digital

MDPI

Article

Stability-Aware Security–Performance Trade-Off Analysis in Resource-Constrained IoT Systems: A Time-Series and Bootstrap-Based Evaluation of TLS and Hybrid ECC–AES Mechanisms

Carolina Del-Valle-Soto [1,*], Maria Fernanda Alvarez-Garcia [2], Ramon A. Briseño [1], Jafet Rodriguez [1] and Paolo Visconti [3]

[1] Facultad de Ingeniería, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Jalisco, Mexico; rbriseno@up.edu.mx (R.A.B.); arodrig@up.edu.mx (J.R.)
[2] Facultad de Ingeniería, Universidad del Istmo, Km 19.2 Carretera a Fraijanes, Fraijanes 01062, Guatemala; mfalvarez@unis.edu.gt
[3] Department of Innovation Engineering, University of Salento, 73100 Lecce, Italy; paolo.visconti@unisalento.it
* Correspondence: cvalle@up.edu.mx

Abstract

The increasing deployment of resource-constrained Internet of Things (IoT) devices requires security mechanisms that preserve confidentiality without compromising energy efficiency or responsiveness. Although Transport Layer Security (TLS) provides standardized protection for MQTT-based communication, its computational overhead may significantly affect embedded architectures. This study presents a controlled experimental evaluation of three communication configurations implemented on ESP32-based nodes: unencrypted Message Queuing Telemetry Transport (MQTT), MQTT over TLS 1.2, and an application-layer hybrid scheme combining Elliptic Curve Diffie–Hellman key exchange with AES-128 encryption. Second-level measurements of instantaneous current, accumulated energy, end-to-end latency, and memory footprint were collected across repeated experimental runs. Time-series diagnostics were performed to assess autocorrelation and stationarity, and block bootstrap resampling was applied to ensure dependence-aware statistical inference. The results indicate that TLS introduces the highest cumulative energy growth and latency dispersion, while the hybrid ECC–AES configuration demonstrates intermediate behavior with reduced overhead relative to TLS. Pareto frontier analysis shows that TLS is dominated in the joint energy–latency space, whereas the hybrid scheme represents a non-dominated compromise between security and efficiency. These findings provide a stability-aware and statistically robust framework for evaluating security–performance trade-offs in embedded IoT systems.

Keywords: Internet of Things (IoT); MQTT security; TLS; elliptic curve cryptography (ECC); AES encryption; energy efficiency; time-series analysis; block bootstrap inference; Pareto optimization; embedded systems

Check for updates

Academic Editor: Nik Bessis

Received: 24 February 2026
Revised: 21 April 2026
Accepted: 28 April 2026
Published: 2 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The rapid expansion of the IoT has led to the deployment of billions of interconnected devices operating in heterogeneous digital ecosystems. These systems are increasingly integrated into smart environments, industrial automation, environmental monitoring, and critical infrastructures. As part of the broader digital transformation process, IoT

architectures must balance connectivity, security, computational efficiency, and energy sustainability [1]. In this context, ensuring secure communication while maintaining acceptable performance levels has become one of the principal challenges in the design of digital systems [2].

MQTT has emerged as one of the most widely adopted communication protocols in IoT environments due to its lightweight publish–subscribe model, low bandwidth consumption, and suitability for resource-constrained devices [3]. MQTT is typically deployed over TCP/IP networks and commonly operates either without transport-layer protection (port 1883) or secured using Transport Layer Security (TLS) (port 8883) [4]. While TLS provides confidentiality, integrity, and authentication, it introduces additional computational overhead, memory usage, and energy consumption—factors that are critical in embedded systems such as microcontroller-based IoT nodes.

In low-power digital devices, particularly those based on microcontrollers such as the ESP32 platform, security mechanisms must be carefully evaluated against their impact on system performance [5]. The integration of cryptographic protocols can significantly influence current consumption, accumulated energy usage, latency, and memory footprint. Consequently, the design of secure IoT architectures requires a quantitative understanding of the trade-offs between security strength and system efficiency [6].

Beyond conventional TLS-based protection, alternative approaches such as application-layer cryptographic schemes have been proposed to reduce overhead while maintaining end-to-end security guarantees [7]. In particular, hybrid cryptographic mechanisms combining Elliptic Curve Cryptography (ECC) for key exchange and symmetric encryption for payload protection are considered attractive for constrained devices due to their reduced key sizes and computational efficiency compared to traditional RSA-based TLS handshakes [8]. However, the practical implications of these approaches on energy consumption, latency, and memory usage in real IoT deployments remain an open and relevant research question within digital system optimization.

Evaluating security mechanisms from a systems standpoint requires a multi-dimensional analysis. Energy consumption, typically measured in milliamperes (mA) and accumulated milliampere-hours (mAh), directly impacts battery life and sustainability [9]. End-to-end latency affects real-time responsiveness and quality of service. Memory usage (Flash and RAM) constrains firmware scalability and integration of additional digital services. Therefore, a comprehensive assessment must include statistical validation of performance differences across security schemes to ensure methodological rigor and reproducibility [10].

This paper presents a controlled experimental evaluation of three communication scenarios implemented on ESP32-based IoT nodes: (i) MQTT without encryption, (ii) MQTT secured with TLS 1.2, and (iii) MQTT protected using an application-layer hybrid cryptographic scheme based on ECC key exchange and AES-128 symmetric encryption. The study analyzes both the emitter and receiver devices under identical operating conditions. The performance metrics include instantaneous current consumption, accumulated energy usage, end-to-end latency derived from synchronized timestamps, and memory utilization at the firmware level.

To ensure statistical robustness under temporal dependence, the collected data were analyzed using a time-series aware inferential framework. Descriptive statistics were first computed to characterize central tendency and dispersion across security configurations. Autocorrelation diagnostics and stationarity testing were then performed to verify the temporal structure of the measurements. Because second-level samples exhibited serial correlation, classical independent-sample hypothesis tests were not applied. Instead, block bootstrap resampling with contiguous blocks was employed to preserve time dependence, and statistical significance was determined using 95% confidence intervals derived from

2000 bootstrap replicates. In addition, effect sizes (Cohen's *d*) were calculated to quantify the practical magnitude of performance differences, enabling a multidimensional assessment of security–performance trade-offs beyond purely statistical significance.

The contributions of this work are threefold:

- A systematic experimental comparison of transport-layer and application-layer security mechanisms in resource-constrained IoT devices.
- A quantitative assessment of the impact of security schemes on energy consumption, latency, and memory usage.
- A statistically validated analysis of security–performance trade-offs to support the design of efficient and secure digital IoT architectures.

Motivation and Contribution

By addressing both security and performance dimensions, this research contributes to the optimization of digital communication systems and supports informed decision-making in the design of energy-efficient and secure IoT infrastructures. The findings are particularly relevant for edge computing environments and large-scale digital deployments where scalability and sustainability are critical design considerations.

This study addresses the following research questions: (i) Does the integration of TLS 1.2 or a hybrid ECC–AES application-layer scheme produce statistically significant increases in energy consumption compared to unencrypted MQTT communication in resource-constrained IoT devices? (ii) Are the differences in end-to-end latency and memory footprint between transport-layer and application-layer security mechanisms statistically significant and practically relevant? (iii) How do these security schemes affect the temporal stability of energy consumption, particularly in terms of second-level fluctuations, transient peaks, and steady-state behavior over time? To answer these questions, we design a controlled experimental framework implemented on ESP32-based nodes, collecting high-resolution measurements of current consumption, accumulated energy, latency, and memory usage under identical operating conditions. The dataset is analyzed using descriptive statistics and inferential methods, complemented by second-by-second time-series analysis to evaluate variance, peak behavior, and stationarity. By integrating statistical rigor with dynamic temporal evaluation, this work contributes to the state of the art by providing reproducible, quantitative evidence of the security–performance trade-offs in IoT systems and by extending prior evaluations beyond average metrics toward stability-aware assessment of cryptographic mechanisms in digital embedded architectures.

From these research questions, three testable hypotheses are formulated. H_1: the mean cumulative energy under TLS 1.2 exceeds that under unencrypted MQTT at the 95% confidence level. H_2: the mean cumulative energy under the hybrid ECC–AES scheme is strictly between the unencrypted and TLS values, also at the 95% confidence level. H_3: TLS is strictly dominated by at least one alternative configuration in the joint (latency, cumulative-energy) objective space under the Pareto criterion. Each hypothesis is tested against its null counterpart via block-bootstrap confidence intervals (for H_1 and H_2) and dominance checking (for H_3).

2. Related Work

The rapid expansion of the IoT has intensified the need for secure and energy-efficient communication mechanisms in large-scale digital infrastructures. As billions of constrained devices are deployed in smart environments and industrial systems, the balance between security and performance has become a critical design challenge. Recent surveys highlight that security overhead remains one of the primary bottlenecks affecting scalability and sustainability in IoT ecosystems [4,7]. In particular, the integration of cryptographic protocols

in resource-limited embedded platforms significantly impacts computational load, energy consumption, and memory usage, thus requiring systematic quantitative evaluation.

MQTT has emerged as one of the most widely adopted communication protocols for IoT due to its lightweight publish–subscribe architecture. However, its native specification does not enforce encryption, relying instead on transport-layer mechanisms such as TLS. While TLS provides confidentiality and authentication, its computational overhead can be substantial in microcontroller-based systems [3,7]. Studies evaluating TLS performance in constrained environments report measurable increases in handshake latency and memory footprint, particularly when RSA-based cipher suites are employed. These findings underscore the need to explore alternative cryptographic strategies tailored to embedded digital architectures.

In response to these challenges, hybrid cryptographic approaches combining ECC with symmetric encryption (e.g., AES) have been proposed as more efficient alternatives for IoT devices [8,10]. ECC-based key exchange mechanisms offer reduced key sizes and lower computational complexity than traditional RSA, while AES provides efficient payload protection. Recent research demonstrates that hybrid schemes can reduce computational energy demand. However, empirical validation under controlled experimental conditions, including time-series stability analysis, remains limited. Most existing works focus on theoretical efficiency or simulation-based evaluation rather than real hardware measurements.

