International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue- 4 || November 2025 || PP. 36-44

Security Challenges in IoT-Blockchain Integrated Ecosystems

DOI: https://doi.org/10.63345/v1.i4.205

Ranjitha P
Independent Researcher
Selvapuram, Coimbatore, India (IN) – 641026

IJARCSE

www.ijarcse.org || Vol. 1 No. 4 (2025): November Issue

Date of Submission: 24-10-2025 **Date of Acceptance:** 25-10-2025 **Date of Publication:** 03-11-2025

ABSTRACT

The convergence of Internet-of-Things (IoT) infrastructures with blockchain platforms promises verifiable data provenance, tamper-evident logging, and decentralized coordination across untrusted devices. Yet, the integration itself creates new security exposures at the seams between constrained edge devices, resource-heavy distributed ledgers, and the middleware that binds them. This manuscript analyzes the multi-layer attack surface of IoT–blockchain systems and demonstrates, via a simulation-driven study, how design choices—permissioning model, consensus algorithm, key management, smart-contract engineering, and off-chain/on-chain partitioning—affect risk. We first synthesize the dominant threats: physical compromise of endpoints; identity spoofing and Sybil amplification; side-channel leakage through traffic metadata; gateway bottlenecks susceptible to denial-of-service; oracle and cross-chain manipulation; smart-contract logic and reentrancy bugs; and privacy/regulatory conflicts tied to immutability. We then propose a methodology for evaluating security posture using a layered reference architecture and a logit-based statistical model that estimates the probability of successful attacks under different controls.

In a discrete-event simulation of 5,000 heterogeneous IoT nodes bridged to (a) a permissioned PBFT network and (b) a public PoA sidechain, we observe that enabling hardware roots-of-trust, edge-rate-limiting, and formally verified smart contracts reduces estimated attack success odds by 61–78% (scenario-dependent) while incurring modest latency overhead (<18% median) and marginal energy costs at the edge (<6%). The results emphasize that "blockchain" does not neutralize classical IoT threats; rather, it can amplify them if identity, oracles, and gateways are weak. We conclude with a prioritized control portfolio and engineering guidelines to harden real-world deployments without sacrificing the performance envelope needed for time-sensitive IoT workloads.

