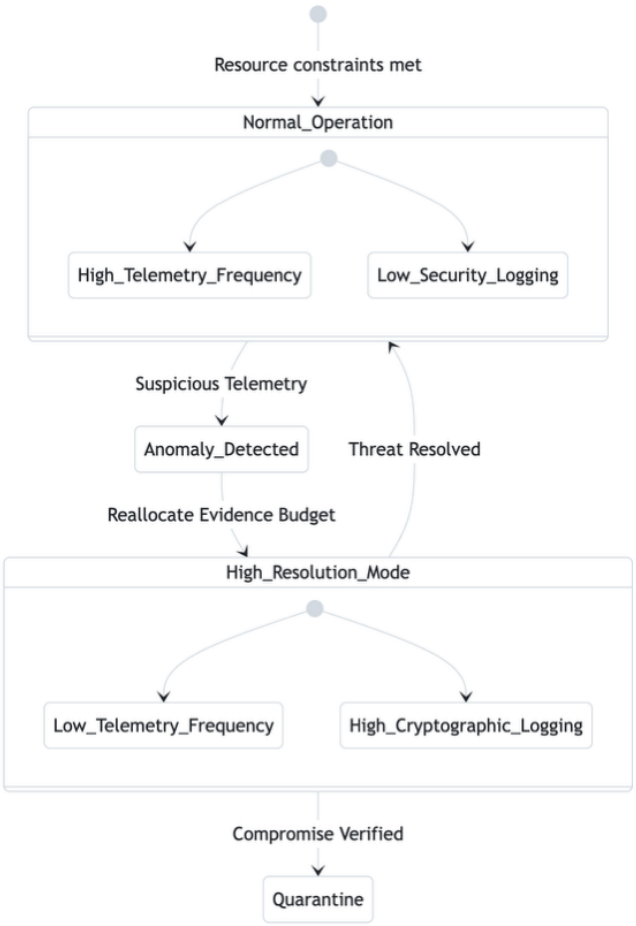


Fig. 2. Energy-aware evidence budgeting mechanism. The diagram demonstrates the dynamic reallocation of transmission frequencies (F_i) between baseline operational telemetry and high-resolution security evidence upon anomaly detection, strictly bounded by the total energy allowance B_{evid}

To prevent the observability apparatus from inducing secondary physical failures (such as rapid battery exhaustion via aggressive polling), telemetry collection must be dynamically adaptive, functioning strictly within a mathematically defined computational "evidence budget" (B_{evid}). Let c_i represent the energy cost per transmission of data type i, and f_i its respective frequency. The system must continuously satisfy the linear inequality:

$$\sum_{i \in \text{tasks}} c_i \cdot f_i \leq B_{evid}. \tag{8}$$

During normal, uninterrupted operation, security risks remain extremely low. However, following a major firmware update or the immediate detection of an anomaly, the edge orchestration policy dynamically reallocates weights to improve diagnostic accuracy, whilst strictly adhering to overall hardware constraints [9].

Detection and decision: evidence fusion and safety constraints

Algorithmic detection in cyber-physical platforms must be explicitly action-aware. A highly sensitive, low-threshold detection algorithm that triggers aggressive network isolation under conditions of mere statistical uncertainty can inadvertently degrade overall system resilience. This occurs by initiating unwarranted hardware reboots, network quarantines, and subsequent energy drain during reconnection handshakes. Therefore, detection efficacy must be quantified by its downstream impact on the system resilience curve rather than isolated, theoretical statistical metrics such as the probability of detection (P_d) and false positive rate (FPR) [7].

To minimize diagnostic ambiguity and reduce false positives, decision engines should implement rigorous evidence fusion. We propose a composite dynamic risk scoring formula $R(t)$ that synthesises real-time behavioural data-plane telemetry (denoted as $\text{Anomaly}(t)$) and historical cryptographic control-plane context (denoted as $\text{Trust}(t)$):

$$R(t) = \alpha \cdot \big(1 - \text{Trust}(t)\big) + \beta \cdot \text{Anomaly}(t) + \gamma \cdot \text{Degradation}(t), \tag{9}$$

where α, β, and γ are tuning coefficients reflecting environmental severity, and $\text{Degradation}(t)$ captures the hardware wear (e.g., sensor drift). Furthermore, automated recovery actions in classical IT systems (such as a hard reboot) may prove catastrophic in a CPS if they unexpectedly disable critical physical actuators. Consequently, self-healing algorithms must be bound by non-negotiable physical safety invariants, employing mathematical hysteresis loops and mandatory cooldown intervals to prevent damaging operational state oscillations [5, 7].

