EdgeTrust-Offload: Quantifying Zero-Trust Overhead for Secure Task Offloading in Real-Time Edge Sensor Networks

Wanghley Soares Martins, Yifei Sun
ECE 654: Edge Computing

February 7, 2026

THE CORE IDEA

Edge computing lets sensor nodes offload heavy computation to nearby servers, saving time and energy. However, most offloading schemes assume every device on the local network can be trusted. When the data involved is protected health information (PHI) under HIPAA [17] or sensitive sensor streams under GDPR, that assumption is dangerous. Encrypting and authenticating every transaction—the Zero-Trust principle—adds overhead that may negate the benefit of offloading in the first place.

The question we answer: At what network latency does the security overhead of Zero-Trust make offloading *slower* than just computing on the sensor itself?

To answer this, we build **EdgeTrust-Offload** (Fig. 1), a testbed and decision framework with three components:

1. **Security Layer.** Every packet between devices travels through a Tailscale/WireGuard [10] encrypted tunnel with mutual authentication—no implicit trust.
2. **Sensing & Workload Engine.** An ESP32-S3 microcontroller continuously produces two representative workloads: a 1024-point FFT on a 256 Hz synthetic ECG stream (modeling ICU waveform analysis [14]) and a 64-tap FIR filter on a 1 Hz DS18B20 temperature stream (modeling thermal drift compensation [15]).
3. **Adaptive Scheduler.** For each task, a lightweight decision engine computes

$$C = \alpha \cdot T_{\text{exec}} + \beta \cdot T_{\text{net}} + \gamma \cdot E_{\text{crypto}} + \delta \cdot P_{\text{loss}}$$

separately for local and remote execution. If the cost of secure offloading exceeds local cost ($C_{\text{offload}} > C_{\text{local}}$), the task runs on the ESP32-S3; otherwise it is sent through the encrypted tunnel to an NVIDIA Jetson Nano for faster processing.

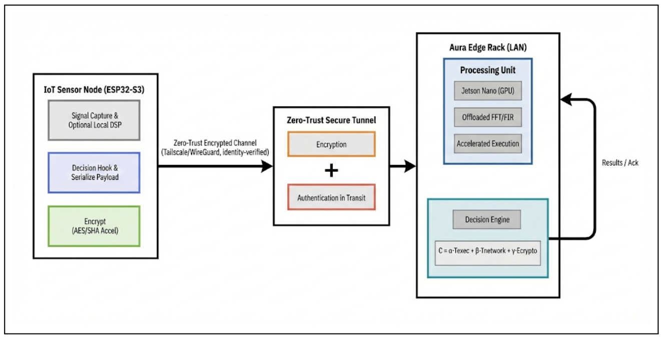
IoT Sensor Node (ESP32-S3)
Signal Capture & Optional Local DSP
Decision Hook & Serialize Payload
Encrypt (AES/SHA Accel)
Zero-Trust Encrypted Channel (Tailscale/WireGuard, identity-verified)
Zero-Trust Secure Tunnel
Encryption
+
Authentication in Transit
Aura Edge Rack (LAN)
Processing Unit
Jetson Nano (GPU)
Offloaded FFT/FIR
Accelerated Execution
Decision Engine
C = α·Texec + β·Tnetwork + γ·Ecrypto
Results / Ack

Figure 1: EdgeTrust-Offload System Architecture

The primary deliverable is the **crossover point** L^*: the network latency at which the encryption and authentication overhead of Zero-Trust makes offloading exactly as slow as local execution. Below L^*, offloading saves time; above it, the sensor should compute locally. No prior work has measured this threshold for resource-constrained IoT devices running continuous health-adjacent workloads.

THE FOUR WHYS

Why this? Every offloading study optimizes latency or energy but ignores the cost of encryption [2]. In HIPAA/FERPA environments, unencrypted offloading is not an option. We produce the missing design rule: *offload if and only if* network latency $L < L^*$, giving architects a concrete threshold [1], [9].

Why now? Three trends make this timely: NIST SP 800-207 established Zero-Trust as the federal standard [1]; hardware crypto accelerators on MCUs like the ESP32-S3 have closed the performance gap [3]; and Duke Health/Smart Home need privacy-preserving edge sensing today [4], [13].

Why me? We have hands-on ESP32/PCB experience, a real-time sensing research background, and the fully wired Aura rack testbed with Tailscale already running—minimizing ramp-up time. Our biomedical interests directly motivate the health-sensing workloads.

