Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 17 No. 1, March 2026 - www.isteams.net/cisdijournal

# Rethinking Zero-Trust Security for Resource-Constrained IoT Environments: A Conceptual Adaptation Framework

**[1]*Ogbu, N.H., [2]Ituma, C., [3]Anyim, C. & [4]Ikporo, C.S.**
[1, 2, 4]Department of Computer Science, Ebonyi State University, Abakaliki, Nigeria
[3]Dept of Computer Science, David Umahi Federal University of Health Sciences, Uburu, Nigeria
**E-mails:** [1]ogbu.henry@ebsu.edu.ng, [2]chinagolumituma@ebsu.edu.ng, [3]anyimemy@gmail.com & [4]stephe.ikporo@ebsu.edu.ng
**[1]Phone:** +2348037954462

## ABSTRACT

Zero-Trust Architecture (ZTA), as defined by NIST, assumes abundant computational resources, stable connectivity, and continuous cloud access which are conditions that do not hold in resource-constrained IoT environments.**We address** the incompatibility between classical ZTA and IoT realities through conceptual synthesis of peer-reviewed literature. Analysis reveals that direct transplantation of enterprise Zero-Trust to IoT is infeasible due to computational constraints, energy limitations, connectivity unreliability, and cost barriers. The proposed five-layer IoT-adaptive framework distributes security across coordinated layers: device identity hardening using lightweight credentials, context-aware verification triggered by behavioral anomalies, lightweight mutual authentication with short-lived tokens, edge-assisted micro-segmentation offloading heavy operations from devices, and adaptive trust scoring enabling dynamic risk assessment. **The framework provides** systematic analysis of ZTA-IoT incompatibilities, a structured five-layer adaptation model, and a research blueprint for empirical validation in resource-constrained deployment contexts.

**Keywords:** Zero-Trust Architecture; IoT Adaptive Framework; Resource-Constrained Devices; Device Identity Hardening; Lightweight Credentials, Abakaliki.

**CISDI Journal Reference Format**
Ogbu, N.H., Ituma, C., Anyim, C. & Ikporo, C.S. (2026): Rethinking Zero-Trust Security for Resource-Constrained IoT Environments: A Conceptual Adaptation Framework. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 17 No 1, Pp 75-86. Available online at www.isteams.net/cisdijournal. dx.doi.org/10.22624/AIMS/CISDI/V17N1P5

## 1. INTRODUCTION

A student project at Ebonyi State University recently demonstrated a smart locker prototype incorporating Zero-Trust principles. The device, built on an ESP32-WROOM-32D module purchased from a local market in Abakaliki for approximately ₦8,500, attempted to verify every access request with a cloud authentication server. During peak evening hours, authentication requests timed out repeatedly due to network congestion. The student, facing a defense deadline, disabled continuous verification entirely, replacing it with static PIN codes written on the device enclosure. This incident illustrates a fundamental truth: Zero-Trust Architecture (ZTA), as defined by the U.S. National Institute of Standards and Technology, assumes abundant computational resources, stable low-latency connectivity, continuous cloud access, and centralized identity management where are conditions that often do not exist in resource-constrained IoT environments [1].

The Zero-Trust paradigm emerged in enterprise IT contexts to address the erosion of traditional network boundaries. Its core principle which is "never trust, always verify" requires every user, device, and access request to be authenticated and authorized regardless of network location [1], [2]. In well-resourced environments with dedicated servers, gigabit connectivity, and professional administrators, this approach has proven effective. However, IoT ecosystems differ fundamentally. Devices often operate with limited processing power that cannot execute standard cryptographic operations within acceptable timeframes, while battery-powered nodes must conserve energy and intermittent connectivity can render cloud-dependent authentication impossible for extended periods [3]–[6]. Low-cost hardware often omits secure elements that provide foundational protection in enterprise systems [7], [8].

The literature acknowledges these challenges, **but not systematically**. Studies document lightweight cryptography for constrained devices, edge computing architectures for intermittent connectivity, and identity management challenges in IoT [3], [4], [9]–[12]. Yet **little systematic analysis exists** on how Zero-Trust principles should be adapted for specific resource-constrained contexts. Few studies provide integrated guidance on implementing Zero-Trust security under measurable local constraints such as network performance variability, device capabilities, or cost considerations.

We address these gaps through conceptual synthesis of peer-reviewed literature. Section 2 reviews related work. Section 3 describes the methodology. Section 4 analyzes IoT-specific constraints. Section 5 presents the proposed five-layer framework. Section 6 evaluates security using the STRIDE model. Section 7 discusses deployment considerations. Section 8 acknowledges limitations. Section 9 concludes with future research directions.