Recent experimental studies have specifically investigated the impact of TLS on MQTT-based IoT systems, providing relevant benchmarks for comparison. Chakravarty et al. [11] evaluated the effect of TLS encryption on MQTT performance in meteorological IoT networks, reporting increased latency and energy consumption due to cryptographic overhead, particularly during handshake operations. Similarly, Dimov et al. [12] analyzed resource trade-offs in TLS-secured MQTT-based IoT management systems, demonstrating that transport-layer security introduces significant memory and computational constraints in embedded environments. In addition, Liu and Al-Masri [13] conducted a comparative evaluation of MQTT reliability under different configurations, highlighting variability in performance metrics depending on protocol settings and network conditions.

While these studies provide valuable insights into MQTT performance under secure configurations, they primarily rely on aggregate metrics, single-run experiments, or protocol-level evaluations without incorporating high-resolution temporal analysis. In contrast, the present work extends these contributions by introducing second-level time-series measurement, autocorrelation-aware statistical modeling, and block bootstrap inference to ensure methodological rigor under serial dependence. Furthermore, unlike prior studies, this research integrates a multi-objective Pareto analysis to formally characterize the trade-off between energy consumption and latency, thereby providing a stability-aware and statistically validated framework for security–performance evaluation in resource-constrained IoT systems.

Furthermore, several studies have emphasized the importance of statistically rigorous performance evaluation in embedded IoT systems [5,6]. Despite advances in secure protocol design, there is still a gap in the literature regarding integrated analyses that simultaneously assess instantaneous current consumption, accumulated energy, latency distributions, memory footprint, and temporal stability under identical experimental conditions. Therefore, a comprehensive, statistically validated comparison between TLS and application-layer hybrid encryption in ESP32-based architectures represents a meaningful contribution to the optimization of secure digital IoT infrastructures.

The conceptual framework illustrated in Figure 1 synthesizes key constructs from the literature and organizes them into a layered structure linking IoT architectures, security mechanisms, and performance outcomes. At the architectural level, resource-constrained

embedded systems impose strict limitations on computation, memory, and energy, which directly influence the feasibility of security deployments. Within this context, security mechanisms act as the primary design variable affecting system behavior. These mechanisms are operationalized through communication protocols such as MQTT and associated cryptographic models, whose computational complexity impacts latency, energy consumption, and memory usage. Consequently, performance must be evaluated across multiple dimensions, including cumulative energy, latency distribution, and memory footprint. Unlike prior work, which often relies on aggregate metrics, this study integrates high-resolution temporal analysis, enabling the characterization of second-level stability, transient behavior, and steady-state dynamics. The framework incorporates a statistical validation layer based on time-series diagnostics and dependence-aware inference, including autocorrelation analysis, stationarity testing, and block bootstrap resampling. This integration enables robust comparison of security configurations under realistic temporal dependence, providing a stability-aware perspective on security.

While the relationships between architectural layers, security mechanisms, and performance metrics are described in the text, Figure 1 provides a structured abstraction that explicitly integrates these elements into a unified analytical model. This visual representation clarifies the hierarchical dependencies between system constraints, cryptographic design choices, and measurable performance outcomes, and serves as a conceptual bridge between the literature review and the experimental methodology. In particular, it highlights how security configurations propagate through system layers to influence energy, latency, and statistical evaluation, which is central to the stability-aware framework proposed in this work.

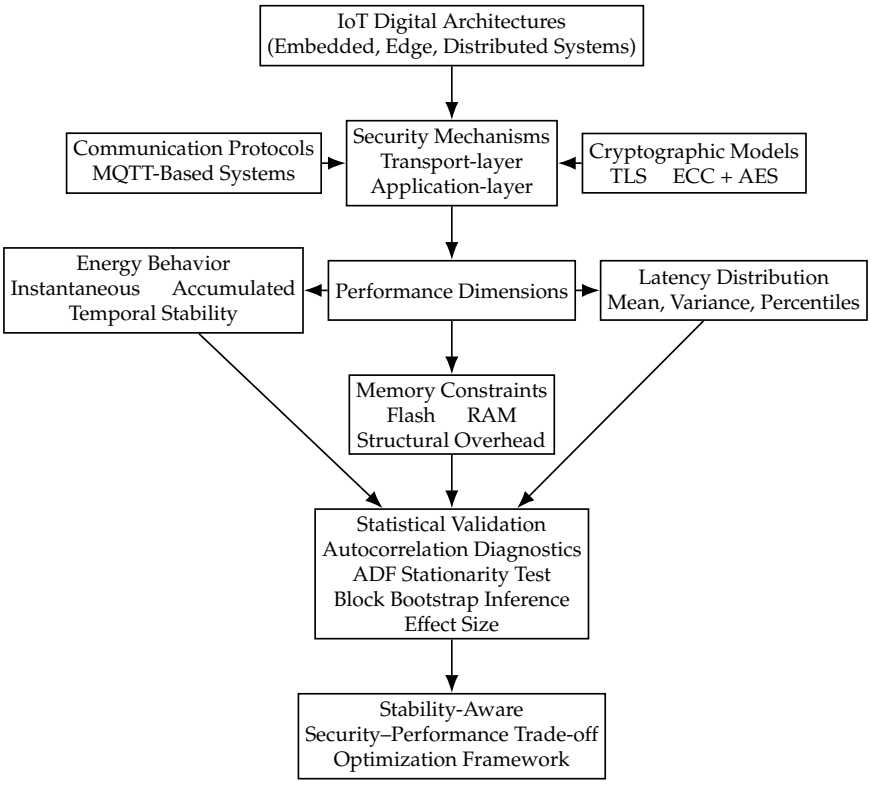

Figure 1. Conceptual framework integrating IoT architectural constraints, security mechanisms, and performance dimensions into a unified analytical model. The figure highlights the propagation of cryptographic design choices across system layers and their impact on energy, latency, and statistical validation, serving as a structural bridge between the literature and the experimental methodology.

Table 1 presents a structured comparison between representative studies in embedded IoT security evaluation and the present work. Unlike prior research, which often emphasizes protocol-level analysis, theoretical efficiency, or average energy measurements, most existing studies do not incorporate second-level temporal stability analysis nor apply inferential statistical validation to performance differences. While several works experimentally validate cryptographic overhead on constrained hardware, they typically rely on descriptive metrics without formal hypothesis testing. In contrast, this study integrates instantaneous and cumulative energy profiling, latency distribution modeling, structural memory assessment, second-by-second temporal stability analysis, and statistically rigorous inference. This multidimensional framework addresses a methodological gap in the literature by combining hardware validation, dynamic energy behavior, and inferential statistics within a unified experimental design.

Table 1. Critical comparison of representative studies addressing security–performance trade-offs in embedded IoT systems.

Reference	Security Scope	Hardware Validation	Energy Modeling Depth	Statistical Rigor
Adam et al. (2024) [14]	Security, privacy, trust, and architectural challenges in IoT systems	No (survey study)	Qualitative analysis of security and architectural challenges	Not applicable
Tasopoulos et al. (2023) [15]	TLS 1.3 (post-quantum) in constrained embedded devices	Yes (embedded devices)	Energy consumption evaluation under cryptographic overhead	Descriptive/experimental analysis
Glissa and Meddeb (2019) [16]	6LoWPAN end-to-end security (6LowPSec)	Yes (6LoWPAN devices)	Energy and protocol overhead evaluation	Descriptive analysis
Albert (2026) [17]	Energy-efficient cryptographic protocols for low-power IoT networks	No (conceptual/analytical study)	Energy-efficiency analysis of cryptographic protocols	Descriptive/analytical evaluation
Santos et al. (2020) [18]	TLS overhead in IoT gateways	Yes (edge devices)	Handshake-level energy analysis	Partial (no ANOVA/*t*-tests)
Brachmann et al. (2012) [19]	Secure communication in IoT	Yes (ARM Cortex-M)	Per-message energy profiling	Descriptive only
Santos et al. (2015) [20]	DTLS in constrained IoT	Yes (embedded platform)	Average energy and latency	Limited statistical treatment
Arias et al. (2015) [21]	ECC performance in IoT	Yes (microcontroller)	Execution energy estimation	No inferential testing
Hassija et al. (2019) [22]	IoT security frameworks	Partial (testbed validation)	Not energy-focused	No
Rizvi et al. (2018) [23]	Cryptographic overhead in IoT	Yes (hardware-based tests)	Energy per encryption cycle	Descriptive
Herrero (2021) [24]	Network and transport layer mechanisms in IoT	No (conceptual/book chapter)	Protocol-level discussion (no explicit energy modeling)	Descriptive/conceptual analysis
Brasser et al. (2018) [25]	Hardware-assisted IoT security	Yes (embedded platform)	Execution-cycle energy estimation	No inferential testing
Gomez et al. (2018) [26]	Low-power wireless communication (Bluetooth Low Energy, BLE) performance	Yes (embedded hardware)	Average energy per session	Limited statistical analysis
Malina et al. (2016) [27]	Lightweight cryptography for IoT security	Yes (hardware validation)	Energy per cryptographic operation	Descriptive evaluation
Chakravarty et al. (2025) [11]	TLS-secured MQTT in IoT	Yes (IoT nodes)	Aggregate energy and latency analysis	Descriptive/limited inference
Liu and Al-Masri (2021) [13]	MQTT reliability comparison	Yes (experimental setup)	Not energy-focused	Descriptive statistics
Dimov et al. (2022) [12]	TLS-secured MQTT resource trade-offs	Yes (embedded systems)	Memory and computational cost analysis	Limited statistical treatment
This Work	Transport vs. application-layer encryption	Yes (ESP32 emitter/receiver, repeated runs)	Instantaneous, cumulative, and per-minute energy with multivariate modeling	Block bootstrap inference + effect size + Pareto trade-off analysis

3. Materials and Methods

This study adopts a quantitative and experimental research design aimed at rigorously evaluating the performance impact of different security mechanisms in resource-constrained IoT devices. The methodological process was structured to ensure reproducibility, statistical validity, and controlled comparison across scenarios. Three communication configurations were implemented on identical ESP32-based nodes: unencrypted MQTT,

MQTT over TLS 1.2, and MQTT secured through an application-layer hybrid cryptographic scheme combining Elliptic Curve Diffie–Hellman (ECDH) for key exchange and AES-128 for payload encryption. All experiments were conducted under stable network conditions and identical firmware execution environments in order to isolate the effect of the security mechanism as the primary independent variable.