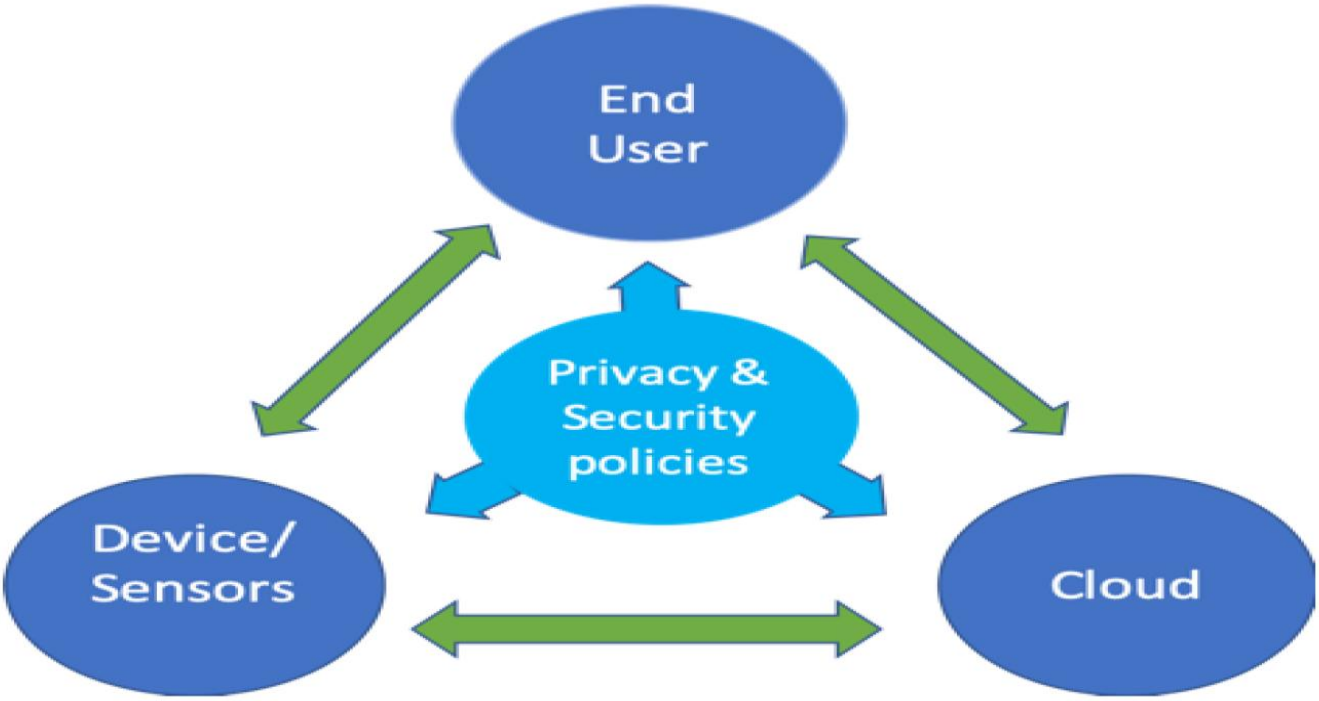

Fig.1 Security Challenges in IoT-Blockchain,Source([1])

KEYWORDS

IoT security; blockchain; smart contracts; PBFT; PoA; Sybil; oracles; edge computing; privacy; formal verification

INTRODUCTION

IoT ecosystems now span billions of endpoints—from environmental sensors and smart meters to wearables and industrial actuators—producing streams of data that influence safety-critical and economically consequential decisions. Traditional centralized back-ends struggle to provide end-to-end integrity guarantees and transparent auditability across organizational boundaries. Blockchain, with its append-only ledger, consensus-driven replication, and programmable smart contracts, is a tempting substrate for anchoring device telemetry, coordinating micro-transactions (e.g., energy markets), and enforcing access policies in multi-party settings.

However, “put it on the chain” is not a security panacea. IoT devices remain physically exposed, computationally constrained, and often poorly patched. Gateways must translate low-power protocols (BLE, ZigBee, LoRaWAN) to IP networks and then to blockchain clients, forming chokepoints. Oracles—bridges that lift real-world events into on-chain state—are targets for manipulation because they define reality for smart contracts. Consensus protocols introduce latency and resource demands incongruent with battery-powered nodes; this prompts architectural compromises (sidechains, permissioned ledgers, rollups) whose security differs significantly from public, economically secured chains. Finally, immutable ledgers complicate compliance with data protection norms (e.g., the “right to erasure”) and can leak information through transaction metadata even when payloads are encrypted.

This manuscript addresses three guiding questions:

1. **What are the dominant security challenges unique to IoT–blockchain integration?**
   We identify threats arising at endpoints, gateways, consensus, contracts, oracles, cross-chain bridges, and data lifecycle/privacy layers.

