Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AI-Driven Cyber-Physical and Supply-Chain Threat Detection in Mobile and Embedded Ecosystems: A Comprehensive Review

Sadeeq Muhammad

sadeeqmauwal@gmail.com

Shobhit University https://orcid.org/0009-0006-4355-4734

Hussain Maharaz Yusuf

Bayero University Kano

Abidina Tijjani Kalarawi

Shobhit University

Sandeep Kumar

Shobhit University

Systematic Review

Keywords: Mobile device security, embedded systems security, cyber‐physical attacks, supply‐chain security, firmware integrity, remote attestation, trusted execution environments, anomaly detection

Posted Date: February 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8968440/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

AI-Driven Cyber-Physical and Supply-Chain Threat Detection in Mobile and Embedded Ecosystems: A Comprehensive Review

Authors: Sadeeq Muhammad[1] · Hussain Maharaz Yusuf[2] · Abidina Tijjani Kalarawi[3] · Sandeep Kumar[4]

Affiliations: [1]School of Computational Sciences and Engineering, Shobhhit Institute of Engineering and Technology · [2]Department of Information Technology, Bayero University Kano · [3]School of Computational Sciences and Engineering, Shobhhit Institute of Engineering and Technology · [4]School of Computational Sciences and Engineering, Shobhhit Institute of Engineering and Technology

Corresponding author: Sadeeq Muhammad (ORCID: https://orcid.org/0009-0006-4355-4734), sadeeqmauwal@gmail.com

Abstract

Mobile and embedded devices—smartphones, wearables, industrial IoT, and vehicles—now underpin critical services but also expand the cyber-physical and supply-chain attack surface. Heterogeneous hardware, baseband stacks, firmware/OS components, and multi-vendor build and distribution pipelines create vulnerabilities across manufacturing, provisioning, deployment, operation, and update. This review consolidates threats and defenses through two coupled lenses: (i) AI/ML-driven monitoring for intrusion, tamper, and anomaly detection, and (ii) device trust foundations (secure/measured boot, trusted execution environments, and remote attestation). Using a systematic, criteria-driven screening and appraisal protocol, we synthesize evidence from peer-reviewed research, standards/specifications, and authoritative platform documentation spanning SUIT-based OTA updates, RATS/EAT and PSA attestation tokens, AVB/dm-verity integrity, and SBOM/SLSA/Sigstore provenance. We contribute (1) a lifecycle taxonomy linking phases, layers, attack classes, and controls; (2) a cross-layer synthesis that connects attestation/TEE mechanisms to telemetry pipelines and remediation workflows; (3) a critical appraisal of datasets and evaluation practices for embedded/IoT intrusion detection; and (4) a gap analysis highlighting fleet-scale attestation, real-time cyber-physical tamper sensing, provenance-to-runtime binding, and baseband security. The result is actionable guidance for building verifiable update pipelines and trustworthy, operationally feasible detection at fleet scale.

Keywords

Mobile device security · embedded systems security · cyber-physical attacks · supply-chain security · firmware integrity · remote attestation · trusted execution environments · anomaly detection

1 Introduction

Mobile and embedded systems—smartphones, wearables, IoT nodes, radios, and industrial controllers—now mediate critical services in every sector. Their ubiquity, heterogeneity, and physical proximity to people and infrastructure make security failures uniquely consequential: compromises do not remain "virtual." They can disable equipment, leak sensitive telemetry, or, in extreme cases, cause physical harm. As these devices become more capable and more connected, the attack surface spans hardware components, firmware stacks, application software, communication protocols, and the supply chain that designs, manufactures, and distributes them.

This review examines cyber-physical and supply-chain threats across the mobile and embedded ecosystem with a particular focus on detection and trust establishment powered by AI and attestation. Rather than isolating risks within a single layer (e.g., app malware or network exploits), we treat the ecosystem as a layered system with interdependencies: a weak update pipeline can undermine a strong boot chain; an unverified sensor can invalidate higher-level behavioral models; a counterfeit component can bypass cryptographic assumptions. Bridging these layers is essential to build resilient defenses that perform under real operational constraints (battery, compute, latency, privacy) and at fleet scale.

Prior surveys often focus on a single slice of this space—general IoT security/privacy/trust [1], [2], ML/IDS methods and datasets for IoT networks [3], [4], [5], [6], BLE protocol security and privacy [7], [8], Android malware and deep-learning-based defenses [9], [10], [11], or cellular security and authentication

analyses [12], [13], [14], [15], [16]. In contrast, this review positions novelty at the *interfaces* between these silos: we connect supply-chain provenance controls (SBOM/SLSA/Sigstore) to runtime integrity (secure boot and attestation) and then to data-driven monitoring and remediation (IDS actions and SUIT-style OTA), while explicitly covering firmware/baseband attack surfaces (e.g., dynamic baseband/firmware analysis) that can invalidate app-layer assumptions [17], [18], [19].

Motivation and scope

- Coverage: smartphones and secure elements; constrained IoT devices and wearables; radios/pagers and specialized communication gear; industrial embedded controllers and gateways.
- Layers: hardware (silicon, sensors, batteries), firmware (bootloaders, kernels), software (apps, services), networks (BLE, Wi-Fi, LTE/5G, LPWAN), and logistics (manufacturing, transit, distribution).
- Lifecycle: manufacturing and provisioning; deployment and operation; maintenance and OTA updates; end-of-life and decommissioning.

Challenges at the cyber-physical boundary

- Integrity of roots of trust: establishing measured/verified boot on diverse silicon, integrating TEEs, and preserving keys against physical side-channels.
- Firmware provenance and update security: ensuring that OTA mechanisms and manifests reflect authentic, untampered images; recovering from failures without bricking devices (SUIT RFC 9124: [20]; NIST SP 800-193: [21]).
- Sensor and battery manipulation: detecting anomalous environmental inputs and power behaviors that can trigger unsafe states or defeat detection models.
- Baseband and protocol exploitation: defending cellular/LPWAN stacks that operate below application firewalls, where weaknesses can enable device-wide compromise (5G survey: [12]; formal 5G-AKA analyses: [14], [15]).
- Supply-chain compromise: preventing counterfeit or modified components from entering fleets; maintaining verifiable provenance (SBOMs, build integrity) from source to device.

Why AI and attestation together?

- Telemetry-rich devices produce behavioral signals (usage, RF patterns, power draw, timing) that AI can learn to model for anomaly detection with limited labels (ML/IDS survey: [3]).
- Remote attestation and device identity frameworks provide cryptographic anchors to trust measurements at run-time (RATS architecture RFC 9334: [22]; PSA token format RFC 9783: [23]).
- Combined, they enable layered assurance: cryptographic trust about what the device is running, and statistical confidence about how it is behaving.

Threat model and assumptions

Although this manuscript is a review (not a single-system proposal), we adopt a consistent threat model to organize evidence and avoid layer-specific blind spots. We assume the defender operates a fleet of heterogeneous mobile/embedded devices and can deploy monitoring and policy enforcement at some aggregation point (device, gateway, MDM/EMM, or cloud).

Adversary capabilities (non-exclusive)

- Supply-chain adversary: can tamper with dependencies, build/release artifacts, mirrors, or vendor firmware distribution (including "valid-looking" updates);
- RF-proximity adversary: can eavesdrop, inject, replay, downgrade, jam, or exploit protocol logic in BLE/LPWAN/cellular in the vicinity of targets;
- Physical-access adversary: can access debug interfaces, attempt fault/side-channel attacks, or extract/modify storage on a subset of devices;
- Fleet-level adversary: can compromise credentials, management infrastructure, or update orchestration to target many devices simultaneously.

Defender and measurement assumptions

- Telemetry visibility is heterogeneous: some fleets can only export coarse metadata (e.g., flow features, update events, attestation tokens) because payload inspection is constrained by encryption, policy, or regulation.
- Monitoring and response must respect operational constraints (battery/CPU/bandwidth) and privacy requirements; therefore, we prefer designs that operate on minimal, purpose-limited features and that can be audited.

- Models and thresholds are exposed to non-stationarity: benign drift (software updates, environment shifts) and adversarial drift (mimicry/evasion) are both assumed possible; for fleets that retrain from telemetry, data poisoning is treated as an explicit risk.

Out of scope: catastrophic silicon fabrication compromise at the foundry level and fully undetectable hardware Trojans are treated as residual risk; we focus on mechanisms that can be deployed and audited in contemporary systems (secure boot, attestation/TEE, OTA hardening, provenance, and telemetry-based detection).

Contributions

1. A unified taxonomy that classifies threats and defenses across device classes, layers, and lifecycle stages, linking supply-chain risks to runtime effects.
2. A synthesis of attacks and detection/defense techniques spanning hardware, firmware, software, networks, and logistics, with emphasis on attestation and AI.
3. An evidence-based gap analysis highlighting limits in fleet-scale attestation, real-time tamper detection, verified update pipelines, and dataset availability.
4. Practical research directions that align attestation, provenance, and AI monitoring to strengthen resilience without imposing untenable operational overhead.

Paper organization

Section 2 describes the review methodology (screening and appraisal). Section 3 defines the device classes and layered ecosystem model. Section 4 categorizes threats and protocol-specific risks with a running example. Section 5 summarizes the cyber-physical attack surface. Section 6 reviews detection techniques and datasets/metrics. Section 7 integrates core defenses (boot integrity, attestation, TEEs, OTA, and provenance) into an operational pipeline. Section 8 covers forensic readiness and evidence collection. Sections 9–11 synthesize gaps, future directions, and conclusions.

Lifecycle overview (Figure)

Fig. 1 provides the lifecycle orientation used throughout the review to connect phases (manufacturing→operation→update) with controls and measurement points.