Edge orchestration and queuing latency implications

While edge orchestration platforms enable crucial localized processing during wide-area cloud partitions, their internal computational scheduling policies critically impact systemic resilience. Empirical evaluations of the KubeEdge framework indicate that routing traffic between disparately located edge nodes severely degrades data throughput and exponentially exacerbates response latency as inter-node physical distances increase [9].

We can model the edge message broker as an M/M/1 queuing system. If λ is the arrival rate of MQTT telemetry and μ is the service rate of the edge node, the average response time T is given by:

$$T = \frac{1}{\mu - \lambda}. \tag{10}$$

During an attack or a fault, the arrival rate λ may increase significantly. If λ approaches μ, the response time tends to infinity. These findings yield a vital resilience insight: pursuing "high availability" purely through topological distribution (sending data to a backup edge node) may prove illusory if the resulting network latency destabilizes local detection and physical control loops. Delayed broker responses will instigate client reconnection storms,

further amplify network load (λ) and mutate a localized software fault into a cascading, regional outage. Future research must, therefore, integrate joint orchestration-security policies, where pod placement and workload migration are determined not solely by CPU availability, but by the cryptographically attested trust state and the real-time risk scoring of the target node [6, 9].

Remote attestation: taxonomy and recovery integration

Remote attestation (RA) mechanisms offer varying trade-offs regarding measurement coverage, network scalability, and embedded resource expenditure. Rather than a singular monolithic protocol, RA typically operates across three distinct operational modes within a cloud-oriented platform [6]:

1. *Rollout verification:* Cryptographically validating the exact memory state immediately following an OTA firmware update.
2. *Anomaly response:* Triggering an on-demand, deep memory measurement when the edge gateway observes suspicious behavioral telemetry (i.e., when $R(t)$ exceeds a predefined threshold).
3. *Periodic auditing:* Executing low-frequency, randomized scheduled attestations to maintain a mathematical baseline of fleet assurance.

The general workflow of remote attestation in a constrained IoT environment is presented in **Fig. 3**.

RA yields maximum utility when directly coupled with the automated recovery loop. For instance, a network-quarantined node may only resume telemetry transmission post-attestation; similarly, rollback commands are only deemed successful once the restored

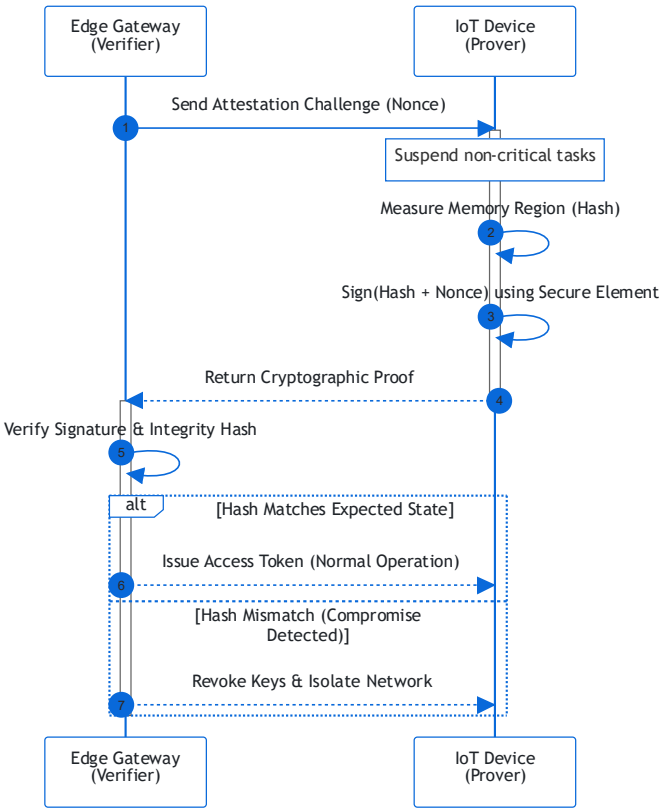


Fig. 3. Remote Attestation workflow. The sequence diagram illustrates the cryptographic challenge-response protocol between a resource-rich edge verifier and a constrained IoT prover

firmware hash is independently verified by the edge verifier. To conserve embedded energy, RA execution frequency must scale dynamically with the device's assigned mission criticality and its real-time anomaly score [6]. This adaptive attestation strategy enables a balance between assurance and resource efficiency in large-scale IoT deployments.