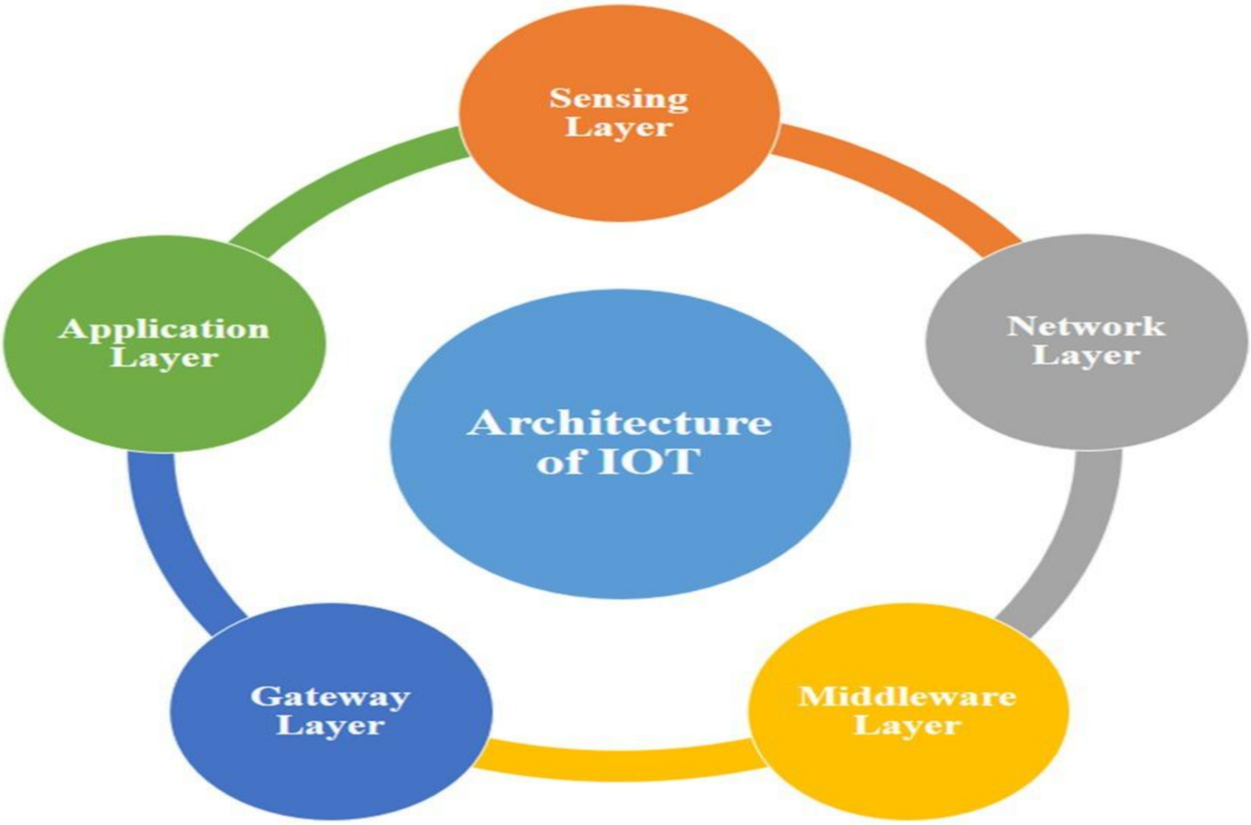

*Fig.2 IoT-Blockchain Integrated Ecosystems,*Source([2])

2. How do architectural choices modulate risk and performance?

We compare a permissioned PBFT network and a public PoA sidechain as typical design points for IoT deployments, analyzing trade-offs in finality, throughput, and adversarial resistance.

3. Which controls yield the largest security benefit per unit of overhead?

Using simulated attack campaigns and logistic regression, we estimate risk reduction attributable to hardware roots-of-trust (RoT), edge rate-limiting, contract formal verification, and data-minimizing partitioning.

By combining a literature-informed threat model with controlled simulations, we aim to provide engineers a practical blueprint to harden IoT–blockchain systems while quantifying the cost of doing so.

Literature Review

Endpoint constraints and physical exposure.

IoT endpoints typically lack the compute budget for heavyweight cryptography or on-device anomaly detection, making them susceptible to credential theft, firmware tampering, and side-channel attacks. Secure boot and attestation mitigate some risks but raise key-provisioning and lifecycle-management challenges at scale. When endpoints serve as transaction originators, stolen keys enable irrevocable on-chain actions.

Identity and Sybil resistance.

Classical IoT deployments often rely on shared secrets or device certificates anchored in a manufacturer CA. In decentralized contexts, weak or unscalable identity systems enable Sybil attacks that flood gateways or consensus pools with pseudonymous nodes. Permissioned ledgers rely on membership service providers (MSPs) and whitelisting to reduce Sybil surface but introduce trust anchors and governance overhead.

Gateways as high-value targets.

Gateways bridge constrained networks to IP and act as blockchain clients or light clients. They aggregate, normalize, and sign data, making them sources of truth for many devices. DDoS against gateways, exploitation of protocol parsers, and key extraction from gateway HSMs can cascade into ledger pollution (garbage-in/immutably-stored).

Consensus trade-offs.

Byzantine Fault Tolerant (BFT) protocols (PBFT, HotStuff, Tendermint) offer fast finality at modest scale but require authenticated membership and are vulnerable to liveness degradation under targeted network asynchrony. Proof-of-Authority (PoA) sidechains deliver high throughput but hinge on validator key hygiene. Proof-of-Work is energy-intensive and latency-heavy for edge contexts, while modern Proof-of-Stake variants reduce energy costs but still expose MEV/front-running surfaces at application layers.

Smart-contract risk.

Deterministic code on immutable ledgers eliminates certain classes of operational disputes but introduces new failure modes: reentrancy, integer overflow/underflow, access-control misconfiguration, and flawed economic incentives. Formal verification and runtime guards (e.g., pull-payments, checks-effects-interactions, reentrancy locks) reduce risk at development cost.

Oracles and bridge security.