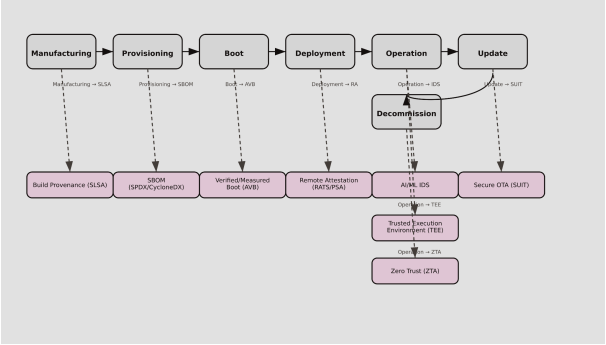


Fig. 1. Lifecycle of mobile and embedded ecosystems with control mappings. Stages (Manufacturing → Provisioning → Boot → Deployment → Operation → Update → Decommission) are connected to controls: AVB (verified/measured boot), RA (remote attestation), TEE (trusted execution environment), SUIT (secure OTA), SLSA (build provenance), SBOM, IDS (AI-driven detection), and Zero Trust patterns.

2 Methodology

This section describes how the reference corpus was assembled and how evidence was appraised and synthesized.

Screening and appraisal protocol

We follow a structured, criteria-driven protocol to identify and screen peer-reviewed works (journals, conferences), authoritative standards/specifications (NIST, IETF, ISO/IEC, TCG), and authoritative platform documentation (AOSP, Arm, Linux kernel). Screening includes title/abstract review and full-text assessment to extract device class, affected layers, lifecycle stage, attack types, detection/defense method, datasets (if any), and key findings.

Search sources and time window

- Sources: IEEE Xplore, ACM Digital Library, Scopus, and Web of Science were used for peer-reviewed discovery; arXiv and Google Scholar were used to identify relevant preprints and to chase citations forward/backward.
- Window: works published between 2012 and 2025 (inclusive) were in-scope, matching the curated corpus span used throughout the review.

Quantitative corpus summary (transparency note): based on the curated reference list cited in this manuscript (not raw database query hits), the current corpus includes 103 numbered sources, spanning 2012–2025 by publication year. Using lightweight parsing/labeling of the References section, 65/103 entries include DOI

metadata and approximately 57/103 correspond to peer-reviewed venues (heuristic keyword match). Source families represented include IETF/RFC (4), NIST (5), platform documentation such as AOSP (4) and Linux kernel docs (1), and standards bodies such as ISO/IEC/IEC (2) and TCG (2). The publication-year distribution is concentrated in 2015–2019 (44) and 2020–2024 (52), with a smaller tail in 2010–2014 (5) and 2025 (2).

Example query (representative)

- ("remote attestation" OR RATS OR EAT OR "PSA attestation" OR "verified boot" OR "Android Verified Boot" OR dm-verity OR "secure OTA" OR SUIT OR SBOM OR SLSA OR Sigstore OR baseband OR "BLE security" OR LoRaWAN) AND (mobile OR embedded OR IoT OR "cyber-physical") AND (detection OR IDS OR anomaly OR forensics OR provenance)

Inclusion criteria

- Focus on mobile/embedded/IoT systems with explicit relevance to (i) cyber-physical or supply-chain threats, (ii) detection/monitoring, and/or (iii) trust foundations (boot/TEE/attestation) and update/provenance mechanisms.
- Provide an actionable technical contribution (empirical evaluation, formal analysis, system design, dataset/tooling, or a normative specification/guideline used in practice).
- English-language and accessible full text.

Exclusion criteria

- General enterprise IT security not tied to mobile/embedded constraints or without device/fleet implications.
- Purely conceptual position papers without technical method or specification.
- Duplicates (same work under multiple venues/URLs) or records lacking sufficient bibliographic metadata for traceability.

Data extraction schema

To keep synthesis consistent across heterogeneous sources, we use a cross-layer schema for (i) controls, (ii) attack classes, and (iii) telemetry/detection signals. Table 1 is used as a compact "coding guide" for mapping results into later sections.

Table 1. Cross-layer mapping of controls, attacks, and detection signals.

Layer	Primary Controls	Typical Attacks	Detection Signals
Hardware	TEE/isolation; keys; anti-tamper	side-channels; fault injection	power/RF signatures; timing anomalies
Firmware	verified/measured boot; provenance	bootloader/kernel compromise	hash/verity checks; attestation claims
Software	least privilege; code signing	malware; RCE; supply-chain injection	app behavior; system calls; logs
Network	auth/crypto; protocol hardening	MITM; replay; downgrade; baseband	flow features; anomalies; formal checks
Logistics	SBOM; SLSA provenance; secure handling	counterfeit; tampering in transit	traceability; signature validation

Quality appraisal (lightweight rubric)

Because the corpus mixes empirical ML/IDS studies, formal protocol analyses, and standards/specifications, we apply a lightweight appraisal rubric during extraction:

- Reproducibility: datasets, parameters, and evaluation setup are described sufficiently to reproduce key claims;
- External validity: evaluation reflects realistic constraints (compute, power, bandwidth, privacy) and deployment placement (device/gateway/cloud);
- Threat-model alignment: adversary capabilities and trust assumptions are explicit and match the system layer being claimed;
- Measurement validity: metrics are appropriate (e.g., false alarm rate and latency for operational IDS) and avoid leakage;
- Actionability: results connect to deployable controls or operational response (e.g., OTA remediation, quarantine).

Reproducibility and artifacts

To support auditability, the review is anchored in a curated reference corpus (a tracking sheet) and script-generated artifacts (e.g., the numeric IEEE-style

bibliography and summary tables). This enables repeatable updates as new sources are added.

Research artifact availability (reproducibility positioning): while we may not be able to publicly release the full tracking sheet (e.g., because it can contain notes, operational context, or non-redistributable metadata), we can share (as supplemental material or via an archival repository) (i) a sanitized, machine-readable list of included sources with stable identifiers (DOI/URL, venue, year), (ii) the search protocol and inclusion/exclusion criteria used to assemble the corpus, (iii) scripts that generate the bibliography/tables and support consistency checks, and (iv) the final diagram sources (SVG figures) used in this manuscript. These artifacts are sufficient for third parties to reproduce the screening logic, regenerate the derived tables/figures, and update the corpus over time.

Limitations

This review synthesizes evidence across heterogeneous sources—peer-reviewed empirical studies, formal analyses, and standards/specifications/vendor documentation—whose goals and validation methods differ. As a result, we do not perform a formal meta-analysis or claim that outcomes (e.g., IDS accuracy) are directly comparable across papers; instead, we emphasize operationally relevant patterns (telemetry feasibility, drift robustness, recovery correctness) and treat standards/specifications as normative anchors rather than empirical proof.

Attestation evidence quality (trust+AI limitation): many of the review's arguments assume that attestation is available as a reliable "state witness," yet in practice the evidentiary value of attestation tokens depends on freshness guarantees (nonce/challenge handling, clock quality, replay resistance, and caching), verifier trust and policy correctness (who operates the verifier, how reference measurements are maintained, and how rollout windows are encoded), and token/measurement semantics stability over time (schema drift across PSA/EAT profiles, vendor-specific claims, and evolving measurement granularity). These factors can induce false assurance (stale-but-valid tokens), inconsistent fleet decisions across verifier versions, or mismatches between what is measured (e.g., boot state) and what the detection model assumes (e.g., runtime configuration). Therefore, attestation should be treated as versioned evidence with explicit freshness windows, verifier policy provenance, and schema/version negotiation, rather than as a binary ground truth.

Counts and summaries in this paper reflect the curated tracking sheet rather than a complete export of all query hits from each database, and the "peer-reviewed" subset depends on consistent tagging in the Notes field. While we aim for broad coverage and auditability, selection and reporting biases remain possible—especially in fast-moving areas (OTA security, supply-chain tooling, cellular protocol implementations) where practice can change faster than archival publication.

Threats to validity (review-oriented)

- Selection bias: database coverage and keyword choices may miss relevant work (e.g., domain-specific venues for automotive/industrial systems).
- Publication bias: positive IDS results are more likely to be published than negative/operationally challenging results.
- Extraction bias: mapping heterogeneous contributions into a single schema can introduce misclassification; we mitigate this by using Table 1 as a coding guide and by capturing notes/assumptions during extraction.
- Temporal validity: standards, platform guidance, and threat techniques evolve; a 2012–2025 snapshot may not reflect the latest post-2025 changes.

3 Mobile and embedded ecosystem overview

This section frames the ecosystem we study—smartphones, wearables, IoT and industrial embedded systems—through a lifecycle and stack lens, highlighting trust anchors, update mechanisms, and protocol realities that shape the threat surface and defenses.

Device classes and stack layers

- Smartphones: application SoC + baseband/modem + secure element; userland apps atop vendor firmware; carrier-controlled cellular stack.
- IoT nodes: MCUs/SoCs with constrained resources; heterogeneous radios (BLE, Wi-Fi, LPWAN such as LoRaWAN); diverse vendor firmware and boot chains.
- Wearables: sensor-dense; BLE-centric connectivity; strict power/privacy constraints.
- Industrial embedded: PLCs/RTUs/gateways; real-time and safety-critical constraints; segmented networks and legacy stacks.

Layers common across classes: hardware (roots of trust), firmware (boot loaders, TEE), software (OS/apps), network (BLE/LoRaWAN/LTE-5G/TLS), and supply-chain (SBOM, provenance).

Table 2 summarizes how device classes map to stack layers and which controls/protocols dominate each class.
Table 2. Device classes × layers (orientation matrix).