Supply-chain transparency and SBOM-guided recovery

Software Bill of Materials (SBOM) standards facilitate the precise, automated mapping of publicly disclosed vulnerabilities to specific active device fleets in the field. Although SBOM adoption across the software industry is accelerating, integration into continuous operational IoT environments remains a profound challenge due to device heterogeneity [11, 16].

Within the IoT context, SBOMs transition from static compliance documents to active, dynamic resilience tools when integrated directly into the control-plane decision matrix. Upon the disclosure of a critical Common Vulnerability and Exposure (CVE), an SBOM-aware engine can execute an automated graph-based impact assessment, immediately isolating affected cohorts and calculating the least disruptive remediation strategy (e.g., targeted memory patching versus temporary feature degradation). To function effectively at scale, these dependency graphs must be immutably linked to cryptographically signed device provenance records and live inventory registries [10, 11].

Digital twins and reproducible scenario suites

Digital twin methodologies provide indispensable, high-fidelity modelling environments for evaluating complex, non-linear resilience policies at scale without jeopardizing physical production systems [8]. In resilience engineering, digital twins are optimally utilized to execute vast, parameterized scenario suites, precisely calibrating detection thresholds (e.g., α, β, γ weights) and recovery delays before physical firmware deployment.

To ensure strict scientific rigor, these experimental suites must encompass a wide taxonomy of faults (e.g., precise network partitioning, memory starvation, sensor drift) alongside adversarial actions (e.g., credential hijacking, update repository poisoning). By structuring these scenarios through version-controlled injection scripts and explicit success criteria, researchers can achieve a hybrid evaluation model: leveraging the cloud simulation for vast statistical scaling, while utilizing the physical ESP32/STM32 testbed to capture empirical energy boundaries and network latency constraints, thereby continuously refining the fidelity of the digital twin [8].

Experimental protocol and resilience loss formalization

Establishing a rigorous mathematical and experimental protocol is foundational for validating the proposed dissertation methodology. The physical testbed topology, baseline firmware configurations, and multi-tier instrumentation strategies must be explicitly defined before execution.

A standard experimental trial should comprise distinct temporal phases: steady-state warm-up, precise fault/attack injection at time t_e, systemic response/detection at t_{det}, and verifiable recovery completed at t_{rec}. System performance is quantified not merely by timestamps, but by calculating the explicit "Resilience Loss" integral (R_{loss}). If Q_{target} is the targeted quality of service (e.g., 100% telemetry throughput) and $Q(t)$ is the actual measured quality over time, the total loss of the cyber-physical system is defined as:

$$R_{\text{loss}} = \int_{t_{\text{e}}}^{t_{\text{rec}}} \left[Q_{\text{target}} - Q(t)\right] dt. \tag{11}$$

Minimizing R_{loss} under the constraint $C_{\text{dev}} \leq C_{\text{max}}$ can be treated as a candidate optimization objective for future empirical studies. The temporal behavior of system

degradation and recovery, including the definition of resilience loss, is illustrated in **Fig. 4**. Given the inherent timing variability in wireless network stacks (e.g., Wi-Fi, LoRaWAN) and edge orchestrators, robust evaluation requires 10 to 30 repetitions per policy configuration. Results should be supported by 95% confidence intervals and non-parametric statistical testing (e.g., Mann-Whitney U test) to validate the practical significance of reviewed or derived self-healing policies against standard baseline configurations.

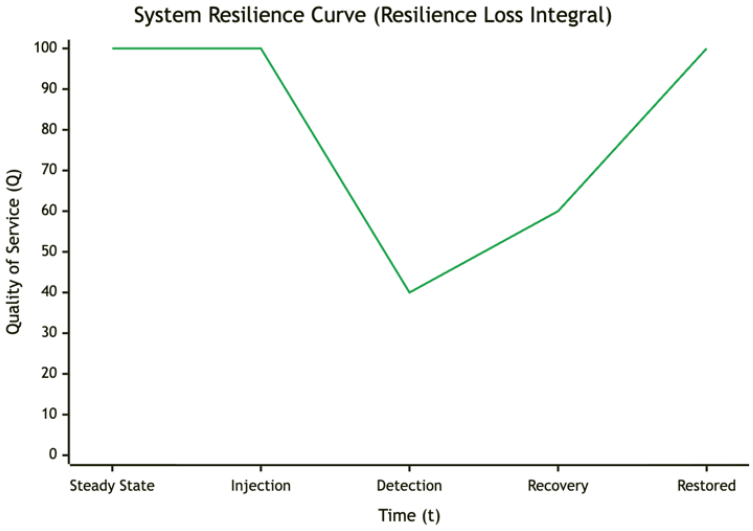


Fig. 4. System resilience curve showing service continuity degradation and recovery. The area above the dip represents the "Resilience Loss" integral ($\mathrm{R}_{\mathrm{loss}}$), captured between the point of fault injection (t_e) and full system recovery (t_{rec})

Summary of open problems and dissertation-aligned research agenda

Significant research gaps remain in the scientific literature regarding the development of unified resilience metrics that mathematically bridge cybersecurity (integrity, confidentiality) and classical dependability (reliability, safety). Furthermore, the scaling of risk-adaptive remote attestation and the execution of cryptographically verifiable firmware rollbacks under severe intermittent connectivity require significant architectural refinement.