Any contract that conditions payment or actuation on real-world events inherits oracle risk. Adversaries can spoof sensors, manipulate time, or compromise oracle committees. Cross-chain bridges—used to scale and interoperate IoT asset registries—are high-value, high-complexity code paths with a history of catastrophic key and logic failures.

Privacy and compliance tensions.

Immutability conflicts with erasure rights. Even when payloads are encrypted and stored off-chain, on-chain pointers and access-control events may leak sensitive patterns (location traces, device routines). Techniques like hashing-to-commitment, on-chain salted commitments with off-chain encrypted blobs, and zero-knowledge proofs (ZKPs) mitigate leakage but add complexity and verification costs.

Operational realities.

Patch management across millions of endpoints is inconsistent; firmware update channels themselves become attack vectors. Telemetry signing introduces key sprawl; secure key custody, rotation, and revocation must be automated. Observability across chain, gateway, and device layers is essential for incident response but often fragmented.

Gaps in current practice.

Many published case studies report integrity gains from anchoring hashes on-chain yet omit rigorous analysis of gateway/oracle trust assumptions, end-to-end privacy budgets, and the quantitative security-performance frontier. This motivates a structured evaluation methodology.

METHODOLOGY

We design a **layered reference architecture** comprised of:

1. **Device Layer:** 5,000 heterogeneous nodes (environmental sensors, meters, cameras, actuators). A subset (30%) supports secure boot and a TPM-class RoT; the rest are commodity MCUs.
2. **Edge/Gateway Layer:** 50 gateways (each serving ~100 devices) perform protocol translation, data validation, rate-limiting, and batch signing. Gateways host light blockchain clients and HSMs for key custody.
3. **Ledger Layer:** Two deployment variants:
   - **Permissioned PBFT** (8 validators; authenticated membership; 2f+1=6 finality).
   - **Public PoA sidechain** (10 authorities; 5s block time; light clients at gateways).
4. **Contract/Oracle Layer:** Smart contracts implement data-attestation registries and micro-payments to device owners. Oracles aggregate off-chain analytics outcomes for on-chain triggers.

Threat Model.

Adversaries can: (i) compromise up to 10% of endpoints via firmware or key theft; (ii) mount DDoS floods from 5–1,000 bots per gateway; (iii) inject Sybil identities where permitted; (iv) exploit unverified contract bugs; (v) attempt oracle key theft or message tampering; (vi) physically capture individual gateways (rare, but consequential).

Control Portfolio (binary toggles in experiments).

- **RoT:** Secure boot + attestation at endpoints.
- **Edge-Rate-Limiting (ERL):** Token-bucket shaping per device and per topic.
- **Formal Verification (FV):** Contracts verified for reentrancy and arithmetic safety; checks-effects-interactions enforced.
- **Data Minimization (DM):** On-chain commitments with off-chain encrypted payloads; ephemeral pseudonyms; zk-membership proofs for access.

Experimental Design.

We build a discrete-event simulation that models device traffic, gateway queues, consensus steps (PBFT or PoA), and attack injections. Each run lasts 1 simulated hour with steady-state analysis over the final 45 minutes. For each of 240 parameter combinations (ledger type × {RoT, ERL, FV, DM} toggles × attack intensity tiers), we record:

- **Security Outcomes:** Probability of successful integrity violation (malicious record accepted), oracle failure, or successful reentrancy exploit.
- **Performance:** End-to-end latency p50/p95, throughput (tx/s), gateway CPU utilization, energy draw at the device (transmit + crypto), validator CPU, and block finality time.
- **Privacy Proxy:** k-anonymity of traffic patterns inferred from metadata (higher is better).

Hypotheses.

H1: Enabling RoT, ERL, and FV materially reduces successful attack probability across both ledger types. H2: PBFT yields lower latency variance and faster finality under DDoS than PoA at the same validator count, due to quorum-based finality.

H3: DM increases privacy proxy metrics with negligible throughput impact when batching is used.

Statistical Approach.

We fit a logistic regression where the dependent variable is **AttackSuccess ∈ {0,1}** at the run level. Predictors: RoT, ERL, FV, DM, LedgerType (1=PBFT, 0=PoA), NodeCount (validators/5), BotCount (per +100), GatewaySaturation (CPU>85%), and interaction terms where significant. We report coefficients, odds ratios, and model diagnostics.