Why you? ECE 654 centers on low-latency, heterogeneous edge architectures—precisely the context where security overhead is most consequential. The crossover-point analysis directly answers the course's motivating question: *how do we architect edge systems that balance latency, security, and resource efficiency?* The findings and open-source code will be contributed back to Duke's ECE infrastructure for future student projects.

RELATED WORK

Zero-Trust Architecture. NIST SP 800-207 [1] defines ZTA, with Rose et al. [5] and Kindervag [6] extending the model. These frameworks focus on enterprise settings; none benchmark ZTA on microcontroller-class edge nodes.

Computation Offloading. Mao et al. [2] survey mobile edge offloading, Kumar et al. [7] formalize offload-vs-local, and Wang et al. [8] address DL inference. Lin et al. [9] study IoMT offloading. None include cryptographic overhead in the decision model.

Continuous Health Sensing at the Edge. Clifford et al. [14] show ICU alarms benefit from edge-local spectral analysis; Tamura et al. [15] highlight real-time temperature filtering; Dunn et al. [13] show wearable signals require low-latency, privacy-preserving pipelines.

Secure Edge Communication. Donenfeld [10] introduces WireGuard; Tailscale builds a user-space mesh with identity-based access; Cloudflare's Magic WAN [11] scales secure edge connectivity; and NVIDIA's TensorRT [12] accelerates inference.

Gap. No existing work models cryptographic overhead as part of the offload decision. We are the first to treat it as a *first-class scheduling variable* and empirically measure its impact on real IoT hardware with health-representative sensor workloads.

METHODOLOGY & VALIDATION

Our approach has four stages: (1) build the testbed, (2) measure local baselines, (3) measure offloading with and without encryption, and (4) sweep network conditions to find L^*.

Hardware Testbed: The Aura Rack

All experiments run on a physical 10-inch 9U rack ("Aura"):

- **Worker Nodes:** NVIDIA Jetson Nano (128 CUDA cores, 4 GB), Raspberry Pi 4 (4 GB), Orange Pi 4A
- **Sensor Node:** ESP32-S3 (T-Display S3) with AES-256/SHA-256 accelerators, DS18B20 probe, synthetic 256 Hz ECG via DAC
- **Network:** TP-Link TL-WR1502X Wi-Fi 6 router, TL-SG108 gigabit switch
- **Security:** Tailscale mesh (WireGuard tunnels, identity-based ACLs)

Workloads

We chose two tasks that bracket the compute spectrum of typical health-sensor processing:

1. **ICU Waveform Analysis (FFT).** A 1024-point FFT on a 256 Hz ECG buffer—a compute-heavy task representative of arrhythmia feature extraction [14]. Run via ESP-DSP on the MCU or NumPy/CuPy on the Jetson.
2. **Temperature Smoothing (FIR).** A 64-tap low-pass FIR on a 1 Hz DS18B20 stream—a lightweight task modeling thermal drift compensation [15]. Its small payload stresses the *minimum* overhead floor of ZTA.

Experimental Configurations

To isolate the effect of security overhead, we run each workload under four progressively harder scenarios:

1. **Local Baseline.** Task runs entirely on the ESP32-S3 (no network). Establishes T_{local} and energy E_{local}.
2. **Insecure Offload.** Task sent via plain TCP to the Jetson Nano—no encryption. Shows pure network + compute cost.
3. **Secure Offload (ZTA).** Same path, but through the WireGuard tunnel. The difference from (2) isolates crypto overhead.
4. **Secure + Congested.** ZTA offload under synthetic impairment (Linux `tc`: 10–100 ms jitter, 1–10% loss) to model real hospital Wi-Fi.

Each scenario runs **500 trials**; we report median, 95th-percentile, and max latency.

Metrics

- **End-to-End Latency:** $T_{e2e} = T_{\text{ser}} + T_{\text{enc}} + T_{\text{tx}} + T_{\text{dec}} + T_{\text{comp}} + T_{\text{ret}}$
- **Security Overhead Ratio:** $R_{\text{sec}} = T_{e2e}^{\text{secure}} \,/\, T_{e2e}^{\text{insecure}}$
- **Crossover Point:** L^* where $T_{e2e}^{\text{secure}}(L^*) = T_{\text{local}}$
- **Throughput:** Tasks per second under sustained load
- **Energy per Task:** INA219 measurements on ESP32-S3 rail

Week	Deliverable	Milestone
1–2 (Feb 7–21)	Literature review; Tailscale deployment; sensor wiring	Proposal
3–4 (Feb 22–Mar 7)	ESP32-S3 FFT & FIR implementations; local baselines	Development
5–6 (Mar 8–21)	Offload protocol (TCP & WireGuard); insecure vs. secure comparison	Progress Report
7–8 (Mar 22–Apr 4)	Scheduler decision engine; congestion injection experiments	Testing
9–10 (Apr 5–18)	Analysis; crossover-point plots; final paper & open-source release	Final Report