Future research directions include the development of unified resilience metrics, reproducible fault and attack injection scenarios, risk-adaptive attestation strategies, and ESP32/STM32-KubeEdge testbed configurations for empirical validation. Such validation should determine whether the generalized model synthesized in this review can reduce resilience loss while respecting the computational, memory, communication, and energy constraints of the embedded domain.

CONCLUSIONS

This comprehensive review shows that fault tolerance in cloud-oriented IoT architectures is closely linked to cyber-resilience, since adversarial attacks and stochastic hardware faults can co-propagate across the distributed device, edge, and cloud continuum. Attaining practical operational resilience requires a shift from static, perimeter-based network defenses toward hierarchical evidence collection, update-centric lifecycle control, and automated self-healing mechanisms that respect established boundaries for embedded resource consumption (C_{max}).

We mathematically establish that remote attestation dramatically strengthens overall system recovery by supplying immutable, cryptographic integrity evidence to algorithmic decision engines [6]. Simultaneously, edge orchestration policies directly dictate the

queuing latency (T) – and therefore the dynamic stability – of physical containment loops [9]. Furthermore, integrating active SBOMs and strict provenance tracking facilitates rapid, fleet-scale vulnerability triaging, enabling significantly less disruptive remediation protocols [10, 11]. The subsequent phase of this ongoing dissertation research will focus directly on formalizing these risk-adaptive mathematical algorithms, minimizing the integral under energy constraints, and empirically validating the framework within a heavily constrained, multi-tier physical testbed.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [V.V., R.L.]; methodology, [V.V.]; validation, [V.V., R.L.]; writing – original draft preparation, [V.V.]; writing – review and editing, [V.V., R.L.]; visualization, [V.V.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog computing and IoT security: A survey and analysis. *Future Generation Computer Systems*, *78*, 680–698. https://doi.org/10.1016/j.future.2016.11.009

[2] Gkonis, P., Giannopoulos, A., Trakadas, P., et al. (2023). A Survey on IoT-Edge-Cloud Continuum Systems: Status, Challenges, Use Cases and Open Issues. *Future Internet*, *15*(12), 383. https://doi.org/10.3390/fi15120383

[3] ISO/IEC. (2024). ISO/IEC 30141:2024 Internet of Things (IoT) – Reference Architecture. https://cdn.standards.iteh.ai/samples/88800/e545dddaf5474251a0fb9af0e2109b9e/ISO-IEC-30141-2024.pdf

[4] ETSI. (2024). ETSI EN 303 645 V3.1.3: Cyber Security for Consumer Internet of Things: Baseline Requirements. https://www.etsi.org/deliver/etsi_en/303600_303699/303645/03.01.03_60/en_303645v030103p.pdf

[5] Ross, R. S., McEvilley, M., & Oren, J. C. (2018). Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems [including updates as of 1-03-2018]. https://doi.org/10.6028/NIST.SP.800-160v1

[6] Kuang, B., Fu, A., Susilo, W., Yu, S., & Gao, Y. (2022). A survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects. *Computers & Security*, *112*, 102498. https://doi.org/10.1016/j.cose.2021.102498

[7] Lezzi, M., Corallo, A., Lazol, M, Nimis, A. (2025). Measuring cyber resilience in industrial IoT: a systematic literature review. Management Review Quarterly. https://doi.org/10.1007/s11301-025-00495-8

[8] Epiphaniou, G., Hammoudeh, M., Yuan, G., et al. (2023). Digital twins in cyber effects modelling of IoT/CPS points of low resilience. *Simulation Modelling Practice and Theory*, *125*, 102744. https://doi.org/10.1016/j.simpat.2023.102744

[9] Kim, S.-H., & Kim, T. (2023). Local Scheduling in KubeEdge-Based Edge Computing Environment. *Sensors*, *23*(3), 1522. https://doi.org/10.3390/s23031522