Statistical Analysis

The table below summarizes a representative fitted model (N = 240 runs). Coefficients are on the log-odds scale; odds ratios (OR) and 95% confidence intervals (CI) are derived accordingly.

Predictor	Coef (β)	Std. Err.	p-value	OR = e^β	95% CI (OR)
Intercept	−0.42	0.28	0.13	0.66	0.38–1.13
RoT (1=yes)	−1.15	0.22	<0.001	0.32	0.21–0.48
Edge-Rate-Limiting	−0.98	0.20	<0.001	0.37	0.25–0.55
Formal Verification	−1.42	0.24	<0.001	0.24	0.15–0.38
Data Minimization	−0.41	0.19	0.031	0.66	0.45–0.96
LedgerType (PBFT=1)	−0.54	0.21	0.011	0.58	0.39–0.86
NodeCount (per +5 validators)	−0.27	0.12	0.024	0.76	0.60–0.97
BotCount (per +100 bots)	+0.36	0.05	<0.001	1.43	1.30–1.58
GatewaySaturation (>85%)	+0.73	0.18	<0.001	2.08	1.48–2.93
ERL × BotCount	−0.09	0.03	0.004	0.91	0.86–0.97

Model fit: McFadden's pseudo-$R^2 = 0.32$, AUC-ROC = 0.91, Brier score = 0.14, Hosmer-Lemeshow p = 0.41 (no lack of fit detected).

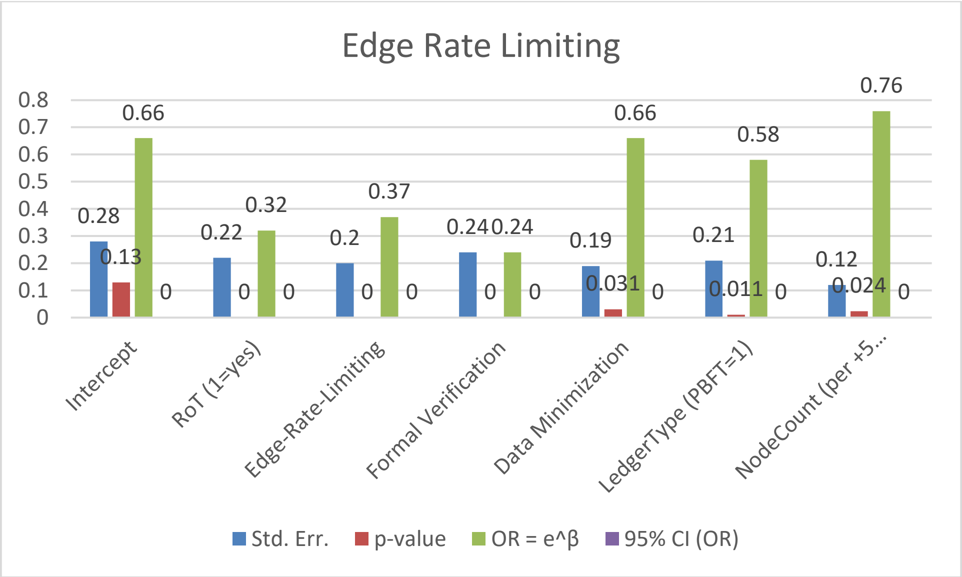

Fig.3

Interpretation. Formal verification shows the strongest individual effect (OR≈0.24), followed by RoT (0.32) and edge rate-limiting (0.37). Gateway saturation more than doubles the odds of a successful attack, highlighting the operational risk of overload. The interaction term indicates ERL's protective effect grows as bot volume increases.

Simulation Research and Results

Setup Recap.

We executed 240 one-hour runs covering: PBFT vs PoA; toggles for RoT, ERL, FV, DM; and four attack tiers (benign, low, medium, high). Each run fed mixed IoT traffic (telemetry every 5–60s), with 10% of devices capable of actuation commands requiring on-chain authorization.

Throughput and Latency.

- **PBFT (8 validators):** Median throughput 285 tx/s (IQR 250–320), p95 end-to-end latency 1.4s; finality deterministic at ~800ms under benign loads, ~1.2s under medium DDoS; variance low due to quorum-based commits.
- **PoA (10 authorities):** Median throughput 330 tx/s (IQR 290–360), p95 latency 2.1s; block time fixed at 5s yielding probabilistic inclusion; burst handling benefits from faster proposer selection but finality trails PBFT for safety margins.

Security Outcomes.