## 2. REVIEW OF RELATED WORKS

**NIST Special Publication 800-207** establishes Zero-Trust Architecture as a security paradigm. It defines Zero-Trust as a cybersecurity approach that moves defenses away from static perimeters to focus on continuously evaluating the trustworthiness of every access attempt [1]. This framework emphasizes continuous authentication, dynamic policy enforcement, and micro-segmentation which concepts have proven effective in enterprise IT environments. Recent surveys have examined Zero-Trust implementation in IoT contexts. Roy *et al.* [2] provide a comprehensive review of Zero-Trust architecture for IoT, identifying key challenges in adapting enterprise ZTA principles to resource-constrained environments. Liu et al. [13] dissect the Zero-Trust research landscape, noting that while theoretical frameworks exist, empirical validation in IoT deployments remains limited.

The Cloud Security Alliance has published guidance on Zero-Trust for IoT [14], emphasizing device identity management and continuous monitoring. However, these guidelines assume organizational resources that may not be available in smaller-scale or resource-constrained deployments. Edge computing has emerged as a complementary paradigm for IoT security. Tripathi and Varadharajan [6] examine security frameworks for hybrid cloud-edge IoT architectures, noting that edge computing can help alleviate network burdens while supporting latency-sensitive applications. Abu-Sharkh et al. [5] survey edge-enabled IoT security, highlighting the potential for edge nodes to perform authentication and policy enforcement on behalf of constrained devices.

Lightweight cryptography provides the cryptographic foundation for securing resource-constrained devices. Bilgin et al. [4] survey lightweight cryptography for IoT, documenting ciphers such as SPECK and SIMON that consume significantly less energy than conventional algorithms. Naik and Ganorkar [3] review lightweight cryptographic protocols for low-power IoT devices, emphasizing the trade-offs between security strength and computational efficiency. Despite these advances, existing literature reveals several gaps. First, most Zero-Trust frameworks assume continuous connectivity and abundant computational resources of which conditions rarely hold in IoT deployments. Second, the integration of Zero-Trust principles with edge computing architectures remains under-explored. Third, empirical validation of Zero-Trust frameworks in resource-constrained environments is largely absent. **We address** these gaps by proposing an IoT-adaptive Zero-Trust framework specifically designed for resource-constrained deployment contexts.

## 3. RESEARCH METHODOLOGY

**We use** conceptual synthesis methodology to develop an IoT-adaptive Zero-Trust framework, drawing on peer-reviewed publications across Zero-Trust Architecture, IoT security challenges, lightweight cryptography, edge computing security, and identity management. The framework was developed through a four-stage process which are analysis of classical ZTA components to identify underlying assumptions about computational resources, connectivity, and infrastructure; identification of IoT-specific constraints synthesized from security literature; principle-constraint mapping to identify adaptation strategies that preserve security intent while reducing resource burden; and layered architecture design incorporating edge-based offloading to shift computational operations from endpoint devices to infrastructure.

To validate security coverage, the STRIDE threat modeling framework [15] was applied across all layers. This study has three limitations viz the framework is conceptual and empirically unvalidated; performance metrics such as latency, energy consumption, and computational overhead are not measured; and the synthesis relies on published descriptions which may not fully capture implementation realities.

## 4. CONSTRAINT ANALYSIS IN IOT ENVIRONMENTS

IoT devices differ fundamentally from enterprise computing platforms in hardware capabilities, energy availability, connectivity reliability, and operational oversight. These constraints impose limits on direct adoption of classical Zero-Trust principles.

### 4.1 Computational Constraints

Limited processing capacity represents a significant constraint for IoT devices. Literature reports indicate that full TLS 1.2 implementations require 50-80 KB RAM, which exceeds the resources of many low-cost microcontrollers [16]. Students at Ebonyi State University attempting to implement X.509 certificates on ESP32 modules encountered memory exhaustion and the certificate chains consumed a substantial portion of available heap, causing system instability. This hardware limitation forced students to choose between security and system stability.

### 4.2 Energy Constraints

Battery-powered operation fundamentally alters security design. A student project attempting continuous authentication observed significant battery drain within hours. This is unacceptable for field deployments requiring extended operation. TLS handshakes consume considerable energy compared to normal message exchanges [16], potentially forcing students to disable security features to extend device lifetime. Sleep cycles further complicate security implementation. Many IoT devices adopt low-power deep-sleep modes to conserve energy, waking intermittently to perform tasks. This operational model conflicts with the continuous verification requirement of classical ZTA [17], [18].