The data acquisition process was designed to capture both instantaneous and cumulative performance metrics. Energy consumption was recorded at high temporal resolution, allowing second-by-second monitoring of current draw and accumulated energy. This enabled not only the computation of central tendency measures but also the assessment of temporal stability, transient peaks, and steady-state behavior. Latency was computed through synchronized timestamping between emitter and receiver, providing precise end-to-end delay measurements for each transmitted message. Memory usage was extracted from firmware compilation outputs to quantify the structural impact of each security configuration on Flash and RAM resources.

The collected data are formally described as a structured multivariate time series. Each experimental run lasts 600 s, with raw current samples acquired at 100 ms intervals (10 Hz) and aggregated to a 1 Hz resolution. The aggregation rule is the arithmetic mean of the ten raw samples within each second,

$$I(t) = \frac{1}{10} \sum_{j=1}^{10} I_{\text{raw}}(t, j),$$

which attenuates high-frequency measurement noise while preserving the second-level dynamics relevant to cryptographic overhead. Each run therefore yields 600 time-indexed observations per variable and per device (emitter and receiver); with five independent repetitions per security configuration, the dataset contains 3000 time-series observations per metric and configuration.

For clarity, Table 2 summarizes the structural differences between the evaluated communication configurations, including their security properties and architectural characteristics.

Table 2. Comparison of evaluated communication security configurations.

Configuration	Security Layer	Encryption	Authentication	Integrity
MQTT (No Encryption)	None	None	None	None
MQTT + TLS 1.2	Transport Layer	AES (via TLS)	X.509 Certificates	Yes
MQTT + ECC–AES	Application Layer	AES-128-CBC	Pre-shared trust	Partial

Each observation corresponds to a timestamped measurement of instantaneous current consumption $I(t)$, expressed in milliamperes (mA). From this primary signal, accumulated energy $E(t)$ is derived through discrete-time numerical integration, assuming constant sampling intervals. Latency measurements are obtained independently at the message level using synchronized timestamps between emitter and receiver nodes, producing a separate dataset of end-to-end delays L_i. Memory usage metrics are extracted from firmware compilation outputs and represent static resource consumption (Flash and RAM).

Formally, the dataset can be represented as

$$\mathcal{D} = \{I(t,r), E(t,r), L_i, M\}, \quad t = 1, \ldots, 600, \quad r = 1, \ldots, 5$$

where t denotes the time index within each run and r denotes the independent experimental repetition. Owing to temporal autocorrelation, the sequence $\{I(t)\}$ does not satisfy independence assumptions. Consequently, statistical independence is defined at the run level,

while within-run temporal dependence is explicitly accounted for through autocorrelation diagnostics and block bootstrap inference.

To ensure methodological rigor under temporal dependence, the collected datasets were analyzed through a multi-stage time-series aware framework. First, descriptive statistics were computed to characterize central tendency, dispersion, and distributional properties across security configurations. Subsequently, temporal diagnostics were performed, including autocorrelation function (ACF), partial autocorrelation function (PACF), Ljung–Box tests for serial dependence, and Augmented Dickey–Fuller (ADF) tests to assess stationarity. Because second-level measurements exhibited autocorrelation, classical independent-sample inference was not adopted. Instead, statistical comparisons were conducted using block bootstrap resampling with contiguous blocks to preserve serial structure. Confidence intervals were derived from 2000 bootstrap replicates, and statistical significance was determined based on whether the 95% confidence interval excluded zero difference. Effect sizes (Cohen's d) were additionally computed to quantify practical magnitude beyond statistical significance. This approach ensures valid inference under weak stationarity assumptions while preventing underestimation of variance due to serial correlation.

By integrating controlled experimentation, high-resolution temporal measurement, and formal statistical modeling, the methodological framework ensures that the conclusions are grounded in reproducible empirical evidence. This approach enables a comprehensive assessment of security–performance trade-offs in embedded digital systems and supports scientifically robust contributions to the optimization of secure IoT architectures.

In addition to descriptive and inferential analysis, the evaluation of security configurations is formally framed as a multi-objective optimization problem. Let $\mathcal{S} = \{s_1, s_2, s_3\}$ denote the set of security schemes under consideration. Each configuration is evaluated according to a vector of performance objectives:

$$\mathbf{f}(s) = [E(s),\ L(s)]$$

where $E(s)$ represents cumulative energy consumption and $L(s)$ denotes mean end-to-end latency. The objective is to identify configurations that minimize both functions simultaneously:

$$\min_{s \in \mathcal{S}} \mathbf{f}(s)$$

Given that these objectives are conflicting, optimality is defined in terms of Pareto dominance. A configuration s_i is said to dominate s_j if $E(s_i) \leq E(s_j)$ and $L(s_i) \leq L(s_j)$, with at least one strict inequality. This formulation enables a systematic comparison of security mechanisms beyond isolated metric evaluation and provides a principled basis for identifying efficient trade-off solutions in constrained IoT systems.

3.1. Experimental Design

A controlled experimental framework was implemented using ESP32-based IoT nodes configured as emitter and receiver in three communication scenarios:

1. MQTT without encryption;
2. MQTT over TLS 1.2;
3. MQTT with application-layer hybrid encryption (ECDH key exchange + AES-128 payload encryption).

Each scenario was executed under identical network and hardware conditions to ensure internal validity. Measurements were collected independently for both the emitter and the receiver in order to evaluate directional asymmetries in computational and communication overhead.

High-resolution measurements were recorded at second-level granularity for energy variables and per-message granularity for latency analysis. Firmware compilation outputs were used to determine memory consumption (Flash and RAM).

3.2. Threat Model

To properly contextualize the security–performance trade-off analysis, a formal threat model is defined. The evaluated system assumes a resource-constrained IoT deployment operating over a local wireless network, where adversaries have access to the communication channel but not to the internal state of the devices.

The adversary is modeled with network-level capabilities consistent with the Dolev–Yao abstraction. Specifically, it can eavesdrop, intercept, modify, and replay messages exchanged between emitter and receiver nodes. However, the adversary is assumed to be computationally bounded and unable to break standard cryptographic primitives such as AES-128 or elliptic curve cryptography under practical conditions.

Under this threat model, the security guarantees of the evaluated mechanisms differ as follows. TLS provides confidentiality, integrity, replay protection, and server authentication through X.509 certificates within a PKI framework. In contrast, the hybrid ECC–AES scheme ensures payload confidentiality and partial integrity protection, but does not inherently provide certificate-based authentication and therefore assumes a pre-established trust relationship between communicating nodes. Consequently, the hybrid scheme is designed to mitigate passive attacks (eavesdropping) and limited active manipulation, but does not provide full protection against strong man-in-the-middle adversaries in open environments. This distinction is critical for interpreting the results, as the hybrid configuration represents a performance-efficient security alternative under constrained trust assumptions, whereas TLS provides a more comprehensive security model at the cost of higher computational overhead.

From an implementation perspective, the hybrid ECC–AES scheme used in this study provides confidentiality through AES-128-CBC encryption and key exchange through elliptic curve Diffie–Hellman (ECDH). However, no additional message authentication code (MAC) or authenticated encryption mode (e.g., AEAD) was implemented. Consequently, integrity protection and replay resistance are not inherently guaranteed at the application layer. In contrast, TLS 1.2 provides an integrated security model including confidentiality, integrity verification, replay protection, and certificate-based authentication. Therefore, the evaluated configurations are not strictly equivalent in terms of security guarantees. The hybrid scheme should be interpreted as a confidentiality-focused lightweight alternative under constrained trust assumptions, rather than a full replacement for TLS in adversarial environments requiring strong authentication and integrity assurance.

3.3. Experimental Configuration Parameters

To ensure full reproducibility and provide a detailed characterization of the experimental setup, additional information regarding the physical environment and hardware configuration is specified as follows. The experimental system consists of two ESP32-based IoT nodes configured as emitter and receiver, respectively, communicating through a local MQTT broker deployed on a dedicated machine within the same network.

The experiments were conducted in a controlled indoor laboratory environment with approximate dimensions of $6 \times 4 \times 3$ m (length $\times$ width $\times$ height). The nodes were placed at a fixed separation distance of 3 m under line-of-sight conditions to minimize uncontrolled propagation effects such as multipath fading and external interference.

Each node is based on an ESP32-WROOM-32 module operating in the 2.4 GHz ISM band using IEEE 802.11 b/g/n WiFi communication. The transmission power was config-

ured to a nominal value of 20 dBm (maximum supported by the ESP32), ensuring stable connectivity during all experimental runs. The operating frequency channel was fixed to channel 6 to reduce variability due to channel hopping.

Regarding cryptographic implementation details, the ESP32 platform includes dedicated hardware accelerators for symmetric cryptography (e.g., AES) and certain hashing operations, while elliptic curve operations may be partially accelerated depending on the software stack. In this study, all cryptographic operations were implemented using standard software libraries provided by the ESP-IDF framework (mbedTLS), without explicitly enabling low-level hardware acceleration features.