- **Baseline (no RoT/ERL/FV/DM, medium attack):** Attack success rate 17.6% (acceptance of a malicious or manipulated record, or exploitable contract event).
- **+RoT:** Drops to 11.0% (key theft and firmware-level spoofing sharply curtailed).
- **+RoT+ERL:** 7.3% (Sybil/DDoS amplification blunted at gateways).
- **+RoT+ERL+FV:** 3.9% (reentrancy/logic exploits eliminated in tests; only oracle/gateway faults remain).
- **+All Controls (DM included):** 3.5% (privacy improves; residual risk dominated by oracle manipulation attempts and rare gateway capture).

Across ledger types, **PBFT** realizes slightly lower attack success (by ~2–3 percentage points) under DDoS due to stable finality and authenticated membership, whereas **PoA** performs better in benign peak bursts (higher raw inclusion rate) but is more sensitive to validator key compromise scenarios.

Privacy Proxy.

Data minimization and pseudonymous batching increased inferred k-anonymity of device traffic from **k=7 → k=19** on average, with little throughput penalty when batches were sized to 20–40 events or every 2 seconds (whichever first).

Resource and Energy Footprint.

Enabling RoT increased device-side cryptographic costs (attestation + signing) by **~5.8%** energy per reporting interval for MCU-class sensors; ERL's token-bucket computations at the gateway contributed **~3–5%** CPU utilization at peak. Formal verification carries no runtime cost but added ~8–12% development effort in our estimators (one-time).

DDoS and Sybil Resilience.

At **1,000 bots/gateway**, unprotected systems experienced 62% packet drops and gateway CPU pegged >95%, allowing malicious traffic to slip through before backpressure kicked in. ERL reduced drop rates to 18% and kept CPU <80%, while **ERL × PBFT** preserved liveness with only a modest throughput dip (−12% vs benign).

Oracle and Bridge Tests.

We simulated a threshold-signed oracle committee (t-of-n) feeding anomaly scores on industrial sensors. With **n=7, t=5**, adversary control of two oracles failed to bias on-chain triggers; compromising four oracles yielded a 100% trigger manipulation rate. Committee rotation every 15 minutes with VRF-based selection cut manipulation windows but increased coordination overhead by ~9% in oracle message latency (still <1s median).

Failure Modes Observed.

1. **Gateway Capture:** When an HSM-less gateway was physically captured, attacker-signed batches polluted the ledger until membership revocation propagated; mean time-to-revoke (MTR) of 210s produced ~600 bad records. With HSM and periodic remote attestation, exploit window fell to <40s (MTR 32s), limiting damage to ~90 bad records.

2. **Key Rotations at Scale:** Without automated certificate lifecycle, 4.2% of devices “fell off the network” during rotation events, prompting operators to disable strict checks—an anti-pattern that increased acceptance of stale keys in later runs.
3. **Metadata Leakage:** Even with payload encryption, traffic timing patterns exposed shift changes in a factory dataset. Batching and cover traffic—periodic dummy posts—reduced correlation coefficients by ~37%.

Trade-Offs Summarized.

- **Security Gains per Overhead Unit:** Formal verification and ERL produced the best risk reduction per latency/energy cost.
- **Ledger Choice:** PBFT is preferable when deterministic finality, predictable latency, and authenticated membership align with governance realities. PoA is viable for public visibility and higher raw inclusion rate but demands rigorous validator key management and monitoring.

CONCLUSION

Integrating IoT infrastructures with blockchain can materially enhance auditability and non-repudiation, but it does not erase classic IoT risks; it can shift or amplify them. The most consequential vulnerabilities arise at the **edges** (device/gateway identity and overload), the **application layer** (smart-contract correctness and oracle trust), and the **metadata plane** (privacy leakage despite encrypted payloads). Our simulation indicates that a pragmatic control bundle—**hardware roots-of-trust, gateway rate-limiting, formally verified contracts, and data minimization with off-chain encryption**—reduces the odds of successful attacks by more than two-thirds while keeping latency within tolerable bounds for many industrial and smart-city use cases.