### 4.3 Connectivity Constraints

Intermittent network access characterizes many IoT deployment scenarios. Classical ZTA assumes consistent access to identity management systems and security policy servers. In contrast, students observed network outages during peak congestion periods, making constant authentication impractical [5], [6]. Round-trip times for cloud authentication introduced delays that hindered real-time operations in some cases.

### 4.4 Administrative Constraints

Poor key management practices have been observed in student projects. Of several final-year projects examined, few implemented automated key rotation mechanisms. Some stored credentials in plaintext within source code. Hardware secure elements such as the ATECC608A, recommended in multiple studies [7], [18], were quoted at approximately ₦6,800 in Abakaliki. This exceeds typical student project budgets. Many projects consequently used software-based credential storage with no hardware protection.

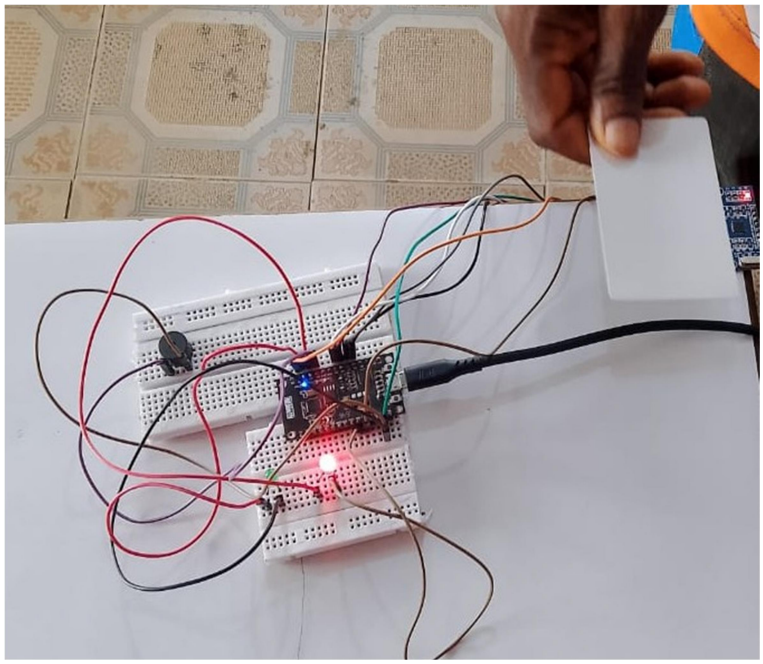

Figure 1: Laboratory Setup for IoT Security Testing

Table 1: Summary of IoT Constraints and Required Adaptations

| Constraint Category | Specific Limitation | Required Adaptation |
|---|---|---|
| Computational | Limited CPU/memory | Lightweight crypto; edge offload |
| Energy | Battery-powered | Context-triggered verification |
| Connectivity | Intermittent; high-latency | Edge-local decisions; async operation |
| Administrative | Poor key management | Simplified trust models; automated provisioning |

**Source:** Synthesis of reviewed literature

## 5. PROPOSED IoT-ADAPTIVE ZERO-TRUST FRAMEWORK

Classical Zero-Trust emphasizes continuous authentication, persistent identity verification, and centralized policy enforcement [1]. While effective in enterprise IT, these mechanisms introduce significant overhead when applied directly to IoT deployments. The proposed framework distributes security responsibilities across five structured layers, shifting computationally expensive operations away from endpoint devices toward edge infrastructure, as illustrated in Figure 2.