PROJECT PLAN & TIMELINE

RISK MANAGEMENT

- **Risk:** ESP32-S3 SRAM (512 KB) limits encryption buffers for 1024-point FFT payloads.
  Mitigation: Segment into 256-sample chunks; leverage AES hardware; fall back to FIR-only if needed.
- **Risk:** Network jitter >50 ms causes TCP retransmissions inflating variance.
  Mitigation: Use UDP with sequence numbers and retransmit logic; report median over 500 trials; characterize jitter distribution.
- **Risk:** Tailscale coordination-server dependency introduces an external failure mode.
  Mitigation: Deploy Headscale on the Aura rack; keep direct WireGuard fallback.
- **Risk:** Jetson Nano thermal throttling under sustained crypto + compute load.
  Mitigation: Monitor `tegrastats`; add a 40 mm fan; report throttled vs. unthrottled throughput.
- **Risk:** Synthetic ECG lacks clinical fidelity, limiting generalizability.
  Mitigation: Use MIT-BIH Arrhythmia Database waveforms [16] via ESP32 DAC; validate spectral content.

DUKE COMMUNITY IMPACT

This work benefits two Duke initiatives. The **Duke Smart Home Program** uses residential sensors whose data falls under FERPA; our L^* thresholds tell architects exactly when Zero-Trust overhead is acceptable. **Duke Health** is piloting edge processing of bedside telemetry (pulse oximetry, continuous temperature) [13]; our results provide the first device-class-specific latency guidance for HIPAA-compliant offloading.

All code, firmware, and analysis notebooks will be open-sourced on the GitHub[1]. The Aura rack remains available for future ECE 654 cohorts, and the latency dataset will serve as a reusable benchmark.

References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” National Institute of Standards and Technology, Tech. Rep. NIST SP 800-207, 2020. DOI: `10.6028/NIST.SP.800-207`.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017. DOI: `10.1109/COMST.2017.2745201`.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” in *IEEE Internet of Things Journal*, vol. 3, 2016, pp. 637–646. DOI: `10.1109/JIOT.2016.2579198`.

[4] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017. DOI: `10.1109/MC.2017.9`.

[5] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” *NIST Special Publication*, vol. 800, p. 207, 2020.

[6] J. Kindervag, “No more chewy centers: Introducing the zero trust model of information security,” *Forrester Research*, 2010.

[7] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” in *Mobile Networks and Applications*, vol. 18, 2013, pp. 129–140. DOI: `10.1007/s11036-012-0368-0`.

[8] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of edge computing and deep learning: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020. DOI: `10.1109/COMST.2020.2970550`.

[9] K. Lin, Y. Li, Q. Sun, S. Zhou, and L. T. Yang, “Computation offloading toward edge computing,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019. DOI: `10.1109/JPROC.2019.2922285`.

[10] J. A. Donenfeld, “Wireguard: Next generation kernel network tunnel,” in *Network and Distributed System Security Symposium (NDSS)*, 2017. DOI: `10.14722/ndss.2017.23160`.

[11] Cloudflare, Inc., *Magic wan: Secure, performant connectivity for branch offices and data centers*, `https://www.cloudflare.com/magic-wan/`, 2021.

[12] NVIDIA Corporation, *Tensorrt: Programmable inference accelerator*, `https://developer.nvidia.com/tensorrt`, 2019.

[13] J. Dunn, R. Runge, and M. Snyder, “Wearables and the medical revolution,” *Personalized Medicine*, vol. 15, no. 5, pp. 429–448, 2018. DOI: `10.2217/pme-2018-0044`.

[14] G. D. Clifford et al., “False alarm reduction in critical care,” *Physiological Measurement*, vol. 37, no. 8, E5–E23, 2016. DOI: `10.1088/0967-3334/37/8/E5`.

[15] T. Tamura, M. Huang, and T. Togawa, “Current developments in wearable thermometers,” *Advanced Biomedical Engineering*, vol. 7, pp. 88–99, 2018. DOI: `10.14326/abe.7.88`.

[16] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH arrhythmia database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. DOI: `10.1109/51.932724`.

[17] U.S. Congress, *Health insurance portability and accountability act of 1996*, Public Law 104-191, 1996.

[1]All code is available at `https://github.com/wanghley/edge-trust-offload`