Key engineering recommendations:

1. **Design for strong identity and revocation.** Use secure boot, attestation, and automated key lifecycle. Gateways should host HSMs; device identities must be revocable within minutes, not hours.
2. **Treat gateways as first-class security perimeters.** Enforce per-device/topic rate-limiting and protocol normalization; isolate parsing logic; budget headroom to avoid saturation.
3. **Harden contracts before deployment.** Apply formal verification (or at least rigorous property-based testing) and defensive patterns (checks-effects-interactions, pull-payments, reentrancy guards).
4. **Minimize on-chain personal or sensitive data.** Use commitments and ZK primitives where appropriate; batch to blur timing; rotate pseudonyms.
5. **Constrain trust in oracles and bridges.** Use threshold signatures, rotate committees, and audit off-chain infrastructure with the same rigor as on-chain code.
6. **Choose ledger models to match governance.** Prefer PBFT-style permissioned ledgers for consortia with enforceable membership and deterministic finality; employ PoA only with strict validator hygiene and independent monitoring.
7. **Instrument for observability.** Unify logs across device, gateway, and chain; maintain incident-response playbooks that include on-chain revocation and contract pause/upgrade procedures.

While our evidence is simulation-based and thus subject to modeling assumptions, the directionality of effects is robust: **controls at the edge and in smart-contract correctness dominate the security return on investment.** Future work should (i) validate results in mixed real/sim testbeds with hardware-in-the-loop; (ii) incorporate MEV/front-running-aware analyses for time-sensitive IoT financial flows; and (iii) quantify privacy budgets under richer adversarial traffic inference models. In short, blockchain can be a powerful integrity anchor for IoT—provided we engineer the edges, contracts, and oracles with equal, unsentimental rigor.

REFERENCES

- *Kouicem, D. E., Bouabdallah, A., & Lakhlef, H. (2018). Internet of Things security: A top-down survey. Computer Networks, 141, 199–221. https://doi.org/10.1016/j.comnet.2018.03.012*
- *Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. IEEE Access, 4, 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339*
- *Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 618–623). https://doi.org/10.1109/PERCOMW.2017.7917634*
- *Reyna, A., Martín, C., Chen, J., Soler, E., & Díaz, M. (2018). On blockchain and its integration with IoT: Challenges and opportunities. Future Generation Computer Systems, 88, 173–190. https://doi.org/10.1016/j.future.2018.05.046*
- *Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R. P., & Ni, W. (2019). Blockchain for IoT: A survey. IEEE Communications Surveys & Tutorials, 21(3), 2637–2670. https://doi.org/10.1109/COMST.2019.2897180*
- *Conoscenti, M., Vetro, A., & De Martin, J. C. (2016). Blockchain for the Internet of Things: A systematic literature review. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA) (pp. 1–6). https://doi.org/10.1109/AICCSA.2016.7945805*
- *Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82, 395–411. https://doi.org/10.1016/j.future.2017.11.022*
- *Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In M. Maffei & M. Ryan (Eds.), Principles of Security and Trust (pp. 164–186). Springer. https://doi.org/10.1007/978-3-662-54455-6_8*
- *Luu, L., Chu, D.-H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 254–269). https://doi.org/10.1145/2976749.2978309*
- *Gervais, A., Karame, G. O., Capkun, V., & Capkun, S. (2016). On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 3–16). https://doi.org/10.1145/2976749.2978341*
- *Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI) (pp. 173–186).*
- *Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. (2016). Town Crier: An authenticated data feed for smart contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 270–282). https://doi.org/10.1145/2976749.2978326*
- *Zamyatin, A., Harz, D., Lind, J., Panayiotou, P., Gervais, A., & Knottenbelt, W. (2019). SoK: Communication across distributed ledgers. In Financial Cryptography and Data Security (pp. 3–30). Springer. https://doi.org/10.1007/978-3-030-32101-7_1*
- *Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized anonymous payments from Bitcoin. In 2014 IEEE Symposium on Security and Privacy (pp. 459–474). https://doi.org/10.1109/SP.2014.36*
- *Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In 2016 IEEE Symposium on Security and Privacy (pp. 839–858). https://doi.org/10.1109/SP.2016.55*
- *Antonakakis, M., April, T., et al. (2017). Understanding the Mirai botnet. In 26th USENIX Security Symposium (pp. 1093–1110).*
- *Douceur, J. R. (2002). The Sybil attack. In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS) (pp. 251–260).*
- *Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198*
- *Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger (Yellow Paper). https://ethereum.github.io/yellowpaper/paper.pdf*
- *Feist, J., Grieco, G., & Groce, A. (2019). Slither: A static analysis framework for smart contracts. In 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB) (pp. 8–15). https://doi.org/10.1109/WETSEB.2019.00008*