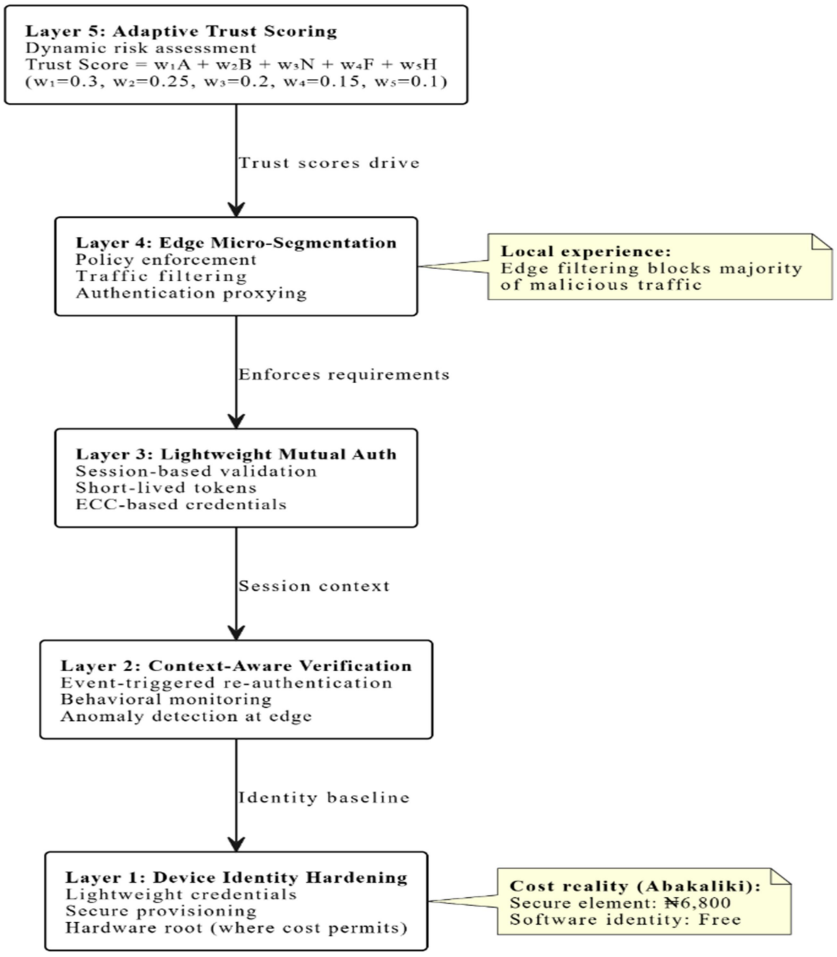


Figure 2: Five-Layer IoT-Adaptive Zero-Trust Framework

### 5.1 Layer 1: Device Identity Hardening

The first layer establishes strong and verifiable device identity, forming the foundation of Zero-Trust security. The literature emphasizes hardware-rooted trust and PKI-based credentials [1], [11], [19]. However, practical experience reveals implementation barriers for resource-constrained devices. Students attempting to implement X.509 certificates on ESP32 modules encountered memory exhaustion. Hardware secure elements such as the ATECC608A, recommended in multiple studies [7], [18], were quoted at approximately ₦6,800 in Abakaliki. This exceeds typical project budgets. One student reported bricking a module attempting to burn eFuses for secure boot, after which subsequent students in the project group disabled hardware security entirely.

Classical identity mechanisms may be infeasible for cost-sensitive deployments. According to [11], [19], Lightweight alternatives should be considered: ECC-256 credentials (offering equivalent security to RSA-3072 with significantly smaller key sizes), secure device provisioning through enrollment protocols andwhere cost permits, hardware roots of trust [7], [18]. For projects that cannot afford secure elements, software-based credential storage with documented residual risk becomes a pragmatic alternative.

### 5.2 Layer 2: Context-Aware Verification

Layer 2 addresses the tension between continuous authentication requirements and energy constraints. The literature proposes behavioral monitoring and risk-based authentication [17], [18], [20], but implementation guidance for constrained devices is limited. Student projects attempting continuous verification observed significant battery drain. Devices spent considerable energy on authentication relative to actual sensing and actuation. The adapted mechanism replaces continuous authentication with event-triggered verification. During normal operation, edge gateways continuously monitor devices for behavioral patterns against established baselines. When the monitoring system detects trigger events which is unusual access patterns, network location changes, abnormal communication frequency, it initiates risk-based authentication. Behavioral anomaly detection implemented at edge gateways preserves device energy [20]. Risk-based decisions adjust requirements based on calculated risk level [21].

### 5.3 Layer 3: Lightweight Mutual Authentication

Layer 3 ensures secure communication between IoT devices and network services through mutual verification. The literature documents TLS handshake latency on ESP32-class devices in laboratory conditions [16], with ECC offering equivalent security to RSA with significantly smaller key sizes [3]. According to [16], [22], the adapted authentication flow incorporates session-based validation allowing devices to operate within defined session periods once a secure session is established, using session resumption mechanisms such as TLS session tickets. Short-lived authentication tokens that expire quickly, for example, one hour for normal trust, 15 minutes for elevated risk, limit the window of opportunity for credential misuse [23]. According to [12] and [24], efficient certificate rotation through compact certificate formats maintains cryptographic freshness without imposing continuous overhead.