Device Class	Hardware (Roots/Sensors)	Firmware (Boot/TEE)	Software (OS/Apps)	Network (Protocols)	Supply Chain (Provenance)
Smartphones	SoC, baseband, secure element	AVB, dm-verity; TrustZone/OP-TEE	Android/iOS; vendor services	Wi-Fi, BLE, LTE/5G; TLS 1.3 [24]	SBOM; SLSA; signed builds
Wearables	Sensors, low-power SoC	Measured boot (vendor); TEE options	RTOS/lightweight OS; companion app	BLE (GATT/Mesh: [58]), Wi-Fi; TLS 1.3 [24]	SBOM; constrained provenance
IoT (Constrained)	MCU/SoC; sensors; HSM/TPM (optional)	Verified/measured boot; TEE (TrustZone-M)	RTOS/bare-metal; services	BLE, Wi-Fi, LPWAN (LoRaWAN spec: [91]); DTLS/TLS (TLS 1.3: [24]	SBOM (SPDX/CycloneDX); SLSA; Sigstore
Industrial Embedded	PLC/RTU/gateway; field sensors	Vendor boot; secure modules	Real-time OS; safety logic	Ethernet/IP, Modbus, OPC UA; VPN/TLS (TLS 1.3: [24]	SBOM; regulated provenance
Gateways	x86/ARM; NICs; crypto modules	Secure boot; TEE (OP-TEE/SGX)	Linux; container runtime	Wi-Fi, cellular backhaul; TLS 1.3 [24]	SBOM; SLSA; artifact signing
Radios/Baseband	RF front-end; DSP; modem	Baseband firmware; secure boot	Control stack; diag tools	2G/3G/4G/5G; paging/AKA	Vendor provenance; signed images

Lifecycle and threats

- Development/manufacturing: supply-chain risks (counterfeit, compromised tooling); provenance integrity gaps. Controls: SBOM (SPDX/CycloneDX), SLSA, Sigstore.
- Boot/deployment: tampering and persistence via boot chain compromise. Controls: verified/measured boot (AVB, dm-verity, and DICE/TPM primitives: [25], [26] and attestation-oriented bootstrapping (e.g., SMART-style): [27], plus TEEs/isolated execution with careful threat modeling [28] and lightweight MCU isolation (e.g., Sancus): [29].
- Operation: protocol exploitation (BLE pairing/mesh and spoofing—survey + representative attack: [7], [30]; LoRaWAN jamming considerations: [31]; LTE/5G implementation risk—survey + systematic analysis: [12], [16]), malware and lateral movement, and side-channels (battery/power:[32]). Controls: TLS 1.3 [24], access control baselines, and ML-driven IDS (online/anomaly NIDS exemplars: [33]; evaluation metrics surveys: [25]).
- Update: OTA threats (rollback, TOCTOU, manifest abuse). Controls: SUIT manifests; signed, atomic updates; and attested post-update verification, aligned with practical update feasibility studies ([34], [35], [36]) and industry/verification perspectives ([37], [38]), plus platform/vertical guidance (Android OTA guidance: [39]; Android security reporting: [40]; connected vehicle OTA designs: [41], [42]).

Trust and update pipeline

- Roots of trust → attestation: hardware/firmware measurements bound into tokens (IETF RATS/EAT, PSA), enabling verifier policies across fleets.

- Verified/measured boot: integrity of bootloaders/partitions (AVB, dm-verity) to prevent persistence.
- TEE integration: isolation for sensitive code/keys (TrustZone/OP-TEE); attested claims about secure world state.
- Secure OTA: manifest-driven updates (SUIT), code signing, provenance; post-update attestation to re-establish trust.

4 Threat landscape

This section summarizes threats across lifecycle phases and stack layers, then highlights protocol-specific risks (BLE, LoRaWAN, LTE/5G AKA) that commonly bypass app-level defenses.

We emphasize protocol and firmware threats that are frequently underrepresented in app-centric security models: baseband analysis and fuzzing [17], [18] and dynamic firmware analysis at scale [19], [43] demonstrate how low-level vulnerabilities can undermine higher-layer controls.

Running example (RE-1): Compromised OTA + sensor spoofing in a mixed fleet
To keep the review concrete, we reuse one vignette across Threat → Detection → Response.

System context

- A mixed fleet: industrial gateway + constrained LoRaWAN sensors + operator smartphones used for commissioning and maintenance.
- The fleet relies on secure boot and partition integrity where available (AVB: [44]; dm-verity: [45]) and on firmware resiliency/recovery expectations (NIST SP 800-193: [21]).
- Updates are delivered OTA using a manifest-driven approach (SUIT RFC 9124: [20]) and fleet trust is anchored via remote attestation (RATS architecture RFC 9334: [22]; PSA token format RFC 9783: [23]).

Adversary path (representative)

1. Supply-chain foothold: attacker tampers with a build or release pipeline artifact (SLSA provenance: [46]; signing/transparency: [47]) and/or injects a dependency visible only through SBOM/provenance gaps (NTIA SBOM elements: [48]).
2. Update-channel abuse: a malicious but valid-looking update attempt targets devices that lack strong rollback protections or have inconsistent verification across components (SUIT RFC 9124: [20]; SUIT feasibility "reality check": [34]).
3. Physical effect via remote trigger: compromised devices or nearby attackers manipulate sensor readings and radio conditions; LoRaWAN join/key establishment weaknesses and jamming-style DoS are plausible enabling mechanisms (LoRaWAN survey: [49]; EuroS&P verification/patched join procedure: [50]).

Phase × layer: attacks and controls
Table 3 maps lifecycle phases to threats and primary controls to support cross-layer reasoning.
Table 3. Lifecycle phase × layer threats and primary controls.

Phase	Hardware Attacks	Firmware Attacks	Software Attacks	Network Attacks	Supply-Chain Attacks	Primary Controls
Manufacturing	counterfeit ICs; implant/trojan	malicious bootloader injection	poisoned SDK/deps	N/A	tampered BOM; logistics diversion	SBOM (SPDX/CycloneDX); SLSA provenance; artifact signing
Provisioning	key mishandling; insecure secure element	insecure flashing; debug interfaces	default creds; weak enrollment	N/A	rogue vendor tooling	HSM-backed keys; hardening; attestation at onboarding
Boot	fault injection; glitching	bootkits; verified/measured boot bypass	early-service persistence	N/A	N/A	AVB/dm-verity; measured boot; TPM/DICE ([26], [51]); SMART-style attestation ([27]); RA pre-policy
Deployment	physical tamper; probes	hidden modules; baseband firmware weaknesses	supply-chain payloads; privilege escalation	misconfig; weak protocols	N/A	TEE isolation; RA of software/firmware state
Operation	side-channels (power/RF); sensor spoofing	runtime patching; update agent abuse	malware; lateral movement; exfiltration	MITM; replay; downgrade; paging/AKA privacy	N/A	TLS/DTLS; protocol hardening; AI/ML IDS; continuous RA
Update	rollback via physical access	manifest forgery; TOCTOU; unused code paths	malicious update scripts	N/A	malicious mirror; supply diversion	SUIT manifests; signed/atomic updates; post-update RA
Decommission	data remanence; key leakage	residual firmware modules	orphaned services	N/A	asset disposal tampering	secure wipe; key destruction; inventory closure

Protocol-specific risks (BLE, LoRaWAN, LTE/5G AKA)

Table 4 highlights protocol-level risks that commonly bypass app-layer defenses.

Table 4. Protocol-level risks that commonly bypass app-level defenses.

Protocol	Key Vulnerabilities	Common Exploits	Representative Mitigations	Key Sources
BLE	pairing downgrade; "Just Works" weakness; mesh routing abuse	MITM during pairing; reconnection spoof; replay; spoofing/impersonation	enforce LE Secure Connections (Bluetooth SIG overview: [7]) and monitor downgrade behavior (KNOB: [30]); harden pairing flows against confusion/downgrade ([52]); detect reconnection spoofing/impersonation patterns ([53], [54], [55])	Surveys/analyses: [7], [8], [56], [57]; representative attacks: [30], [52], [53], [54], [55], [58], [59], [60].

LoRaWAN	join/key establishment insecurities; jamming; MAC misuse	replay; key compromise; DoS via jamming	enforce frame counters and rekey; tune ADR; apply formally verified join-procedure patches (EuroS&P: [50]); align configurations with the LoRaWAN security specification ([61]) and jamming analyses ([31])	LoRaWAN survey: [49]; EuroS&P join verification: [50]; spec/jamming: [31], [61].
LTE/5G AKA	AKA privacy leaks; paging; baseband implementation risk	tracking via AKA; paging abuse; L2 exploits; botnet-driven signaling storms	apply protocol fixes and privacy hardening guided by surveys ([12], [13]); validate AKA via formal analyses (e.g., NDSS/EuroS&P: [14], [15]) and verification frameworks/tools ([62], [63], [64], [65], [66], [67], [68]); incorporate empirical paging privacy and load/DoS findings ([69], [70], [71], [72]) and systematic LTE analyses (LTEInspector: [73]; L2 weaknesses: [74], [75])	Surveys/analysis: [12], [13]; formal AKA: [14], [15], [62], [63], [65], [66], [67], [68]; systematic LTE: [73], [74], [75]; paging privacy/DoS: [69], [70], [71], [72].

Software, firmware, hardware, and supply-chain attacks

Software attacks

- Malware and spyware targeting mobile/embedded OSes and apps: privilege escalation, data exfiltration, and command-and-control; mobile-focused surveys and SLRs systematize families and detection patterns [9], [10], [11].

Firmware attacks

- Bootkits and verified/measured boot bypass; persistence via bootloader/partition tampering. Baseline hardening is anchored in verified boot guidance (AVB: [44]) and platform firmware resiliency (NIST SP 800-193: [21]).
- Firmware vulnerability discovery and auditing at scale remains challenging in embedded ecosystems; surveys and taxonomies highlight recurring classes and analysis techniques ([76], [77]), and dynamic analysis pipelines illustrate practical scaling limits [19], [43].
- OTA compromise: manifest forgery, rollback, TOCTOU; hidden backdoors in vendor images. Open standards for constrained updates are defined in SUIT (RFC 9124: [20]) and evaluated in practice [34]; operational guidance and applied designs highlight real-world failure modes and mitigations (Android OTA guidance: [39]; connected vehicle OTA: [41]).

Hardware attacks

- Chip-level tampering and implants; fault injection/glitching; and side-channels (power/RF/battery). Hardware isolation is typically provided via TEEs such as TrustZone (overview: [78]) and implemented by TEE OSes (OP-TEE docs: [79]), but practical side-channel surfaces remain (battery side-channels: [32]).