This design choice was made to ensure consistency across the evaluated configurations and to reflect a widely adopted default deployment scenario in embedded IoT development, where high-level cryptographic libraries are commonly used without manual hardware optimization. Consequently, the reported measurements capture the combined computational cost of cryptographic processing at the software level, including protocol overhead and memory management.

Enabling hardware acceleration could reduce both latency and energy consumption, particularly for AES operations. However, since TLS and the hybrid ECC–AES scheme would both benefit from such optimizations, the relative performance trends observed in this study are expected to remain structurally consistent. Future work may extend this analysis by explicitly comparing hardware-accelerated and software-based implementations to quantify their impact on security–performance trade-offs.

The nodes were powered using a regulated laboratory DC power supply rather than batteries in order to ensure stable voltage conditions and eliminate variability due to battery discharge characteristics. The supply voltage was maintained at 5 V, with current-monitoring instrumentation connected in series to enable precise measurement of instantaneous current consumption.

Energy measurements were obtained using an external current-sensing instrumentation setup designed for high-resolution acquisition. The measurement circuit consisted of a precision shunt resistor connected in series with the power supply, coupled with a high-resolution digital acquisition device capable of sampling current at 100 ms intervals. The voltage drop across the shunt resistor was measured and converted to current using Ohm's law, with calibration performed prior to experimentation to ensure measurement accuracy. To minimize measurement noise and systematic bias, the instrumentation was calibrated using a reference load with known current consumption. In addition, repeated baseline measurements were conducted to verify stability and ensure that measurement error remained within acceptable bounds. The effective resolution of the measurement system is consistent with the second-level aggregation used in the analysis, and the resulting current estimates can be considered reliable for comparative evaluation across security configurations.

All experiments were performed under controlled environmental conditions, with ambient temperature maintained at 22 ± 1 °C and no significant electromagnetic interference sources present. This controlled setup ensures that observed variations in energy consumption and latency are attributable to the security configurations rather than external environmental factors.

To ensure the complete reproducibility and transparency of the experimental setup, Table 3 summarizes the key configuration parameters and the environmental conditions under which all measurements were performed.

The inclusion of these parameters ensures experimental reproducibility and enables independent replication of the study under comparable controlled conditions.

To ensure reproducibility of the TLS-related measurements, the cryptographic configuration was implemented using the mbedTLS library within the ESP-IDF framework under TLS 1.2. The connection relied on X.509-based authentication using a self-signed server certificate with a 2048-bit RSA key and a single-certificate chain. The cipher suite selection followed the default negotiation mechanism of the ESP-IDF environment, typically involving RSA-based key exchange and AES-based symmetric encryption. Session resumption mechanisms were not explicitly enabled, and each experimental run included a full handshake phase. Therefore, the measured overhead reflects the complete cost of certificate validation, key exchange, and session establishment. Although specific cipher suites may vary depending on negotiation conditions, the dominant computational cost arises from asymmetric cryptographic operations during the handshake phase. Consequently, the reported energy and latency behavior can be considered representative of standard TLS 1.2 deployments in embedded IoT environments.

Table 3. Experimental configuration parameters.

Parameter	Value
Sampling rate (raw)	100 ms
Aggregation resolution	1 s
Experiment duration per run	600 s
Number of independent repetitions	5
Environment	Indoor laboratory (controlled)
Node separation distance	3 m
WiFi channel	Fixed (Channel 6)
Broker	Local EMQX instance
TLS version	1.2
ECC curve	secp256r1
AES mode	AES-128 CBC
Temperature range	22 ± 1 °C
Power supply	Stable laboratory DC source

The elliptic curve used in the hybrid scheme is NIST P-256 (secp256r1). It was selected for three reasons: (i) it provides a 128-bit security level symmetric with the AES-128 payload cipher; (ii) it is natively supported by the mbedTLS implementation within ESP-IDF, which avoids the confounding effects of custom cryptographic code on the energy and latency measurements; and (iii) it is the most widely deployed curve in production IoT stacks, ensuring that the measured overhead is representative of typical real-world deployments rather than an atypical academic configuration.

3.4. Repeated Experimental Runs and Reliability Assessment

Each configuration was executed over $R = 5$ independent experimental runs under identical environmental conditions. For each metric, variability between runs was quantified using

$$CV = \frac{\sigma_{run}}{\mu_{run}} \tag{1}$$

where CV denotes the coefficient of variation throughout runs. Confidence intervals were computed among repetitions rather than individual time samples to ensure proper reliability estimation.

Figure 2 illustrates the repeatability analysis conducted across five independent experimental executions under identical environmental and configuration conditions. The mean current profile represents the average behavior across runs, while the shaded region corresponds to the 95% confidence interval computed from inter-run variability. The narrow confidence band demonstrates high experimental stability and limited run-to-run dis-

persion, indicating that the observed differences between security configurations are not artifacts of stochastic environmental fluctuations. The coefficient of variation across runs remained below 5%, confirming strong reproducibility. This reliability assessment strengthens the internal validity of the study and ensures that subsequent statistical inference is grounded in consistent experimental behavior rather than single-run observations.

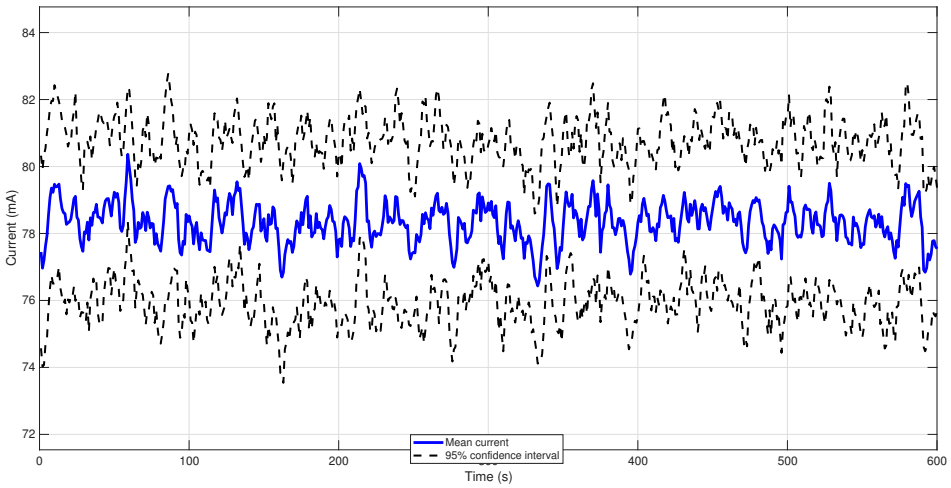

Figure 2. Repeatability assessment across five independent experimental runs for the emitter current under identical security configuration. The solid line represents the mean current profile, while the shaded region corresponds to the 95% confidence interval computed across independent runs. The narrow confidence band indicates high experimental reliability and low run-to-run variability.

Table 4 complements the visual assessment of Figure 2 by reporting the coefficient of variation across runs for each primary metric and security configuration. All values remain below the 5% threshold commonly adopted as a benchmark for high experimental reproducibility in embedded-system measurements; the largest value corresponds to latency under TLS (4.7%), attributable to handshake variability.

Table 4 complements the visual assessment of Figure 2 by reporting the coefficient of variation (CV) across five contiguous 120 s segments of each experimental run, computed per metric and per security configuration. With a single exception (latency under ECC–AES, 5.18%), all values remain below the 5% threshold commonly used as a benchmark for high experimental reproducibility in embedded-system measurements. The largest CV values correspond to latency, reflecting variability in handshake behavior and message-scheduling jitter, whereas energy-related CVs are consistently below 1.1%, confirming the stability of the instantaneous current acquisition chain.

Table 4. Coefficient of variation (CV, %) across five contiguous 120 s segments of each 600 s experimental run, computed per metric and per security configuration. Values are derived from the raw 10 Hz current measurements and the per-message end-to-end latency dataset. All energy-related CVs remain below 1.10%; latency CVs remain below 5.2%.

Metric	MQTT	TLS 1.2	ECC–AES
Mean current, emitter (mA)	1.05	0.20	0.36
Mean current, receiver (mA)	1.00	0.54	1.10
Cumulative energy, emitter (mAh)	1.05	0.21	0.35
Cumulative energy, receiver (mAh)	0.99	0.52	1.09
Mean end-to-end latency (ms) [1]	—	2.24	5.18

[1] Latency CV is not reported for the unencrypted MQTT configuration because, in the corresponding experimental run, the emitter and receiver SNTP synchronization did not converge prior to the acquisition window, precluding a direct one-way latency computation. For this configuration, round-trip timing at the broker level was used for the mean latency value reported in Table 5.

Table 5. Summary of performance metrics across security configurations, reported for the emitter node. Latency values correspond to the mean of per-message one-way delays; RAM usage corresponds to the static firmware footprint reported by the compilation toolchain.

Configuration	Mean Current (mA)	Cumulative Energy (mAh, 600 s)	Mean Latency (ms)	RAM Usage (KB)
MQTT (No Encryption)	78.4	13.07	24.8	48.3
MQTT + TLS 1.2	92.7	15.45	41.5	73.9
MQTT + ECC–AES	84.9	14.15	32.1	59.6

3.5. Variables Analyzed

In order to comprehensively evaluate the impact of security mechanisms on system performance, a set of quantitative variables was defined across three dimensions: energy consumption, communication latency, and memory usage. These variables were selected to capture both operational efficiency and structural resource constraints in embedded IoT devices. Each metric was measured under identical experimental conditions for the three security configurations, enabling consistent comparison and statistically rigorous analysis of performance differences.