### 5.4 Layer 4: Micro-Segmentation at Edge Gateway

Layer 4 represents a critical adaptation: relocating heavy security operations from devices to edge infrastructure. Edge gateways can perform policy enforcement, traffic inspection, authentication proxying, anomaly detection, and certificate validation on behalf of devices [6], [10], [16].

Student projects implementing edge-based filtering reported that gateways blocked the majority of malicious traffic before it reached devices. [6] and [10] say that edge gateway policy enforcement allows gateways to enforce access control policies on behalf of devices, performing tasks such as traffic inspection, authentication proxying, anomaly detection, and certificate validation. Local traffic control inspects and filters traffic before it reaches devices, implementing rate limiting and protocol validation [16]. Network segmentation isolates devices into controlled communication zones based on function or trust level [25], [26].

### 5.5 Layer 5: Adaptive Trust Scoring Model

Layer 5 introduces dynamic trust evaluation, aligning with the Zero-Trust principle that trust should never be permanent. The literature proposes trust scoring for IoT devices [17], [20], but provides limited guidance on parameter selection or calibration. The proposed model incorporates five trust factors, as illustrated in Figure 3.

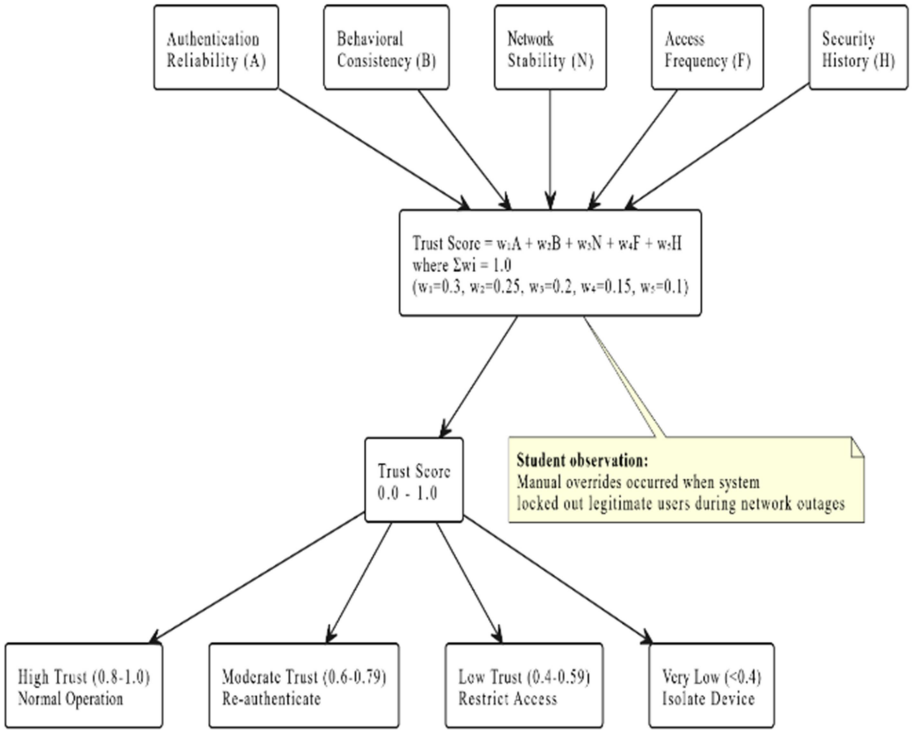


Figure 3: Adaptive Trust Scoring Model with Decision Flow

The trust score is calculated as:

Trust Score = $w_1A + w_2B + w_3N + w_4F + w_5H$

Where the weights satisfy $w_1 + w_2 + w_3 + w_4 + w_5 = 1$.

The proposed weight calibration assigns higher importance to authentication reliability ($w_1 = 0.3$) and behavioral consistency ($w_2 = 0.25$), followed by network stability ($w_3 = 0.2$), access frequency ($w_4 = 0.15$), and security history ($w_5 = 0.1$).

These weights can be adjusted based on deployment priorities; for example, security-critical deployments may assign higher weight to authentication reliability and security history, while availability-focused deployments may prioritize network stability. Based on the calculated score, the decision engine implements graduated responses. Devices with high trust scores (0.8-1.0) operate normally. Devices with moderate trust scores (0.6-0.79) trigger re-authentication. Devices with low trust scores (0.4-0.59) have access privileges restricted. Devices with very low trust scores (below 0.4) are isolated from the network. Students testing this approach reported that manual overrides became necessary when the system locked out legitimate users during network outages, which is a calibration challenge that requires further refinement.