Supply-chain attacks

- Manufacturing compromise and build pipeline tampering. Risk management guidance includes NIST SP 800-161 Rev.1: [80].
- Mitigations: SBOM practices (NTIA: [48]; SPDX: [81]; CycloneDX: [82]), build provenance via SLSA ([46]), and signing/transparency via Sigstore ([47]).

5 Cyber-physical attack surface

Cyber-physical attacks exploit the tight coupling between software, firmware, hardware, and the physical environment. Unlike purely digital compromise (e.g., credential theft), cyber-physical compromise changes how devices sense, decide, and act—often through subtle manipulation of sensors, timing, and control paths.

System view (Figure)

Fig. 2 illustrates the coupled cyber-physical loop that motivates multi-layer defenses and telemetry fusion.

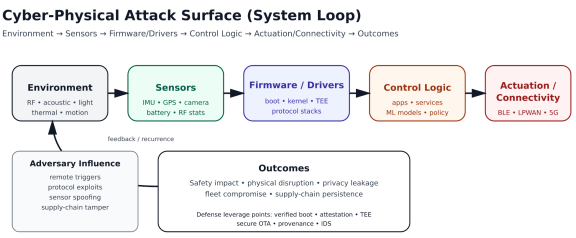


Fig. 2. A simplified cyber-physical loop showing how adversaries can influence physical signals (environment/sensors), software control logic, firmware/drivers, and connectivity paths to trigger unsafe outcomes.

Hardware–software interaction

Key interaction points

- Sensors → drivers → control logic: a sensor measurement becomes a privileged input to software decisions.
- Firmware/boot chain as a control dependency: persistence in firmware (bootloader/kernel) changes all higher-layer guarantees; verified boot reduces persistence risk (AVB: [44]; dm-verity: [45]) and recovery expectations are formalized in NIST SP 800-193 [21].
- Isolation boundaries: TEEs reduce blast radius for key material and sensitive services (TrustZone: [78]; OP-TEE: [79]).

Remote triggering risks

Wireless protocols below the app layer can expose device-wide attack paths even when apps are sandboxed.

Table 5 summarizes observable signals, common exploit classes, and defensive levers across the main cyber-physical surfaces.

Table 5. Attack surface matrix (signals → exploits → controls).

Surface	Observable Signals (for detection)	Typical Exploits	Defensive Levers (inline)
Sensors & environment	sensor streams, inconsistencies, cross-sensor residuals	spoofing, context manipulation	fusion + calibration; drift tests; forensic readiness playbooks
Firmware & boot	boot logs, partition hashes, version claims	bootkits, rollback, hidden modules	AVB/dm-verity ([44], [45]); NIST SP 800-193 ([21])
TEE boundary	attested enclave/service state, API traces	interface abuse, trusted app bugs	TrustZone/OP-TEE isolation ([78], [79])
Network protocols	flows, RSSI/time features, protocol events	MITM/replay/downgrade, tracking	protocol hardening; monitoring; 5G survey ([12])
OTA/update	update events, manifest validation results, rollback attempts	manifest forgery, mirror compromise	SUIT RFC 9124 ([20]); post-update verification
Supply chain	SBOM/provenance coverage, signing logs	counterfeit, poisoned build	NTIA SBOM ([48]); SLSA ([46]); Sigstore ([47])

Sensor/actuator semantics and context manipulation

A recurring challenge in mobile/embedded cyber-physical security is that "maliciousness" may manifest as plausible sensor values that are contextually inconsistent. This includes:

- Sensor spoofing with physically feasible ranges but incorrect correlations (e.g., inertial vs. GNSS mismatch), motivating cross-sensor residual tests and context-aware fusion ([3]).
- Actuation abuse where the command is syntactically valid but unsafe for the operating context (e.g., timing, magnitude, sequence), which often bypasses purely network-based IDS.

Implication: defenses must treat sensor/actuator interfaces as high-privilege inputs and explicitly model context (state, environment, mission phase) rather than only packet/process features.

Time, power, and RF as both signal and surface

Time and energy constraints shape both attacks and detection fidelity:

- Battery/power and side-channel signals can leak sensitive information or reveal application

behavior, while also providing telemetry for anomaly detection (battery side-channels: [32]).
- RF proximity and protocol-layer events (e.g., reconnection attempts, downgrades) can enable remote triggering even without compromising app code; surveys of wireless security emphasize protocol-level hardening and monitoring ([7], [12]).

Implication: "cyber-physical" monitoring is often multi-modal, spanning software logs, network metadata, and coarse hardware/energy signals.

Supply-chain → runtime → physical outcome path
Cyber-physical risk is amplified by the fact that supply-chain and update compromise can translate into persistent, fleet-wide unsafe behavior:

- Build or dependency poisoning can introduce logic that selectively activates under physical conditions (time/location/sensor triggers), complicating root cause analysis.
- Update attacks (mirror compromise, rollback, manifest forgery) can reintroduce known-bad states; structured update frameworks (SUIT RFC 9124: [20]) and provenance baselines (SLSA: [46]) provide the scaffolding needed to connect "what was built" to "what is running."

Implication: attack-surface analysis should explicitly connect provenance artifacts (SBOM/provenance/signatures) to runtime measurements (boot hashes/attestation claims) and to post-incident forensics.

6 Detection techniques

Detection in this ecosystem must address both cyber behaviors (process/network anomalies) and cyber-physical behaviors (sensor/power/RF manipulation), under constraints (battery, compute, bandwidth, privacy) and with an explicit trust baseline (boot + attestation).

Running example (RE-1): what detection looks like end-to-end
Signals to collect (minimal but actionable)

- Boot/integrity state from verified/measured boot when available (AVB: [44]; dm-verity: [45]) and resiliency signals aligned to recovery expectations (NIST SP 800-193: [21]).
- Attestation claims from devices/gateways (RFC 9334: [22]; RFC 9783: [23]) to detect trust baseline drift.
- Network telemetry at gateways and collectors informed by public dataset baselines (IoT-23: [83]; TON_IoT: [84]; CIC IoT: [85]).
- Cyber-physical residuals: cross-sensor checks and context features to reduce false alarms ([3]).

Practical detection policy

1. If attestation fails (or is stale) → isolate/quarantine and prioritize remediation (RFC 9334: [22]).
2. If attestation passes but behavioral model flags strong anomalies → stage containment and request re-attestation.
3. If anomalies correlate with update events → verify update manifests and enforce rollback protections (RFC 9124: [20], [34]).

Pipeline overview (Figure)
Fig. 3 summarizes the end-to-end detection pipeline and where trust signals (boot/attestation) are fused with telemetry.

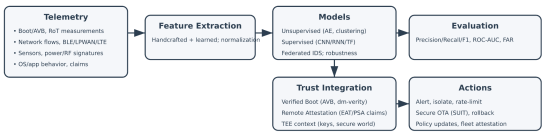


Fig. 3. Telemetry flows through feature extraction and models, integrated with trust (AVB/dm-verity, RA/EAT/PSA, TEE), producing actions such as alerts, isolation, and secure OTA via SUIT.

Table 6 lists representative public datasets and metrics commonly used to evaluate embedded/IoT IDS.

Table 6. Datasets × metrics summary for embedded/IoT IDS evaluation.

Dataset	Modality /Scope	Typical Models	Key Metrics	Sources
IoT-23	network flows; diverse IoT malware	classical ML; deep tabular/time series	precision/ recall/F1; ROC-AUC	[83]
TON_IoT	IIoT telemetry (net, system, sensor)	unsupervised AE; supervised DL	F1; FAR; latency; resource usage	[84]

CIC-IoT	IoT traffic; labeled scenarios	ML baselines; CNN/RNN	F1; ROC-AUC	[85]

Beyond benchmark datasets, a recurring theme in embedded/IoT monitoring is operational feasibility: online, lightweight anomaly detection (e.g., ensemble autoencoder NIDS) and clear evaluation metrics and threat models ([4], [25], [33]).

To help readers navigate the diversity of "detection" contributions, Table 7 provides a compact synthesis mapping representative works to (i) the layer/domain they study, (ii) what telemetry they require, and (iii) how the output is intended to be used operationally.

Table 7. Representative works × telemetry × deployment synthesis.

Representative Work	Layer/Domain	Telemetry Inputs	Technique/Focus	Typical Deployment	Operational Output
ML/IDS survey ([3])	network/host IDS	mixed (flows, host events)	taxonomy of methods and pitfalls	SOC / research baseline	guidance for model choice + evaluation
IoT-23 ([83])	IoT malware/network IDS	labeled flows/pcaps	dataset baseline	gateway/cloud	benchmark + feature baselines
TON_IoT ([84])	IIoT cyber-physical	net + system + sensor	dataset for multi-modal IDS	edge/cloud	benchmark for fusion + drift
Lightweight/online NIDS exemplar ([4])	IoT network IDS	gateway flow features	practical deployment framing	edge/gateway	alerting + correlation
Ensemble autoencoder NIDS ([33])	network anomaly detection	flow/time-series features	unsupervised AE ensembles	edge/gateway	anomaly score + triage
LTEInspector ([73])	cellular/baseband security	protocol traces + device behavior	systematic LTE security analysis	lab + operator tooling	vulnerability discovery + hardening
KNOB attack ([30])	BLE security	pairing negotiation metadata	protocol weakness exploitation	proximity attacker/defender	detection/hardening targets
LoRaWAN key-establishment verification ([50])	LPWAN security	join/handshake traces	formal verification + patches	device + network server	configuration + patch guidance
Firmware dynamic analysis at scale ([19], [43])	firmware/supply-chain	firmware images + emulation traces	scalable analysis pipelines	lab/cloud	triage, vuln discovery, provenance checks

Detection taxonomy: where, what, and under which trust assumptions

In practice, detection pipelines are best categorized along three axes:

Where detection runs

- On-device: lowest latency and best access to local signals, but constrained by battery/CPU and limited model update bandwidth.
- Edge/gateway: good vantage point for network telemetry and fleet-scale correlation; often the practical deployment point for IoT IDS ([4]).
- Cloud/SOC: best analytics and cross-fleet aggregation, but may be limited by privacy/regulatory constraints and delayed response windows.