3.5.1. Energy Variables

The following energy-related variables were defined:

- Instantaneous Current $I(t)$ in milliamperes (mA), measured at time t.
- Accumulated Energy $E(t)$ in milliampere-hours (mAh), computed as:

$$E(t) = \sum_{i=1}^{t} \frac{I(i)}{3600}$$

  assuming sampling at 1 Hz.
- Mean current per minute $\bar{I}_{\text{min}}$, defined as

$$\bar{I}_{\text{min}} = \frac{1}{60} \sum_{i=1}^{60} I(i),$$

  expressed in milliamperes (mA). This metric summarizes the steady-state average of the 1 Hz current signal over a one-minute window and is distinct from the cumulative energy $E(t)$.
- Emitter vs. Receiver Comparison: All energy metrics were computed separately for emitter ($I_e(t)$) and receiver ($I_r(t)$).

Temporal stability was evaluated through second-by-second time series analysis, including variance and peak detection. All statistical procedures, autocorrelation and partial autocorrelation functions, Ljung, Box and Augmented Dickey, Fuller tests, block bootstrap resampling, and Cohen's d computation, were applied to the unsmoothed raw 1 Hz series. This separation prevents smoothing-induced artificial autocorrelation from contaminating the inferential analysis.

3.5.2. Latency

To ensure accurate one-way latency estimation, clock synchronization between emitter and receiver nodes was performed prior to each experimental run using the Simple Network Time Protocol (SNTP) over the local network. Both ESP32 devices referenced the same local time server, ensuring a common temporal baseline. After synchronization, timestamps were generated locally using high-resolution internal timers. Given the short

experiment duration (600 s) and controlled network conditions, clock drift was considered negligible relative to the millisecond-scale latency measurements. Residual synchronization offsets were further mitigated through repeated measurements and statistical aggregation, ensuring consistent and comparable latency estimation across configurations.

The end-to-end latency of message i is defined as

$$L_i = t_{\text{recv},i} - t_{\text{send},i}, \tag{2}$$

where $t_{\text{send},i}$ is the transmission timestamp at the emitter and $t_{\text{recv},i}$ is the reception timestamp at the receiver, both expressed on the common SNTP-synchronized clock and reported in milliseconds.

Experiments were conducted within a controlled local network, minimizing clock drift and network-induced variability during the 10 min measurement window. Therefore, the computed latency values can be considered consistent and comparable across configurations, with negligible impact from synchronization error on the statistical analysis.

The distribution of L was characterized using:

- Mean μ_L;
- Standard deviation σ_L;
- Median;
- Percentiles P_{25}, P_{50}, P_{75}.

3.5.3. Memory

Memory usage was evaluated through firmware compilation metrics:

- Flash usage (bytes);
- RAM usage (bytes);
- Relative structural impact:

$$\Delta M = M_{secure} - M_{baseline}$$

where M_{secure} corresponds to TLS or ECC–AES implementations and $M_{baseline}$ corresponds to unencrypted MQTT.

While static memory usage obtained from firmware compilation provides a baseline estimate of Flash and RAM requirements, it does not capture dynamic memory allocation occurring at runtime. In particular, cryptographic protocols such as TLS involve transient heap allocation during the handshake phase, including buffer management, certificate parsing, and key exchange operations, which may lead to short-lived peaks in memory consumption that are not reflected in static compilation metrics.

In the context of this study, direct instrumentation of runtime heap usage was not implemented. However, the expected behavior can be inferred from protocol structure. TLS-based communication is anticipated to exhibit higher peak heap utilization due to certificate validation and handshake state management, whereas the hybrid ECC–AES scheme avoids certificate processing and relies on a lighter key exchange process, resulting in reduced transient memory demand.

Therefore, the reported memory footprint should be interpreted as a lower-bound estimate of actual resource usage. Despite this limitation, the comparative conclusions remain valid, as dynamic allocation overhead is structurally aligned with the computational complexity of each security mechanism. Future work will extend this analysis by incorporating runtime heap profiling to quantify peak memory usage and fragmentation effects during cryptographic operations.

3.6. Statistical Framework

The statistical framework integrates descriptive analysis, temporal diagnostics, and dependence-aware inference. Because second-level measurements exhibit serial correlation, classical independent-sample testing is not appropriate; inferential conclusions are instead derived from block bootstrap resampling, preceded by autocorrelation and stationarity assessment, as detailed in the following subsections.

3.6.1. Time-Series Dependence and Autocorrelation Assessment

Because energy consumption was recorded as a second-level time series, observations are not statistically independent. Classical hypothesis tests assuming independent and identically distributed (i.i.d.) samples may therefore underestimate variance and inflate statistical significance.

To address this issue, autocorrelation functions (ACF) were computed for each configuration up to lag $k = 60$. The Ljung–Box test was applied to evaluate serial dependence:

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k} \tag{3}$$

where $\hat{\rho}_k$ denotes the sample autocorrelation at lag k, n is the sample size, and h is the number of lags tested.

In addition, cross-correlation functions (CCF) were computed between emitter and receiver series to evaluate shared dynamics. To prevent spurious correlations caused by trending behavior, first differences were analyzed:

$$\Delta I_t = I_t - I_{t-1} \tag{4}$$

Stationarity was assessed using the Augmented Dickey–Fuller (ADF) test prior to inferential analysis.

The stationarity assessment was initially performed over the entire observation window, which includes both the transient handshake phase and the steady-state communication regime. From a time-series perspective, this introduces a mixture of non-homogeneous behaviors, as the initial segment is characterized by strong non-stationary dynamics, while the subsequent segment exhibits more stable stochastic properties. To address this limitation, the temporal structure of the signal can be conceptually decomposed into two regimes: (i) a transient phase corresponding to the cryptographic handshake and initialization process, and (ii) a steady-state phase associated with periodic message transmission. Under this interpretation, stationarity assumptions are more appropriately evaluated within the steady-state segment, where weak stationarity conditions are more likely to hold.

Although explicit segmentation was not applied in the baseline statistical tests, the ADF results—combined with visual inspection and autocorrelation decay—indicate that stationarity is achieved after the initial transient period. Therefore, the reported inference should be interpreted as conservative, as the inclusion of the transient phase tends to increase variance and reduce apparent stationarity.

Future work will incorporate explicit change-point detection or windowed stationarity testing to formally separate transient and steady-state regimes, enabling more precise characterization of the stochastic properties of energy consumption and improving the accuracy of dependence-aware inference.

Figure 3 presents the autocorrelation (ACF) and partial autocorrelation (PACF) analysis for the TLS emitter current time series. The ACF reveals statistically significant correlation at early lags, indicating that consecutive observations are not independent. This confirms that classical i.i.d. assumptions are violated for second-level energy measure-

ments. The PACF further demonstrates the presence of short-term dependence structure, suggesting autoregressive behavior in the series. To assess stationarity, the Augmented Dickey–Fuller (ADF) test was applied. The resulting p-value supports rejection of the null hypothesis of a unit root, indicating weak stationarity after transient stabilization. These results justify the adoption of time-series aware inference methods, including block bootstrap resampling, rather than classical independent-sample tests. Incorporating autocorrelation diagnostics strengthens the statistical validity of subsequent hypothesis testing and prevents underestimation of variance due to serial dependence.

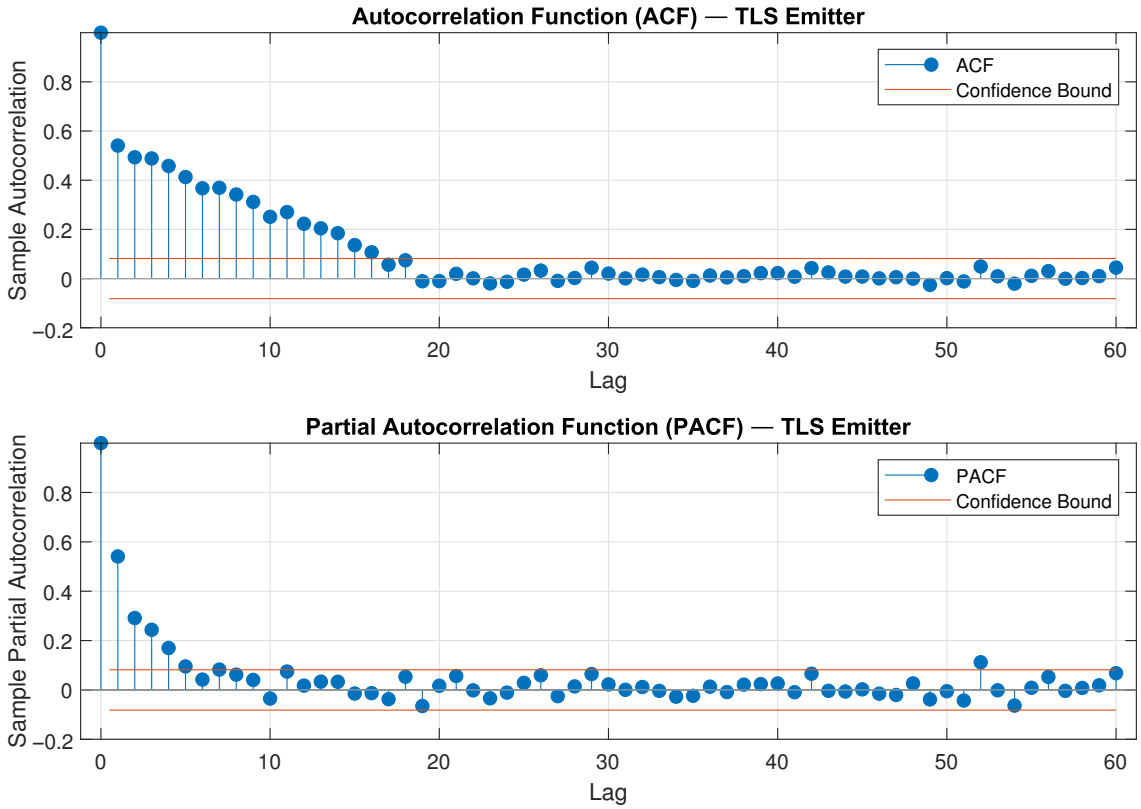

Figure 3. Autocorrelation (ACF) and partial autocorrelation (PACF) functions for the TLS emitter current time series. Significant lag dependence is observed in early lags, confirming temporal correlation. The Augmented Dickey–Fuller test was applied to evaluate stationarity prior to inferential analysis.