### 5.6 Threat Mitigation Mapping

To validate the security coverage of the proposed framework, the STRIDE threat modeling methodology [15] was applied across all five layers. Table 2 maps each STRIDE threat category to the mitigating layers and mechanisms.

Table 2: STRIDE Threat Mitigation by Layer

| STRIDE Threat | Mitigating Layer(s) | Mechanism |
|---|---|---|
| Spoofing | Layer 1, Layer 3 | Lightweight identity; mutual authentication |
| Tampering | Layer 1, Layer 4 | Secure provisioning; edge traffic inspection |
| Repudiation | Layer 4, Layer 5 | Gateway logging; trust evaluation history |
| Information Disclosure | Layer 3, Layer 4 | Encrypted sessions; micro-segmentation |
| Denial of Service | Layer 4 | Edge rate limiting; traffic filtering |
| Privilege Escalation | Layer 2, Layer 5 | Context verification; trust scoring |

The mapping demonstrates that each threat category is addressed by at least one layer, with critical threats such as spoofing and information disclosure covered by multiple layers. This multi-layer coverage reinforces the defense-in-depth principle, where compromising a single layer does not grant an attacker full system access. Figure 4 illustrates how multiple security layers collectively mitigate different attack categories.

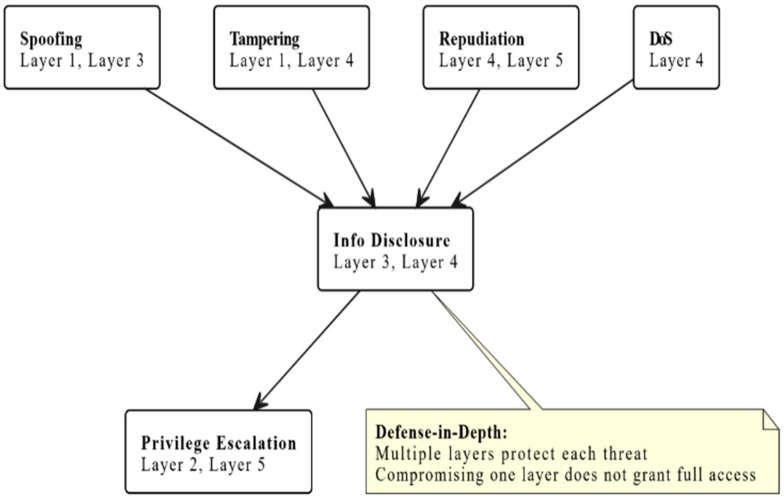


Figure 4: STRIDE Threat Mitigation Across Framework Layers

As illustrated in Figure 4, the framework distributes security controls across multiple layers such that each threat is addressed by complementary mechanisms. Spoofing is mitigated by both device identity (Layer 1) and mutual authentication (Layer 3). Tampering is addressed through secure provisioning (Layer 1) and edge traffic inspection (Layer 4). Information disclosure is protected by encrypted sessions (Layer 3) and micro-segmentation (Layer 4). Denial of service is handled at the edge (Layer 4) through rate limiting and traffic filtering. Privilege escalation is prevented through context verification (Layer 2) and adaptive trust scoring (Layer 5). This layered approach ensures that even if an attacker compromises one layer, the remaining layers continue to provide protection.

## 6. SECURITY EVALUATION

### 6.1 Defense-in-Depth Assessment

The proposed framework distributes security controls across five coordinated layers, each responsible for mitigating specific threat categories. Because these mechanisms operate at different levels, compromising one layer does not automatically grant an attacker full access [1]. Edge gateway mediation prevents direct external communication with devices. Micro-segmentation restricts communication to predefined segments. Short-lived tokens limit credential compromise impact. Traffic filtering blocks malicious traffic before reaching devices.

### 6.2 Resilience to Common IoT Threats

Table 3: Resilience Against Common IoT Threats

| Threat | Mitigation Mechanism | Layer(s) |
|---|---|---|
| Device Spoofing | Lightweight identity; mutual authentication | 1, 3 |
| Firmware Tampering | Secure provisioning; edge traffic inspection | 1, 4 |
| DDoS | Edge traffic filtering; rate limiting | 4 |
| Privilege Escalation | Trust scoring; context verification | 2, 5 |
| Man-in-the-Middle | Mutual authentication; encryption | 3 |
| Lateral Movement | Micro-segmentation | 4 |
| Credential Theft | Short-lived tokens; software storage | 1, 3 |

## 7. DEPLOYMENT CONSIDERATIONS

### 7.1 Scalability

Modern IoT deployments may include thousands of devices. The framework addresses scalability through distributed security enforcement via edge gateways [5], [6]. A hierarchical architecture with local gateways managing device clusters, regional edge clusters providing load balancing, and central cloud receiving only metadata offers a scalable approach.