What detection models

- Control-plane/protocol anomalies (e.g., downgrade/replay patterns) informed by protocol-specific analyses and surveys ([7], [12]).
- Host/process behavior and mobile malware indicators (mobile malware surveys/SLRs: [9], [10], [11]).
- Firmware integrity drift (unexpected partitions/versions) anchored in verified boot and firmware resiliency guidance ([21], [44]).
- Cyber-physical residuals (cross-sensor inconsistency, context violations) to reduce false positives in noisy environments ([3]).

Trust anchoring

- Purely behavioral models implicitly assume the sensor/telemetry pipeline is trustworthy.
- Trust-aware models explicitly condition decisions on boot/attestation state (RFC 9334: [22]; RFC 9783: [23]) so that "anomaly" is interpreted relative to a known baseline.

Evaluation pitfalls and reporting requirements

Many papers report high benchmark scores but under-specify the deployment regime. For TIFS-style rigor, evaluations should state:

- Threat model and adaptation budget: evasion, mimicry, and whether the attacker can influence training data ([3]).
- Data provenance and leakage controls: time-based splits, device-wise splits, and de-duplication to avoid optimistic leakage in IDS benchmarks ([4], [5]).
- Resource cost: inference latency, memory footprint, battery/energy overhead, and communications costs for telemetry offload.
- Calibration and operational metrics: false alarm rate (FAR) under benign distribution shifts; time-to-detect; and mean time to remediate.

Adversarial ML and telemetry/model poisoning

Embedded detection is vulnerable to attacker feedback loops:

- Evasion and mimicry: an attacker can shape behaviors to appear "normal," particularly when models rely on a narrow modality (only flows or only host events) or when the defender publishes stable thresholds.
- Adversarial drift: attackers can induce gradual behavior changes (or leverage legitimate software updates) to shift feature distributions until alarms are suppressed.
- Poisoning/backdoors: if models are updated from fleet telemetry, compromised devices can bias updates, inject backdoors, or skew normalization statistics; provenance and attestation can be used as filters (SLSA: [46]; RFC 9334: [22]).

Practical mitigation patterns include multi-modal fusion (harder to spoof consistently), trust-conditioned scoring (down-weight untrusted devices and gate decisions on attestation freshness), and gated model updates tied to signed artifacts and verified baselines. When retraining is required, additional safeguards include robust aggregation/outlier filtering for fleet updates, canary deployments, and rollback of model updates using the same secure-OTA and provenance discipline applied to firmware.

Privacy and ethics note Telemetry pipelines for mobile/embedded fleets frequently process sensitive behavioral and location-adjacent signals: logs can contain identifiers and user or process-level activity, RF "fingerprints" can become uniquely identifying across time/places, and attestation claims can leak device model, configuration, patch level, or stable identifiers that enable cross-session linkage. Designs should implement data minimization (collect only what is necessary), prefer on-device/edge feature extraction, and aggregate or bucket features when fine-grained values are not required for decisions. Access should be tightly controlled (least privilege, auditable access, and secure transport), and retention policies should be explicit and short by default (retain only what is needed for detection/forensics windows). Evaluations should state what raw data is collected versus derived features and how identifiers are handled (pseudonymization/rotation), and confirm that datasets are ethically sourced and legally shareable.

7 Defense mechanisms

This section integrates core controls—boot integrity, remote attestation, TEEs, secure OTA, and provenance—into a measurable pipeline.

In practice, these technical controls are deployed within governance and compliance baselines that constrain acceptable risk and required safeguards (ISO/IEC 27001: [86]; IEC 62443: [87]; NIST SP 800-53: [88]; NISTIR 8259A: [89]; OWASP MASVS: [90]).

Controls integration (Figure)

Fig. 4 illustrates how boot integrity, attestation, TEEs, secure OTA, and provenance combine into an operational control plane.

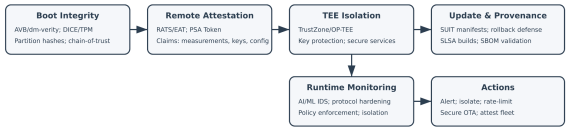


Fig. 4. AVB/dm-verity → RA/EAT/PSA → TEE → SUIT/SLSA/SBOM with runtime IDS and actions. Controls work together to provide cryptographic assurance and behavioral monitoring with operational responses.

Table 8 links each core control to measurable operational outcomes.

Table 8. Controls → measurable outcomes.

Control	Layer/ Phase	Measurable Outcome	Sources
Verified/ measured boot	firmware; boot	boot integrity rate; partition hash verification	AVB: [44]; dm-verity: [45]; NIST SP 800-193: [21]
Remote attestation (RATS/PSA)	deployment/operation	attestation pass rate; claim verification	RFC 9334: [22]; RFC 9783: [23]; RA surveys/representative schemes: [91], [92], [93]
TEE (TrustZone/OP-TEE)	hardware/software	key protection; secure service integrity	[78], [79]; TEE clarification/isolation designs: [28], [94]
Secure OTA (SUIT)	update	update success; rollback prevention	RFC 9124: [20], [34]; applied guidance/designs: [39], [41]
Provenance (SLSA, SBOM)	supply chain	build integrity; dependency traceability	[46], [48], [81], [82]

Where available, hardware-backed trust anchors complement remote attestation and measured boot: TPM 2.0 ([26]) and DICE ([51]) enable device identity composition and measurement binding, while lightweight attestation architectures and microcontroller-centric designs motivate minimal mechanisms for constrained devices ([27], [29]) and safety-critical/industrial settings ([95], [96]).

Fleet provisioning, identity, and key management

Defenses in mobile/embedded fleets often fail at "day 0" and "day 2" operations rather than at cryptographic primitives:

- Provisioning: establishing device identity, ownership, and initial trust anchors must resist factory compromise and counterfeit insertion (C-SCRM: [80]).
- Key lifecycle: rotation, revocation, and recovery are required for fleet-scale containment; TEEs and hardware roots (TrustZone/OP-TEE: [78], [79]; TPM: [26]; DICE: [51]) reduce key exposure but do not remove operational complexity.

Operational recommendation: treat keys and attestation credentials as managed assets with explicit revocation paths and incident runbooks.

Attestation at scale: freshness, sampling, and policy

Remote attestation architectures (RFC 9334: [22]) define roles and trust boundaries, but operational deployment raises fleet-level questions: Representative lightweight and fleet-oriented schemes include WISE ([97]), ERASMUS ([93]), CHARM ([98]), Delica ([99]), and hardware/low-resource designs ([100], [101]), including TrustZone-based secure boot + attestation patterns ([102]).

- Freshness windows: how often to attest given bandwidth and power constraints.
- Sampling policies: when to attest all devices vs. a subset (risk-based sampling).
- Verifier policy: how to encode acceptable states and how to handle "unknown but plausible" states (e.g., phased rollouts).

Practical pattern: combine periodic attestation with event-triggered attestation (after anomalies, after updates, after configuration changes) and couple

decisions to measured boot baselines (AVB/dm-verity: [44], [45]).

Running example (RE-1): threat → detection → response playbook
Containment (minutes)

- Quarantine devices or segments that fail attestation or show strong anomaly clusters; enforce least privilege and continuous verification patterns (RFC 9334: [22]; NIST SP 800-207: [103]).

Eradication & recovery (hours)

- Push a manifest-driven signed remediation update and re-verify post-update state (RFC 9124: [20]; [34]).
- Require post-update re-attestation before restoring privileges (RFC 9334: [22]).

Root-cause narrowing (days)

- Use SBOM + provenance to determine whether compromise is a dependency/build issue ([46], [48]) and validate signing/transparency evidence ([47]).

Secure OTA and provenance binding (what "good" looks like)
Secure remediation requires that updates are both verifiable and attributable:

- Verifiable update intent: manifest-driven updates with explicit component identifiers, constraints, and signatures (SUIT RFC 9124: [20]).
- Rollback resistance: ensure versions and policies prevent reintroduction of vulnerable images (firmware resiliency: [21]).
- Provenance linkage: bind SBOM/provenance records (NTIA SBOM: [40]; SLSA: [42]) to the exact image digests that devices measure and attest.

This binding closes a common gap: having "supply-chain paperwork" that is not cryptographically connected to what devices actually run.

8 Digital forensics in mobile/embedded devices

This section outlines practical evidence collection and analysis for constrained devices, integrating cryptographic trust (boot/attestation) with acquisition under operational/legal constraints.

Evidence flow (Figure)
Fig. 5 summarizes an evidence flow suitable for constrained devices while preserving cryptographic trust anchors.

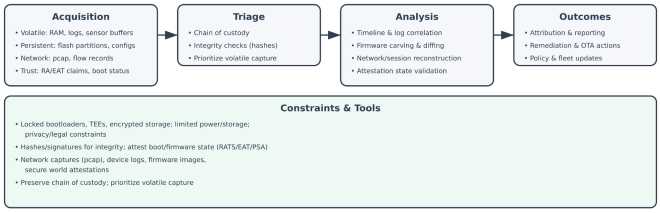


Fig. 5. Acquisition → triage → analysis → outcomes, covering volatile (RAM/logs/sensor buffers), persistent (flash/configs), network (pcap/flows), and trust (RA/EAT claims, boot status).

Table 9 enumerates practical evidence sources and the main constraints affecting collection and integrity.

Table 9. Evidence sources × constraints.