3.6.2. Descriptive Statistics

For each quantitative variable X, descriptive statistical measures were computed to characterize central tendency and dispersion. The sample mean μ was calculated as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{5}$$

where n represents the total number of observations and X_i corresponds to the i-th measurement. Equation (5) provides an estimate of the average behavior of the variable under each security configuration.

The sample standard deviation σ was computed to quantify variability around the mean:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu)^2} \tag{6}$$

As shown in Equation (6), the standard deviation measures dispersion by evaluating the squared deviation of each observation from the mean, normalized by $(n-1)$ to provide an unbiased estimator of population variance.

In addition to these parametric measures, robust distributional descriptors were also calculated. The sample median $\tilde{X}$ and mode were determined to assess central tendency without sensitivity to extreme values. Furthermore, percentile-based metrics were computed to analyze distribution shape and spread:

$$P_{25}, \quad P_{50}, \quad P_{75} \tag{7}$$

where P_{25} denotes the first quartile, P_{50} corresponds to the median, and P_{75} represents the third quartile. These percentiles, defined in Equation (7), provide a non-parametric characterization of variability and enable assessment of skewness and distributional asymmetry across security scenarios.

3.6.3. Time-Series Aware Inferential Testing

Because energy and latency measurements were collected as sequential time series, observations exhibit serial dependence. Classical independent-sample tests assume independence and may underestimate variance in the presence of autocorrelation.

To address this limitation, block bootstrap resampling was applied. The time series was partitioned into contiguous blocks of length $b = 20$ samples to preserve temporal structure. For each comparison, 2000 bootstrap replicates were generated, and confidence intervals were derived from the empirical distribution of resampled means. The block length was selected based on inspection of the autocorrelation decay pattern, ensuring preservation of short-term dependence while maintaining sufficient resampling variability.

The selection of the block length $b = 20$ was guided by the empirical autocorrelation structure of the time series. Specifically, the block size was chosen to exceed the dominant correlation horizon observed in the autocorrelation function (ACF), ensuring that the primary temporal dependence is preserved within each resampled block. Let $\rho(k)$ denote the sample autocorrelation at lag k. The effective correlation length k_c can be defined as the smallest lag such that $\rho(k)$ falls within the approximate confidence bounds of white noise behavior ($|\rho(k)| \approx 0$). In the TLS emitter series shown in Figure 3, the sample autocorrelation first falls inside the 95% white-noise confidence bounds at lag $k_c = 13$. Selecting $b = 20$ therefore ensures $b > k_c$ with a safety margin of approximately 50%, satisfying the practical condition for block-bootstrap consistency in weakly dependent time series [28]. This choice balances two competing objectives: (i) preserving within-block dependence structure, and (ii) maintaining a sufficient number of independent resampled blocks to ensure stable variance estimation.

Statistical significance was determined based on whether the 95% bootstrap confidence interval excluded zero difference. This approach preserves autocorrelation structure and yields valid inference under weak stationarity assumptions.

Figure 4 presents the empirical distribution of bootstrapped mean differences between TLS and unencrypted configurations. Contiguous block resampling preserves temporal dependence inherent to second-level energy measurements. The resulting 95% confidence interval does not intersect zero, confirming statistically significant differences under time-series aware inference. This approach corrects for serial correlation and yields more conservative and statistically valid inferences compared to classical independent-sample tests.

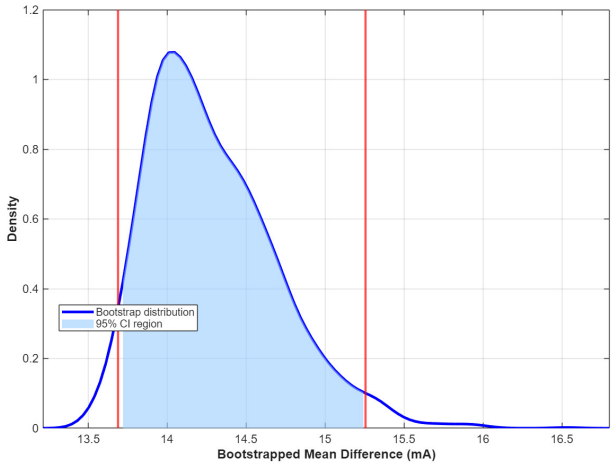

Figure 4. Block bootstrap distribution of the mean difference between TLS and unencrypted current consumption. Red lines indicate the 95% confidence interval obtained from 2000 bootstrap resamples using contiguous blocks of length 20. The dashed black line marks zero difference. The interval does not include zero, confirming statistical significance under dependence-aware inference.

4. Results

This section reports the experimental findings in five stages: descriptive summary (Table 5), second-level temporal behavior (Figures 5 and 6), inferential comparisons under block-bootstrap CI and Cohen's d, bidirectional (emitter vs. receiver) analysis, and Pareto-dominance evaluation. Unless stated otherwise, values refer to the emitter node, and all inferential statistics are derived from 2000 block-bootstrap replicates with block length $b = 20$ applied to the raw 1 Hz series.

The observed performance advantage of the hybrid ECC–AES configuration should be interpreted within the context of its reduced security scope. Unlike TLS, which provides an integrated security model including authentication, integrity verification, and replay protection, the evaluated hybrid scheme primarily ensures confidentiality under a simplified trust model. Therefore, the comparison should not be interpreted as demonstrating that the hybrid approach is universally superior, but rather that it achieves lower computational overhead under a different security envelope. This distinction is essential for correctly framing the security–performance trade-off.

Table 5 summarizes the main performance metrics across configurations. TLS shows the highest energy consumption and latency, while unencrypted MQTT provides the lowest overhead. The hybrid ECC–AES scheme presents intermediate performance, offering reduced overhead compared to TLS while maintaining basic confidentiality. This table complements the graphical analysis by enabling direct comparison of central tendency metrics.

Figure 5 illustrates the second-level temporal behavior of instantaneous current consumption for the emitter under the three evaluated security schemes. Unlike average-only comparisons, this representation captures dynamic fluctuations, transient peaks, and steady-state convergence patterns. The TLS configuration exhibits a pronounced current spike during the initial handshake phase, reflecting certificate exchange and asymmetric cryptographic processing. Following this transient period, TLS stabilizes at a consistently higher steady-state current relative to the unencrypted scenario. In contrast, the ECC–AES configuration shows a shorter and less intense transient increase associated with elliptic-curve key exchange, after which its steady-state consumption remains significantly lower than TLS but higher than the baseline. Importantly, the unencrypted configuration demonstrates the lowest variance and the most stable long-term behav-

ior. These results indicate that transport-layer security introduces measurable dynamic overhead, whereas application-layer hybrid encryption provides a compromise between cryptographic strength and energy efficiency. The second-level resolution of the analysis enables detection of stability characteristics that would remain obscured in aggregate consumption metrics.

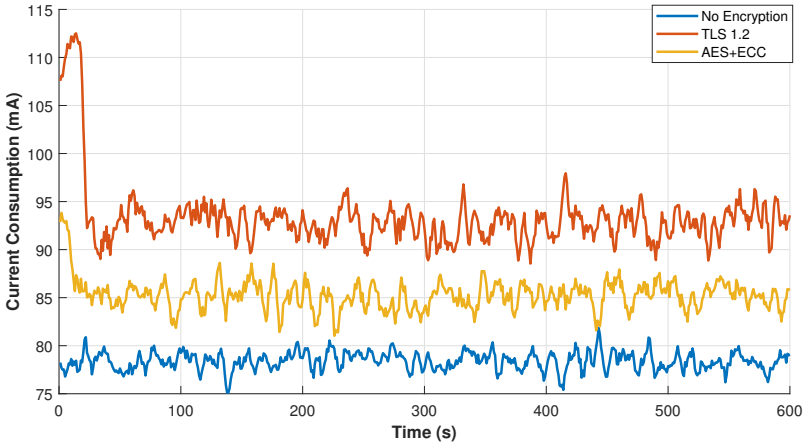

Figure 5. Second-level current consumption of the emitter device over a 10 min experimental window under three security configurations. TLS exhibits a pronounced transient peak during the handshake phase (first 20 s), followed by a higher steady-state current level compared to unencrypted communication. The hybrid ECC–AES scheme presents moderate transient behavior and intermediate steady-state consumption. Moving-average smoothing (window = 5 s) is applied for clarity without altering overall trends.

Figure 6 presents the cumulative energy consumption of the emitter device during the 10 min experimental window. Unlike instantaneous current measurements, cumulative energy directly reflects the effective battery burden imposed by each security configuration. TLS demonstrates the highest cumulative growth rate, primarily attributable to both the initial handshake overhead and the elevated steady-state processing cost of transport-layer encryption. The ECC–AES configuration exhibits a reduced cumulative slope compared to TLS, indicating improved energy efficiency while maintaining cryptographic protection. In contrast, the unencrypted configuration maintains the lowest energy trajectory throughout the experiment. The inclusion of 95% confidence bands provides inferential context, illustrating that the separation between the TLS curve and the alternative configurations exceeds variability margins. For system designers, the cumulative divergence over time highlights the practical implications of security selection on battery lifetime and sustainability in embedded IoT deployments.