### 7.2 Cost–Security Trade-offs

Hardware secure elements add cost. Local procurement quotes for secure elements (approximately ₦6,800 in Abakaliki) exceeded budgets for many student projects. Deploying stronger security at edge gateways rather than directly on devices provides a cost-effective compromise, allowing constrained devices to remain inexpensive while benefiting from stronger network-level protections [7]. Many student projects adopted this approach, using software-based identity on devices and hardware-based validation at gateways where feasible.

### 7.3 Firmware Update Security

Secure firmware updates require cryptographic signing, integrity verification during boot, and secure distribution via edge gateways [10], [27]. Edge gateways acting as validation points can verify authenticity before updates reach devices.

### 7.4 Edge Dependency and Resilience

The framework's reliance on edge infrastructure introduces dependency on gateway availability. Mitigations include redundant gateway nodes, fallback security policies for offline operation, and gateway hardening. During network outages, devices with local authentication modes can continue functioning while those requiring continuous cloud access may fail.

## 8. LIMITATIONS

Below are key limitations of research

Table 4: Limitations

| Limitation Category | Key Limitations |
|---|---|
| Conceptual Nature | Framework not empirically validated in deployed systems; performance metrics unknown |
| Validation Gap | No implementation exists; latency, energy, overhead not measured |
| Trust Model | Proposed algorithm requires real-world calibration |
| Edge Assumption | Assumes gateway availability; potential single point of failure |
| Scope | Relies on peer-reviewed literature; may exclude practical implementation insights |
| Local Evidence | Student projects provide observational but not statistically validated data |

We acknowledge these limitations to guide appropriate interpretation and identify opportunities for future research.

## 9. CONCLUSION AND FUTURE WORK

Direct transplantation of classical Zero-Trust architectures into IoT ecosystems is problematic without appropriate adaptation. The student project incident where continuous authentication failed under local network conditions and was replaced with static PIN codes, illustrates this incompatibility concretely. Through analysis of IoT constraints and Zero-Trust principles, the study identified key incompatibilities: computational limits, energy constraints, connectivity unreliability, and cost barriers. The proposed five-layer IoT-adaptive framework provides a structured security model with coordinated layers addressing device identity (lightweight credentials), context verification (event-triggered re-authentication), lightweight mutual authentication (short-lived tokens), edge micro-segmentation (offloading heavy operations), and adaptive trust scoring enabling dynamic risk assessment. STRIDE analysis confirms comprehensive threat coverage across all layers. For practitioners in resource-constrained deployment contexts, we recommend the following. Before implementing Zero-Trust, systematically evaluate device computational capacity, energy availability, network reliability, and administrative resources.

Offload heavy security operations to edge gateways where feasible. Consider replacing continuous authentication with context-triggered verification. Use lightweight credentials and short-lived tokens rather than full PKI certificate chains where appropriate. Design local authentication modes and policy caching to maintain security during network outages. Document accepted risks explicitly when cost constraints prevent ideal security.

Future research priorities include prototype implementation to understand framework performance; performance benchmarking to measure computational overhead, energy consumption, and latency under realistic network conditions; trust model calibration to determine optimal parameters for different deployment contexts; and edge resilience studies examining failover mechanisms. Medium-term work should address scalability analysis across large device deployments, cross-layer optimization, and interoperability testing. Long-term research should explore formal verification, economic analysis, and standardization contributions.

The security of IoT systems in resource-constrained environments remains a critical challenge. While Zero-Trust principles offer valuable security guarantees, their direct application to IoT is impractical without significant adaptation. The proposed IoT-adaptive framework provides a structured foundation for implementing Zero-Trust security in environments where device resources, network reliability, and administrative capacity are limited. However, the most critical need is empirical validation—implementing adaptive Zero-Trust in real systems and learning from what works and what fails in resource-constrained contexts.