Evidence Source	Volatile/Persistent	Collection Constraints	Integrity/Trust Anchors	Notes
RAM snapshots/log buffers	volatile	power/time-limited; access via debug/agent	hashes; chain-of-custody	prioritize capture; avoid overwriting
Flash partitions/configs	persistent	locked bootloaders; encryption; vendor tools	AVB: [44]; dm-verity: [45]	image and verify; diff known-good
Network pcaps/flows	volatile/persistent	capture windows; privacy policies	TLS/DTLS metadata; signed logs	correlate timelines
Attestation/boot state	persistent	token retrieval; verifier availability	RFC 9334: [22]; RFC 9783: [23]	anchor state claims
Sensor/power/RF traces	volatile	noise; sampling limits; storage constraints	signed sample metadata; hashes	detect tampering/spoofing

Forensic readiness by design
Forensics in constrained ecosystems is rarely "add-on"; it must be engineered into products and fleets:

- Logging strategy: define minimal high-value logs (boot state, update events, policy changes, anomaly alerts) that are privacy-aware and storage-bounded.
- Evidence preservation hooks: support secure export of logs/attestation tokens from devices and gateways during incident response windows.
- Time synchronization: without stable time sources, correlation across logs/pcaps/sensor traces becomes unreliable; designs should record clock quality and drift.

These readiness practices align with resiliency expectations for platforms and recovery (NIST SP 800-193: [21]) and reduce mean time-to-understand incidents.

Trust-anchored evidence: using boot/attestation claims
Attestation evidence is most useful when treated as a cryptographic "state witness" that accompanies traditional artifacts:

- Pair forensic bundles (logs/pcaps/partitions) with contemporaneous attestation claims (RFC 9334: [22]; RFC 9783: [23]) to bound what code/configuration could have produced the artifacts.
- Record verification results (pass/fail plus policy version) so later review can reproduce decisions.

This approach improves post-incident confidence even when devices are intermittently connected or when some telemetry paths are untrusted.

Anti-forensics and practical constraints
Embedded adversaries may target the forensic process itself:

- Log tampering and deletion, especially when storage is limited.
- Induced resets/power cycling to erase volatile evidence.
- Update-triggered "cleaning" behavior that removes indicators.

Mitigations include append-only or remotely signed logs where feasible, rapid triage playbooks, and prioritization of volatile capture before remediation updates are pushed.

9 Research gaps

This section consolidates key gaps surfaced across the ecosystem, threats, detection, defense, and forensics.
Table 10 summarizes key gaps, their impacts, and suggested work items.
Table 10. Gap → impact → evidence → suggested work.

Gap	Impact	Evidence	Suggested Work

Tamper detection (constrained devices)	undetected physical attacks; unsafe states	TON_IoT: [84]	fuse power/RF + sensor logs; lightweight models
Fleet-scale attestation	blind spots; unverifiable runtime	RFC 9334: [22]; RFC 9783: [23]	claims schema; caching; offline policies; sampling
Provenance ↔ runtime trust	update trust breaks; rollback risks	RFC 9124: [20]; SLSA: [46]; NTIA SBOM: [48]	link images→SBOM→attestation; rollout checks
Multi-modal datasets/benchmarks	poor generalization; high false alarms	IoT-23: [83]; CIC: [85]; survey: [5]	define robustness tasks; leakage-safe splits
Forensic readiness	weak attribution/remediation	NIST SP 800-193: [21]	tooling for volatile capture; attest state in reports

10 Future research directions

This section outlines actionable directions linked to the identified gaps, with suggested milestones and evaluation criteria.

Directions

- AI-driven hardware monitoring and sensor fusion: power/RF + sensor integrity models deployable on constrained devices; drift-aware; evaluated on multi-modal telemetry (TON_IoT: [84]).
- Secure chip design and verifiable roots of trust: measurable boot + attestation primitives aligned with RFC 9334 ([22]) and token formats such as PSA attestation (RFC 9783: [23]).
- Supply-chain tracking integrated with attestation: SLSA provenance ([46]) and SBOM programs (NTIA: [48]; SPDX: [81]; CycloneDX: [82]) wired to images and attestation claims.
- Autonomous device trust verification and remediation: policy engines that fuse attestation outcomes with IDS signals; remediation via secure OTA (RFC 9124: [20]).

11 Conclusion

Mobile and embedded ecosystems are now critical infrastructure: they sense the physical world, execute control logic, and communicate over heterogeneous wireless networks. As a result, their security failures are not confined to data loss—compromise can translate into unsafe physical outcomes, fleet-wide disruption, and persistent supply-chain risk. This review synthesized threats and defenses across hardware, firmware, software, networks, and logistics, emphasizing that effective protection requires treating these layers as a coupled system rather than independent silos.

Key conclusions

- Cyber-physical assurance is two-part: cryptographic confidence in what is running (verified/measured boot and attestation) must be paired with statistical confidence in how the system behaves (telemetry-driven anomaly detection). Baseline integrity mechanisms such as Android Verified Boot ([44]) and dm-verity ([45]) reduce persistence, while remote attestation architectures (RFC 9334: [22]) enable verifiers to enforce policy at deployment and during operation.
- Secure remediation is as important as detection: fleet response depends on update mechanisms that are verifiable, rollback-safe, and operationally feasible on constrained devices. Manifest-driven secure update frameworks (RFC 9124: [20]) support structured remediation paths, and platform resiliency guidance (NIST SP 800-193: [21]) clarifies recovery requirements when compromise is suspected.
- Supply-chain security must be measurable and link to runtime truth: SBOM and provenance initiatives provide value only when bound to deployed images and runtime measurements. Practical baselines include NTIA SBOM minimum elements ([48]) and build-provenance hardening via SLSA ([46]), complemented by signing and transparency ecosystems such as Sigstore ([47]).

Declarations

Funding

No funding was received for this work.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials
Not applicable. This review did not generate new datasets; all sources analyzed are cited in the References. See the "Reproducibility and artifacts" statement in the Methodology section.

Code availability
Not applicable.

Author contributions
Sadeeq Muhammad led the research, including the literature review, synthesis, and manuscript drafting. Sandeep Kumar provided supervision and research guidance. All authors reviewed the manuscript and contributed to revisions.

Ethics approval
Not applicable.

Consent to participate
Not applicable.

Consent for publication
Not applicable.

References

[1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015, doi: 10.1016/j.comnet.2014.11.008.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A Survey on Security and Privacy Issues in Internet-of-Things,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017, doi: 10.1109/JIOT.2017.2694844.

[3] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” *IEEE Commun. Surv. Tutor.*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network Intrusion Detection for IoT Security Based on Learning Techniques,” *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[5] D. Gümüşbaş, T. Yıldırım, A. Genovese, and F. Scotti, “A Comprehensive Survey of Databases and Deep Learning Methods for Cybersecurity and Intrusion Detection Systems,” *IEEE Syst. J.*, vol. 15, no. 2, pp. 1717–1731, Jun. 2021, doi: 10.1109/JSYST.2020.2992966.

[6] Q. A. Al‐Haija and A. Droos, “A comprehensive survey on deep learning‐based intrusion detection systems in Internet of Things (IoT),” *Expert Syst.*, vol. 42, no. 2, p. e13726, Feb. 2025, doi: 10.1111/exsy.13726.

[7] M. Cäsar, T. Pawelke, J. Steffan, and G. Terhorst, “A survey on Bluetooth Low Energy security and privacy,” *Comput. Netw.*, vol. 205, p. 108712, Mar. 2022, doi: 10.1016/j.comnet.2021.108712.

[8] M. R. Ghori, T.-C. Wan, and G. C. Sodhy, “Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols,” *Sensors*, vol. 20, no. 12, p. 3590, Jan. 2020, doi: 10.3390/s20123590.

[9] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, “The Evolution of Android Malware and Android Analysis Techniques,” *ACM Comput. Surv.*, vol. 49, no. 4, pp. 1–41, Dec. 2017, doi: 10.1145/3017427.

[10] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, “A Survey of Android Malware Detection with Deep Neural Models,” *ACM Comput Surv*, vol. 53, no. 6, p. 126:1-126:36, Dec. 2020, doi: 10.1145/3417978.

[11] Y. Liu, C. Tantithamthavorn, L. Li, and Y. Liu, “Deep Learning for Android Malware Defenses: a Systematic Literature Review,” Aug. 09, 2022. doi: 10.1145/3544968.

[12] J. Cao *et al.*, “A Survey on Security Aspects for 3GPP 5G Networks,” *IEEE Commun. Surv. Tutor.*, vol. 22, no. 1, pp. 170–195, 2020, doi: 10.1109/COMST.2019.2951818.

[13] “Security for 5G Mobile Wireless Networks.” Accessed: Feb. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8125684

[14] C. Cremers and M. Dehnel-Wild, “Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion,” *Proc. 2019 Netw. Distrib. Syst. Secur. Symp.*, 2019, doi: 10.14722/ndss.2019.23394.

[15] A. Koutsos, “The 5G-AKA Authentication Protocol Privacy,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Jun. 2019, pp. 464–479. doi: 10.1109/EuroSP.2019.00041.

[16] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, “LTEInspector: 25th Annual Network and Distributed System Security Symposium, NDSS 2018,” *25th Annu. Netw. Distrib. Syst. Secur. Symp. NDSS 2018*, 2018, doi: 10.14722/ndss.2018.23313.

[17] G. Hernandez *et al.*, “FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware,” in *Proceedings 2022 Network and Distributed System Security Symposium*, San Diego, CA, USA: Internet Society, 2022. doi: 10.14722/ndss.2022.23136.

[18] A. Ranjbar, T. Yang, K. Tu, S. Khalilollahi, and S. R. Hussain, “Stateful Analysis and Fuzzing of Commercial Baseband Firmware: 46th IEEE Symposium on Security and Privacy, SP 2025,” *Proc. - 46th IEEE Symp. Secur. Priv. SP 2025*, pp. 1120–1139, 2025, doi: 10.1109/SP61157.2025.00143.