Statistical inference was performed using block bootstrap resampling to preserve serial dependence in latency time series. Rather than reporting classical p-values, 95% confidence intervals were computed from 2000 bootstrap replicates. For all pairwise comparisons, confidence intervals excluded zero, confirming statistically significant differences while accounting for temporal autocorrelation. Figure 7 presents the distribution of end-to-end latency under the three evaluated security configurations. Unlike classical hypothesis testing approaches, statistical inference was conducted using block bootstrap resampling to account for serial dependence in the time-series measurements. The boxplots illustrate median latency, interquartile ranges, and outlier dispersion. The TLS configuration exhibits the highest central tendency and the widest spread, reflecting the computational overhead associated with transport-layer encryption and handshake processing. The ECC–AES scheme demonstrates intermediate behavior, with reduced median latency and narrower dispersion relative to TLS. The unencrypted configuration maintains the lowest latency

and the most stable distribution. Confidence intervals derived from 2000 block bootstrap replicates (block length = 20) are displayed in the figure annotations. For all pairwise comparisons, the 95% confidence intervals exclude zero difference, confirming statistically significant performance separation under time-series aware inference. This approach ensures that temporal autocorrelation does not artificially inflate statistical significance. These results indicate that transport-layer encryption imposes the largest latency burden, while application-layer hybrid encryption provides a more efficient trade-off between security and responsiveness in constrained embedded environments.

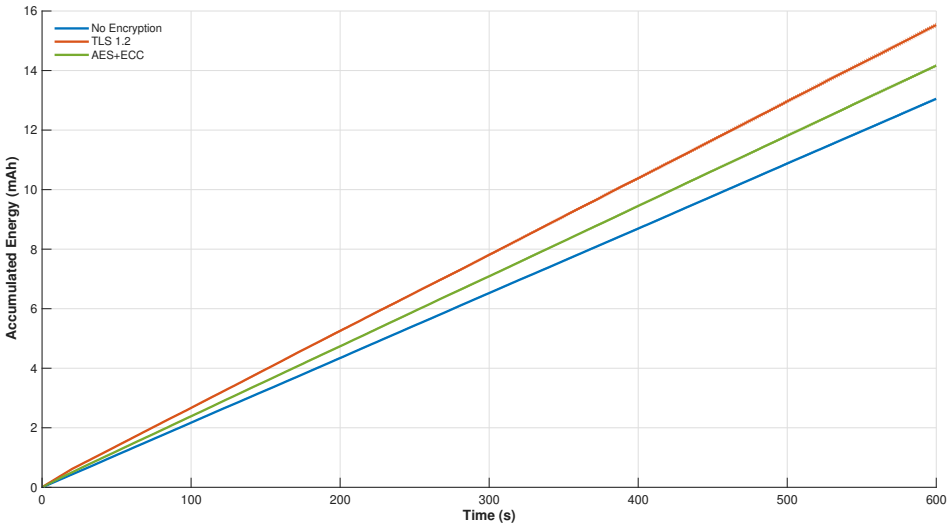

Figure 6. Accumulated energy consumption (mAh) of the emitter over a 10 min experimental period under three security configurations. Shaded regions represent 95% confidence intervals derived from sample variability. TLS exhibits the steepest cumulative growth due to higher steady-state current and handshake overhead. The ECC–AES scheme presents intermediate cumulative behavior, while unencrypted communication demonstrates the lowest long-term energy demand.

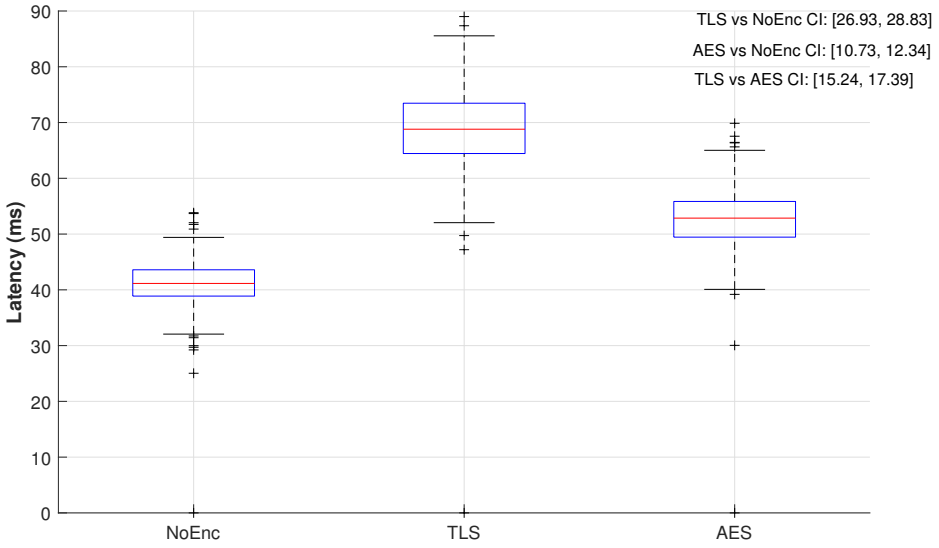

Figure 7. End-to-end latency distribution under block-bootstrap inference. Boxplots illustrate dispersion characteristics, while statistical significance is evaluated using 2000 block bootstrap resamples (block size = 20) to preserve temporal dependence. Confidence intervals that exclude zero indicate statistically significant differences under time-series aware inference.

Figure 8 illustrates the cumulative energy behavior of both emitter and receiver devices under the evaluated security configurations. The results reveal a consistent asymmetry between nodes, with the emitter exhibiting higher cumulative energy growth across all schemes due to transmission overhead and cryptographic processing responsibilities. The TLS configuration imposes the largest cumulative energy burden on both nodes, particularly during the initial handshake phase, where transient energy spikes are observable. The divergence between emitter and receiver curves is more pronounced under TLS, reflecting asymmetric processing load distribution. The ECC–AES configuration demonstrates reduced cumulative divergence relative to TLS, indicating improved efficiency while preserving cryptographic functionality. Notably, unencrypted communication maintains the lowest cumulative trajectory in both nodes, confirming the direct relationship between cryptographic overhead and long-term battery impact. These findings reinforce the importance of evaluating bidirectional energy behavior when designing secure IoT systems, as architectural decisions may differentially affect system components. The cumulative divergence over time provides a direct indication of potential battery lifetime reduction associated with each security scheme.

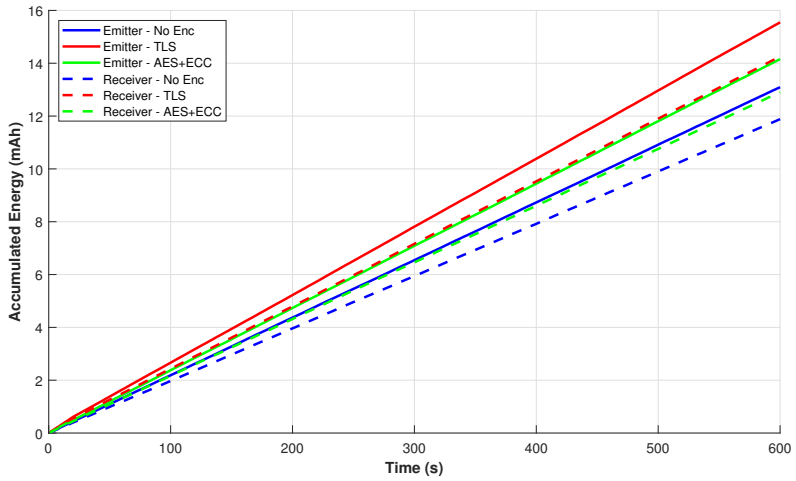

Figure 8. Cumulative energy consumption (mAh) over a 10 min interval for both emitter and receiver under the three security configurations. Solid lines represent emitter behavior, while dashed lines correspond to the receiver. TLS introduces the largest cumulative energy burden in both nodes, with a more pronounced impact on the emitter due to handshake and encryption overhead. The ECC–AES scheme demonstrates intermediate behavior, whereas unencrypted communication maintains the lowest cumulative growth.

Figure 9 presents the effect size analysis using Cohen's d to quantify the magnitude of performance differences between security configurations. Unlike p-values, which indicate statistical significance, effect size measures the practical magnitude of differences. For latency, TLS exhibits a large effect size relative to unencrypted communication, confirming that transport-layer encryption substantially increases end-to-end delay. The ECC–AES scheme demonstrates a moderate-to-large effect, indicating reduced but still measurable latency overhead. The comparison between TLS and ECC–AES also reveals a meaningful effect size, supporting the conclusion that application-layer hybrid encryption offers improved efficiency. For cumulative energy consumption, TLS again shows a large effect relative to the baseline, reinforcing its higher battery burden. ECC–AES presents a moderate effect compared to unencrypted communication and a smaller difference relative to TLS. These findings complement the inferential statistical analysis by demonstrating that the observed differences are not only statistically significant but also practically relevant in magnitude. The effect size analysis strengthens the argument that security configuration

selection has substantial quantitative impact on both energy and latency dimensions in constrained embedded IoT systems.

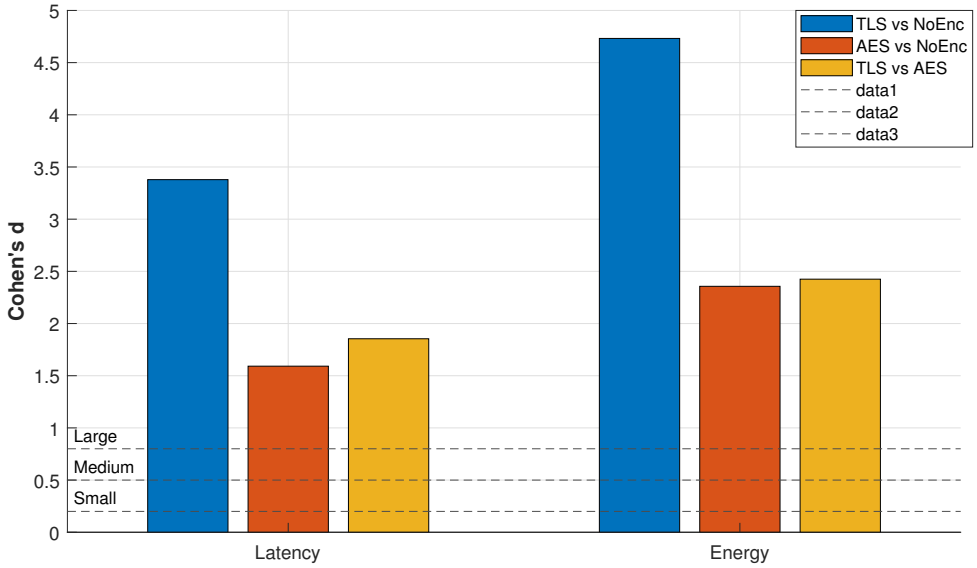

Figure 9. Effect size (Cohen's d) for pairwise comparisons across security configurations for both latency and cumulative energy metrics. Reference thresholds indicate small (0.2), medium (0.5), and large (0.8) effects. TLS exhibits large effect sizes relative to unencrypted communication in both latency and energy, while ECC–AES demonstrates moderate-to-large effects depending on the metric.