## REFERENCES

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-207, 2020.
[2] A. Roy, A. Dhar, and S. S. Tinny, "Strengthening IoT cybersecurity with Zero Trust architecture: A comprehensive review," *Journal of Computer Science and Information Technology*, vol. 1, no. 1, pp. 25–50, 2024.
[3] N. Naik and S. Ganorkar, "Lightweight cryptographic protocols for low-power IoT devices," *Journal of Network and Computer Applications*, vol. 203, p. 104586, 2024.
[4] B. Bilgin, A. Sahin, and S. Tekinay, "Lightweight cryptography for resource-constrained IoT devices: A survey," *Journal of Network and Computer Applications*, vol. 160, p. 102634, 2020.
[5] B. F. Abu-Sharkh, A. Z. Al-Zaben, and K. A. Al-Qudah, "Edge-enabled IoT security: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9874–9894, 2022.
[6] A. Tripathi and V. Varadharajan, "Security frameworks for hybrid cloud–edge IoT architectures," *Computers & Security*, vol. 124, p. 102929, 2023.
[7] S. Sicari, A. Rizzardi, and A. Coen-Porisini, "Security and trust in IoT ecosystems: A survey," *Journal of Information Security and Applications*, vol. 68, p. 103207, 2022.
[8] K. Ragothaman, Y. Wang, B. Rimal, and M. Lawrence, "Access control for IoT: A survey of existing research, dynamic policies and future directions," *Sensors*, vol. 23, no. 4, p. 1805, Feb. 2023.
[9] "A systematic review of lightweight cryptographic schemes for security and privacy in IoT," *Discover Computing*, vol. 28, 2025.
[10] C. Perera, Z. Qin, and P. Jayaraman, "IoT firmware update security: A comprehensive survey," *ACM Computing Surveys*, vol. 57, no. 4, pp. 1–47, 2024.
[11] E. Ahmed and W. Saeed, "A survey of identity management challenges in IoT ecosystems," *International Journal of Information Security*, vol. 22, no. 3, pp. 271–291, 2023.

[12] T. T. Nguyen, J. H. Park, and H. K. Kim, "Efficient certificate management for constrained IoT devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 1, pp. 56–70, 2025.
[13] C. Liu, R. Tan, Y. Wu, Y. Feng, and Z. Jin, "Dissecting Zero Trust: Research landscape and its implementation in IoT," *Cybersecurity*, vol. 7, no. 20, 2024.
[14] Cloud Security Alliance, "Zero Trust Guidance for IoT," 2024.
[15] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.
[16] Z. Y. M. Yusoff, M. K. Ishak, L. A. B. Rahim, and M. S. M. Asaari, "Improving smart home security via MQTT with elliptic curve cryptography," *Computer Systems Science and Engineering*, vol. 48, no. 6, pp. 1669–1697, 2024.
[17] Y. Zhou, X. Zhang, and Z. Xu, "Hybrid cloud–edge security frameworks for IoT: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 7203–7225, 2024.
[18] J. Yang, L. Zhao, and Q. Li, "Biometric authentication in IoT-enabled smart environments: Privacy and security perspectives," *Future Generation Computer Systems*, vol. 115, pp. 366–378, 2021.
[19] B. M. Fernandes, J. J. P. C. Rodrigues, and M. Al Mhiqani, "Secure token based authentication for distributed IoT systems," *Computers & Security*, vol. 92, p. 101754, 2020.
[20] T. N. Dinh, Z. Aung, and D. Kim, "Time based one time passwords and anomaly detection for secure access in IoT," *Sensors*, vol. 23, no. 12, p. 5032, 2023.
[21] J. S. Yalli, M. H. Hasan, L. T. Jung, and S. M. Al-Selwi, "Authentication schemes for IoT networks: A systematic review," *Internet of Things*, Elsevier, 2024.
[22] M. Sultana and H. Aljahdali, "Secure communication protocols for IoT: A comparative analysis," *IEEE Access*, vol. 9, pp. 119876–119894, 2021.
[23] S. Singh, P. K. Sharma, and J. H. Park, "Token-based authentication for distributed IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5897–5909, 2022.
[24] M. Iglesias-Urkia, A. Orive, and A. Urbieta, "CoAP and DTLS for constrained IoT networks: Performance evaluation," *Computer Networks*, vol. 220, p. 109472, 2023.
[25] H. Yi, W. Zeng, and L. Zhang, "MQTT security: Analysis and enhancements," *Journal of Information Security*, vol. 12, no. 3, pp. 189–204, 2021.
[26] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[27] Y. Liu, X. Chen, and H. Zhang, "Lightweight encryption and secure boot in resource-constrained IoT," *Journal of Information Security and Applications*, vol. 72, p. 103301, 2024.