[19] Andrei Costin, Apostolis Zarras, and Aurélien Francillon, “Automated Dynamic Firmware Analysis at Scale | Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security,” ACM Conferences. Accessed: Feb. 10, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/2897845.2897900

[20] B. Moran, H. Tschofenig, and H. Birkholz, “A Manifest Information Model for Firmware Updates in Internet of Things (IoT) Devices,” Internet Engineering Task Force, Request for Comments RFC 9124, Jan. 2022. doi: 10.17487/RFC9124.

[21] A. Regenscheid, “Platform Firmware Resiliency Guidelines,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-193, May 2018. doi: 10.6028/NIST.SP.800-193.

[22] H. Birkholz, D. Thaler, M. Richardson, N. Smith, and W. Pan, “Remote ATtestation procedureS (RATS) Architecture,” Internet Engineering Task Force, Request for Comments RFC 9334, Jan. 2023. doi: 10.17487/RFC9334.

[23] H. Tschofenig, S. Frost, M. Brossard, A. L. Shaw, and T. Fossati, “Arm’s Platform Security Architecture (PSA) Attestation Token,” Internet Engineering Task Force, Request for Comments RFC 9783, Jun. 2025. doi: 10.17487/RFC9783.

[24] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” Internet Engineering Task Force, Request for Comments RFC 8446, Aug. 2018. doi: 10.17487/RFC8446.

[25] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues, “Model-Based Quantitative Network Security Metrics: A Survey,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2704–2734, 2017, doi: 10.1109/COMST.2017.2745505.

[26] “TPM 2.0 Library,” Trusted Computing Group. Accessed: Feb. 09, 2026. [Online]. Available: https://trustedcomputinggroup.org/resource/tpm-library-specification/

[27] K. E. Defrawy, A. Francillon, D. Perito, and G. Tsudik, “SMART: Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust”.

[28] M. Sabt, M. Achemlal, and A. Bouabdallah, “Trusted Execution Environment: What It is, and What It is Not,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 57–64. doi: 10.1109/Trustcom.2015.357.

[29] J. Noorman *et al.*, “Sancus: Low-cost Trustworthy Extensible Networked Devices with a Zero-software Trusted Computing Base,” presented at the 22nd USENIX Security Symposium (USENIX Security 13), 2013, pp. 479–498. Accessed: Feb. 10, 2026. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/noorman

[30] D. Antonioli, N. O. Tippenhauer, and K. B. Rasmussen, “The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR”.

[31] M. Ingham, J. Marchang, and D. Bhowmik, “IoT security vulnerabilities and predictive signal jamming attack analysis in LoRaWAN,” *IET Inf. Secur.*, vol. 14, no. 4, pp. 368–379, Jul. 2020, doi: 10.1049/iet-ifs.2019.0447.

[32] “Android Security,” Android Open Source Project. Accessed: Feb. 11, 2026. [Online]. Available: https://source.android.com/docs/security

[33] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection,” arXiv.org. Accessed: Feb. 09, 2026. [Online]. Available: https://arxiv.org/abs/1802.09089v2

[34] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, “Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check,” *IEEE Access*, vol. 7, pp. 71907–71920, 2019, doi: 10.1109/ACCESS.2019.2919760.

[35] S. El Jaouhari, “Toward a Secure Firmware OTA Updates for constrained IoT devices,” in *2022 IEEE International Smart Cities Conference (ISC2)*, Sep. 2022, pp. 1–6. doi: 10.1109/ISC255366.2022.9922087.

[36] S. El Jaouhari and E. Bouvet, “Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions,” *Internet Things*, vol. 18, p. 100508, May 2022, doi: 10.1016/j.iot.2022.100508.

[37] L. Seitz, M. Tiloca, M. Gunnarsson, and R. Höglund, “Secure Software Updates for IoT Based on Industry Requirements: 9th International Conference on Information Systems Security and Privacy, ICIPSS 2023,” *Proc. 9th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 1, pp. 698–705, Feb. 2023, doi: 10.5220/0011790100003405.

[38] A. Mukherjee, R. Gerdes, and T. Chantem, “Trusted Verification of Over-the-Air (OTA) Secure Software Updates on COTS Embedded Systems,” in *Proceedings Third International Workshop on Automotive and Autonomous Vehicle Security*, Virtual: Internet Society, 2021. doi: 10.14722/autosec.2021.23028.

[39] “OTA updates,” Android Open Source Project. Accessed: Feb. 09, 2026. [Online]. Available: https://source.android.com/docs/core/ota

[40] “Android Security Bulletin—November 2022 | Android Open Source Project.” Accessed: Feb. 09, 2026. [Online]. Available: https://source.android.com/docs/security/bulletin/2022-11-01

[41] A. Anwar, N. Chakraborty, and M. Zulkernine, “Secure OTA Software Updates for Connected Vehicles Using LoRaWAN and Blockchain,” *IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion*, 2024, Accessed: Feb. 09, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Secure-OTA-Software-Updates-for-Connected-Vehicles-Anwar-Chakraborty/b7342accc2e9e65d8f1f1339cb3906334dd5c0eb

[42] S. Yeasmin, A. Haque, and A. Sayegh, “A novel and failsafe blockchain framework for secure OTA updates in connected autonomous vehicles,” *Veh. Commun.*, vol. 43, p. 100658, Oct. 2023, doi: 10.1016/j.vehcom.2023.100658.

[43] D. D. Chen, M. Egele, M. Woo, and D. Brumley, “Towards Automated Dynamic Analysis for Linux-based Embedded Firmware,” in *Proceedings 2016 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2016. doi: 10.14722/ndss.2016.23415.

[44] “Verified Boot,” Android Open Source Project. Accessed: Feb. 09, 2026. [Online]. Available: https://source.android.com/docs/security/features/verifiedboot

[45] “dm-verity — The Linux Kernel documentation.” Accessed: Feb. 09, 2026. [Online]. Available: https://www.kernel.org/doc/html/latest/admin-guide/device-mapper/verity.html

[46] “Supply-chain Levels for Software Artifacts,” SLSA. Accessed: Feb. 09, 2026. [Online]. Available: https://slsa.dev/

[47] “sigstore,” sigstore. Accessed: Feb. 09, 2026. [Online]. Available: https://www.sigstore.dev/undefined/

[48] “The Minimum Elements For a Software Bill of Materials (SBOM) | National Telecommunications and Information Administration.” Accessed: Feb. 09, 2026. [Online]. Available: https://www.ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom

[49] H. Noura, T. Hatoum, O. Salman, J.-P. Yaacoub, and A. Chehab, “LoRaWAN security survey: Issues, threats and possible mitigation techniques,” *Internet Things*, vol. 12, p. 100303, Dec. 2020, doi: 10.1016/j.iot.2020.100303.

[50] S. Wesemeyer, I. Boureanu, Z. Smith, and H. Treharne, “Extensive Security Verification of the LoRaWAN Key-Establishment: Insecurities & Patches,” in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, Sep. 2020, pp. 425–444. doi: 10.1109/EuroSP48549.2020.00034.

[51] “DICE,” Trusted Computing Group. Accessed: Feb. 10, 2026. [Online]. Available: https://trustedcomputinggroup.org/work-groups/dice-architectures/

[52] M. von Tschirschnitz, L. Peuckert, F. Franzen, and J. Grossklags, “Method Confusion Attack on Bluetooth Pairing,” in *2021 IEEE Symposium on Security and Privacy (SP)*, May 2021, pp. 1332–1347. doi: 10.1109/SP40001.2021.00013.

[53] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, “BIAS: Bluetooth Impersonation AttackS,” in *2020 IEEE Symposium on Security and Privacy (SP)*, May 2020, pp. 549–562. doi: 10.1109/SP40000.2020.00093.

[54] J. Wu, Y. Nan, V. Kumar, A. Bianchi, M. Payer, and D. Xu, “BLESA: Spoofing Attacks against Reconnections in Bluetooth Low Energy”.

[55] A. C. T. Santos, J. L. S. Filho, Á. Í. S. Silva, V. Nigam, and I. E. Fonseca, “BLE injection-free attack: a novel attack on bluetooth low energy devices,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 5, pp. 5749–5759, May 2023, doi: 10.1007/s12652-019-01502-z.

[56] S. Sevier and A. Tekeoglu, “Analyzing the Security of Bluetooth Low Energy,” in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, Jan. 2019, pp. 1–5. doi: 10.23919/ELINFOCOM.2019.8706457.

[57] K. Lounis and M. Zulkernine, “Bluetooth Low Energy Makes ‘Just Works’ Not Work,” in *2019 3rd Cyber Security in Networking Conference (CSNet)*, Oct. 2019, pp. 99–106. doi: 10.1109/CSNet47905.2019.9108931.

[58] K. Lounis, “Cut It: Deauthentication Attack on Bluetooth,” in *2021 14th International Conference on Security of Information and Networks (SIN)*, Dec. 2021, pp. 1–8. doi: 10.1109/SIN54109.2021.9699265.

[59] E. Biham and L. Neumann, “Breaking the Bluetooth Pairing – The Fixed Coordinate Invalid Curve Attack,” in *Selected Areas in Cryptography – SAC 2019*, K. G. Paterson and D. Stebila, Eds., Cham: Springer International Publishing, 2020, pp. 250–273. doi: 10.1007/978-3-030-38471-5_11.

[60] C. Hensler and P. Tague, “Using bluetooth low energy spoofing to dispute device details: demo,” in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, in WiSec ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 340–342. doi: 10.1145/3317549.3326321.

[61] “LoRaWAN® Specification v1.0,” LoRa Alliance®. Accessed: Feb. 10, 2026. [Online]. Available: https://hz1.37b.myftpupload.com/resource_hub/lorawan-specification-v1-0/

[62] S. R. Hussain, M. Echeverria, I. Karim, O. Chowdhury, and E. Bertino, “5Greasoner: 26th ACM SIGSAC Conference on Computer and Communications Security, CCS 2019,” *CCS 2019 - Proc. 2019 ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 669–684, Nov. 2019, doi: 10.1145/3319535.3354263.