[10] National Institute of Standards and Technology. (2022). Secure Software Development Framework (SSDF) Version 1.1 (NIST SP 800-218). https://doi.org/10.6028/NIST.SP.800-218
[11] Nocera, S., et al. (2025). On the adoption of software bill of materials in open-source software projects. *Journal of Systems and Software*, *230*, 112540. https://doi.org/10.1016/j.jss.2025.112540
[12] Catuogno, L., & Galdi, C. (2023). Secure Firmware Update: Challenges and Solutions. *Cryptography*, *7*(2), 30. https://doi.org/10.3390/cryptography7020030
[13] Samuel, J., Mathewson, N., Cappos, J., & Dingledine, R. (2010). Survivable key compromise in software update systems. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 61–72. https://doi.org/10.1145/1866307.1866315
[14] Moran, B., Tschofenig, H., Brown, D., & Meriac, M. (2021). A Firmware Update Architecture for Internet of Things (RFC 9019). Internet Engineering Task Force. https://doi.org/10.17487/RFC9019
[15] IEEE-ISTO. (2022). IEEE-ISTO 6100.1.0.0 Uptane Standard for Design and Implementation of Secure Update Systems. https://uptane.org/papers/ieee-isto-6100.1.0.0.uptane-standard.pdf
[16] Stalnaker, T., Wintersgill, N., Chaparro, O., Di Penta, M., German, D. M., & Poshyvanyk, D. (2024). BOMs Away! Inside the Minds of Stakeholders: A Comprehensive Study of Bills of Materials for Software Systems. *Proceedings of the 46th ACM/IEEE International Conference on Software Engineering*, 1–13. https://doi.org/10.1145/3597503.3623347

МЕТОДИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА КІБЕРСТІЙКОСТІ ХМАРНО-ОРІЄНТОВАНИХ ПЛАТФОРМ ІНТЕРНЕТУ РЕЧЕЙ В УМОВАХ ОБМЕЖЕНЬ ВБУДОВАНИХ ПРИСТРОЇВ

Володимир Войтович**, Роман Лис
Кафедра сенсорної та напівпровідникової електроніки,
Львівський національний університет імені Івана Франка,
вул. Генерала Тарнавського, 107, м. Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. Хмарно-орієнтовані платформи Інтернету речей поєднують вбудовані сенсорні й виконавчі пристрої, периферійні вузли та хмарні сервіси. У таких архітектурах кіберзагрози й експлуатаційні відмови можуть поширюватися одночасно, знижуючи доступність, цілісність даних і фізичну безпеку. Забезпечення відмовостійкості та кіберстійкості за умов обмеженого енергоспоживання, пам'яті, обчислювальних ресурсів і зв'язку залишається актуальною науковою проблемою.

Матеріали та методи. Розглянуто архітектури типу «пристрій–периферійний рівень–хмарний рівень» для ресурсно-обмежених вбудованих систем. Узагальнено підходи системної інженерії, викладені в ISO/IEC 30141, NIST SP 800-160, ETSI EN 303 645, NISTIR 8259A, а також підходи до безпечного оновлення SUIT і TUF. Побудовано таксономію атак і відмов, систематизовано шаблони забезпечення стійкості для керування ідентичністю, моніторингу стану та відновлення працездатності з урахуванням деградації сенсорів. Теоретична частина охоплює метрики втрати стійкості, марковські моделі переходів станів і розподіл ресурсів для перевірки доказів довіри.

Результати. Показано, що практична кіберстійкість забезпечується поєднанням багаторівневого моніторингу, контролю життєвого циклу компонентів з урахуванням їхнього походження та механізмів самовідновлення, які діють за визначеними політиками й перевіряють інваріанти безпеки. Математична формалізація засвідчує, що підвищення доступності лише топологічним розподіленням ресурсів може порушувати контури стримування через затримки черг на периферійному рівні. Виявлено прогалини: нестачу уніфікованих метрик кібербезпеки й надійності, складність атестації пристроїв та обмежене виявлення загроз у мікроконтролерних вузлах.

Висновки. Відмовостійкість і кіберстійкість доцільно розглядати як адаптивний цикл «спостереження–виявлення–прийняття рішень–відновлення», оптимізований з урахуванням обмежень вбудованих систем. Робота формує основу для дослідження ризик-адаптивної віддаленої атестації та локально-орієнтованої оркестрації в середовищі ESP32/STM32–KubeEdge–AWS/Azure.

Ключові слова: кіберстійкість, відмовостійкість, хмарно-орієнтований Інтернет речей, периферійні обчислення, безпечні оновлення, самовідновлення.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
30 April, 2026	08 June, 2026	10 June, 2026	28 September, 2026