[63] E. K. K. Edris, M. Aiash, and J. K.-K. Loo, “Formal Verification and Analysis of Primary Authentication based on 5G-AKA Protocol,” in *2020 Seventh International Conference on Software Defined Systems (SDS)*, Paris, France: IEEE, Apr. 2020, pp. 256–261. doi: 10.1109/SDS49854.2020.9143899.

[64] I. Karim, S. R. Hussain, and E. Bertino, “ProChecker: An Automated Security and Privacy Analysis Framework for 4G LTE Protocol Implementations,” in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2021, pp. 773–785. doi: 10.1109/ICDCS51616.2021.00079.

[65] A. Braeken, “Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability,” *Comput Netw.*, 2020, Accessed: Feb. 09, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Symmetric-key-based-5G-AKA-authentication-protocol-Braeken/66978d9fa2f6531ecb6983e3d93d774062e0cd8b

[66] J. Munilla, M. Burmester, and R. Barco, “An enhanced symmetric-key based 5G-AKA protocol,” *Comput. Netw.*, vol. 198, p. 108373, Oct. 2021, doi: 10.1016/j.comnet.2021.108373.

[67] Y. Ko, I. W. A. J. Pawana, and I. You, “Formal Security Reassessment of the 5G-AKA-FS Protocol: Methodological Corrections and Augmented Verification Techniques,” *Sensors*, vol. 24, no. 24, p. 7979, Jan. 2024, doi: 10.3390/s24247979.

[68] Y. Fei, H. Zhu, and J. Yin, “FVF-AKA: A Formal Verification Framework of AKA Protocols for Multi-server IoT,” *Form. Asp. Comput.*, vol. 35, no. 4, pp. 1–36, Dec. 2023, doi: 10.1145/3599731.

[69] C. Sørseth, S. X. Zhou, S. F. Mjølsnes, and R. F. Olimid, “Experimental Analysis of Subscribers’ Privacy Exposure by LTE Paging,” *Wirel. Pers. Commun.*, vol. 109, no. 1, pp. 675–693, Nov. 2019, doi: 10.1007/s11277-019-06585-7.

[70] K. Fang and G. Yan, “Paging storm attacks against 4G/LTE networks from regional Android botnets: rationale, practicality, and implications,” in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, in WiSec ’20. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 295–305. doi: 10.1145/3395351.3399347.

[71] R. Borgaonkar, L. Hirschi, S. Park, and A. Shaik, “New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols,” *Proc. Priv. Enhancing Technol.*, 2019, Accessed: Feb. 09, 2026. [Online]. Available: https://petsymposium.org/popets/2019/popets-2019-0039.php

[72] T. Ta and J. S. Baras, “Enhancing Privacy in LTE Paging System Using Physical Layer Identification,” in *Data Privacy Management and Autonomous Spontaneous Security*, R. Di Pietro, J. Herranz, E. Damiani, and R. State, Eds., Berlin, Heidelberg: Springer, 2013, pp. 15–28. doi: 10.1007/978-3-642-35890-6_2.

[73] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, “LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE,” in *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2018. doi: 10.14722/ndss.2018.23313.

[74] D. Rupprecht, K. Kohls, T. Holz, and C. Popper, “Breaking LTE on Layer Two,” in *Proceedings - 2019 IEEE Symposium on Security and Privacy, SP 2019*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 1121–1136. doi: 10.1109/SP.2019.00006.

[75] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, “Call Me Maybe: Eavesdropping Encrypted LTE Calls With ReVoLTE,” presented at the 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 73–88. Accessed: Feb. 16, 2026. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/rupprecht

[76] T. Bakhshi, B. Ghita, and I. Kuzminykh, “A Review of IoT Firmware Vulnerabilities and Auditing Techniques,” *Sensors*, vol. 24, no. 2, p. 708, Jan. 2024, doi: 10.3390/s24020708.

[77] I. Nadir, H. Mahmood, and G. Asadullah, “A taxonomy of IoT firmware security and principal firmware analysis techniques,” *Int. J. Crit. Infrastruct. Prot.*, vol. 38, p. 100552, Sep. 2022, doi: 10.1016/j.ijcip.2022.100552.

[78] B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, “TrustZone Explained: Architectural Features and Use Cases,” in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, Pittsburgh, PA, USA: IEEE, Nov. 2016, pp. 445–451. doi: 10.1109/CIC.2016.065.

[79] “OP-TEE Documentation — OP-TEE documentation documentation.” Accessed: Feb. 09, 2026. [Online]. Available: https://optee.readthedocs.io/en/latest/

[80] J. Boyens, A. Smith, N. Bartol, K. Winkler, A. Holbrook, and M. Fallon, “Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-161 Rev. 1 (Withdrawn), May 2022. doi: 10.6028/NIST.SP.800-161r1.

[81] “SPDX Specification 3.0.1.” Accessed: Feb. 09, 2026. [Online]. Available: https://spdx.github.io/spdx-spec/v3.0.1/

[82] “Specification Overview | CycloneDX.” Accessed: Feb. 09, 2026. [Online]. Available: https://cyclonedx.org/specification/overview/

[83] “IoT-23 Dataset: A labeled dataset of Malware and Benign IoT Traffic.,” Stratosphere Laboratory. Accessed: Feb. 09, 2026. [Online]. Available: https://www.stratosphereips.org/datasets-iot23

[84] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, “Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China: IEEE, Dec. 2020, pp. 848–855. doi: 10.1109/TrustCom50675.2020.00114.

[85] “IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Accessed: Feb. 09, 2026. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html

[86] “ISO/IEC 27001:2022,” ISO. Accessed: Feb. 09, 2026. [Online]. Available: https://www.iso.org/standard/27001

[87] “IEC 62443-3-3:2013.” Accessed: Feb. 10, 2026. [Online]. Available: https://webstore.iec.ch/en/publication/7033

[88] J. T. Force, “Security and Privacy Controls for Information Systems and Organizations,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-53 Rev. 5, Dec. 2020. doi: 10.6028/NIST.SP.800-53r5.

[89] M. Fagan, K. Megas, K. Scarfone, and M. Smith, “IoT Device Cybersecurity Capability Core Baseline,” National Institute of Standards and Technology, NIST Internal or Interagency Report (NISTIR) 8259A, May 2020. doi: 10.6028/NIST.IR.8259A.

[90] “OWASP MASVS - OWASP Mobile Application Security.” Accessed: Feb. 09, 2026. [Online]. Available: https://mas.owasp.org/MASVS/

[91] B. Kuang, A. Fu, W. Susilo, S. Yu, and Y. Gao, “A survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects,” *Comput. Secur.*, vol. 112, p. 102498, Jan. 2022, doi: 10.1016/j.cose.2021.102498.

[92] J. Lee, “Collective Attestation for Manageable IoT Environments,” *Appl. Sci.*, vol. 8, no. 12, p. 2652, Dec. 2018, doi: 10.3390/app8122652.

[93] X. Carpent, N. Rattanavipanon, and G. Tsudik, “ERASMUS: Efficient remote attestation via self-measurement for unattended settings,” *Des. Autom. Test Eur.*, 2017, Accessed: Feb. 09, 2026. [Online]. Available: https://www.semanticscholar.org/paper/ERASMUS%3A-Efficient-remote-attestation-via-for-Carpent-Rattanavipanon/91a1757170b42d400281b06ce06e6616d048215f

[94] V. Costan, I. Lebedev, and S. Devadas, “Sanctum: Minimal Hardware Extensions for Strong Software Isolation,” presented at the 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 857–874. Accessed: Feb. 10, 2026. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/costan

[95] X. Carpent, K. Eldefrawy, N. Rattanavipanon, A.-R. Sadeghi, and G. Tsudik, “Reconciling remote attestation and safety-critical operation on simple IoT devices,” in *Proceedings of the 55th Annual Design Automation Conference*, in DAC ’18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 1–6. doi: 10.1145/3195970.3199853.

[96] B. Genge, A.-V. Duka, P. Haller, B. Crainicu, H. Sándor, and F. Graur, “Design, verification and implementation of a lightweight remote attestation protocol for process control systems,” in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, Jul. 2017, pp. 75–82. doi: 10.1109/INDIN.2017.8104750.

[97] M. Ammar, M. Washha, and B. Crispo, “WISE: Lightweight Intelligent Swarm Attestation Scheme for IoT (The Verifier’s Perspective),” in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, pp. 1–8. doi: 10.1109/WiMOB.2018.8589107.

[98] D. Li Calsi and V. Zaccaria, “Interruptible Remote Attestation of Low-end IoT Microcontrollers via Performance Counters,” *ACM Trans. Embed. Comput. Syst.*, vol. 22, no. 5, pp. 1–19, Sep. 2023, doi: 10.1145/3611674.

[99] Z. Wang, C. Sun, Q. Yao, D. Ding, and J. Ma, “Delica: Decentralized Lightweight Collective Attestation for Disruptive IoT Networks,” in *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, Dec. 2021, pp. 364–371. doi: 10.1109/ICPADS53394.2021.00051.

[100] J. Ahn, I.-G. Lee, and M. Kim, “Design and Implementation of Hardware-Based Remote Attestation for a Secure Internet of Things,” *Wirel. Pers. Commun.*, vol. 114, no. 1, pp. 295–327, Sep. 2020, doi: 10.1007/s11277-020-07364-5.

[101] W. Feng, Y. Qin, S. Zhao, and D. Feng, “AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS,” *Comput. Netw.*, vol. 134, pp. 167–182, Apr. 2018, doi: 10.1016/j.comnet.2018.01.039.

[102] Z. Ling *et al.*, “Secure boot, trusted boot and remote attestation for ARM TrustZone-based IoT Nodes,” *J. Syst. Archit.*, vol. 119, p. 102240, Oct. 2021, doi: 10.1016/j.sysarc.2021.102240.

[103] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero Trust Architecture,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-207, Aug. 2020. doi: 10.6028/NIST.SP.800-207.