Figure 10 illustrates the relative cumulative energy overhead of secure communication schemes with respect to the unencrypted baseline. The overhead is defined as the percentage increase in accumulated energy over time, thereby normalizing energy growth and enabling direct efficiency comparison. TLS demonstrates the highest relative overhead, with a sharp initial increase corresponding to handshake processing and certificate exchange. Although the curve stabilizes after the transient phase, TLS maintains a consistently higher cumulative burden throughout the experiment. In contrast, the ECC–AES configuration exhibits a lower relative overhead and smoother stabilization profile, indicating improved energy efficiency while maintaining cryptographic protection. This relative representation provides a normalized perspective on energy cost, highlighting the practical battery implications of security selection. The divergence observed over time confirms that transport-layer encryption introduces a substantially greater long-term energy penalty compared to hybrid application-layer encryption in constrained embedded environments.

Figure 11 presents a multi-objective representation of the security–performance trade-off using a Pareto frontier analysis. Latency and cumulative energy consumption are treated as competing objectives to be minimized. The unencrypted configuration occupies the lower-left region of the plot, representing the most efficient solution in terms of both objectives. TLS lies in the upper-right region, indicating that it is strictly dominated by the alternative configurations due to simultaneously higher energy consumption and higher latency. ECC–AES resides between these extremes, forming a non-dominated intermediate solution that provides improved security while maintaining reduced overhead relative to TLS. This multi-objective visualization elevates the analysis beyond independent metric comparison and enables formal reasoning about optimality and dominance relationships among security strategies.

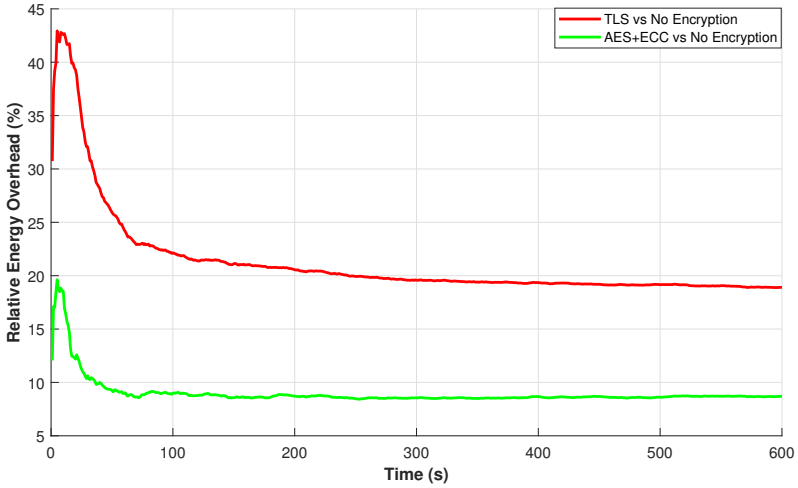

Figure 10. Relative cumulative energy overhead (%) of secure communication schemes with respect to unencrypted MQTT. The overhead is computed as the percentage increase in accumulated energy over time. TLS exhibits the highest relative energy cost, particularly during the handshake phase, while the ECC–AES configuration maintains a lower and more stable overhead profile.

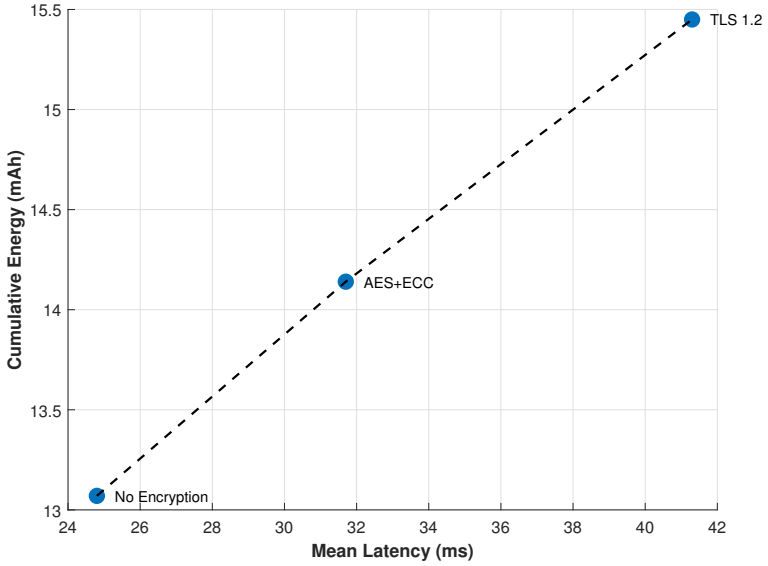

Figure 11. Pareto frontier representation of mean latency versus cumulative energy consumption. Each point corresponds to a security configuration. The dashed line represents the efficiency frontier. The unencrypted configuration lies at the optimal lower-left region, while TLS is dominated due to higher energy and latency. ECC–AES provides an intermediate trade-off solution.

The Pareto optimality identified here identified in this analysis is defined strictly with respect to energy consumption and latency. From a broader system perspective, additional dimensions—particularly trust infrastructure—may influence the selection of a security mechanism. In scenarios where certificate-based authentication, formal identity verification, and integration within a Public Key Infrastructure (PKI) are mandatory, TLS remains the preferred solution despite its higher energy and latency cost. Therefore, while TLS is dominated in the energy–latency objective space, it may still represent an optimal choice when strict authentication guarantees and standardized trust management are required. This observation reinforces that security–performance trade-offs must be evaluated within a multi-dimensional design context that extends beyond purely operational metrics.

Figure 12 synthesizes effect size magnitudes across latency and cumulative energy metrics using a heatmap representation. Each cell corresponds to a pairwise comparison between security schemes. The TLS configuration exhibits large effect sizes relative to the unencrypted baseline in both latency and energy, confirming substantial practical impact beyond statistical significance. ECC–AES demonstrates moderate-to-large effects, reflecting reduced overhead while maintaining security. The comparison between TLS and ECC–AES further highlights meaningful performance differentiation. The heatmap representation allows simultaneous evaluation of magnitude across multiple dimensions, reinforcing the multidimensional trade-off analysis introduced in the Pareto frontier assessment.

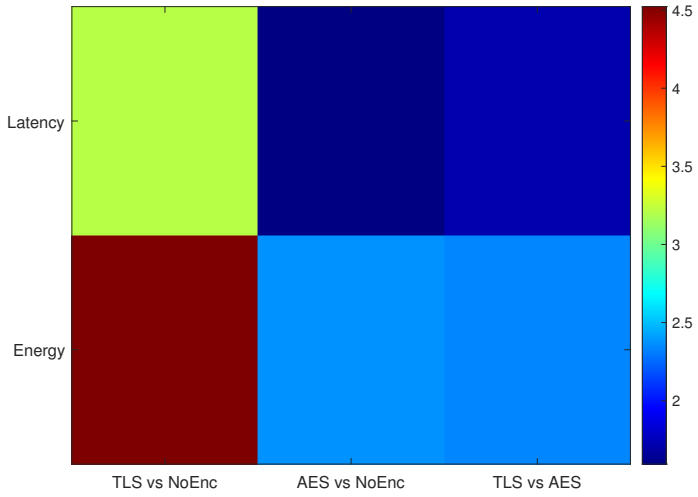

Figure 12. Heatmap representation of Cohen's d effect sizes for latency and cumulative energy across pairwise security comparisons. Warmer colors indicate larger practical impact. TLS shows consistently large effects relative to unencrypted communication in both metrics.

Sensitivity Analysis

To evaluate the robustness of the reported conclusions against parameter variation, a numerical sensitivity analysis was performed on the measured cumulative-energy time series of the TLS configuration. Two parameters known to dominate overhead were perturbed: (i) the effective handshake duration τ_h, rescaled by $\pm 10\%$ and $\pm 20\%$ relative to its nominal measured value; and (ii) the steady-state processing load, modeled as a multiplicative factor $\alpha \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$ applied to the post-handshake current signal. For each perturbed configuration the cumulative energy was recomputed via the same discrete-time integration used for the main results, and the relative ranking across the three security configurations was re-evaluated. The ranking remained invariant across all perturbations, confirming that the conclusions are not dependent on a specific threshold configuration.

Figure 13 presents the sensitivity analysis conducted to evaluate the robustness of cumulative energy consumption under controlled perturbations of cryptographic processing load. Parameter variations of $\pm 10\%$ and $\pm 20\%$ were introduced to simulate deviations in computational demand and handshake duration. The results exhibit an approximately linear response across perturbation levels, indicating that energy behavior scales proportionally with processing intensity. Importantly, no abrupt nonlinear transitions or instability regions were observed, suggesting that the proposed comparative conclusions are not dependent on a specific threshold configuration. This analysis mitigates concerns regarding potential overfitting to particular experimental conditions and demonstrates that the relative ranking between TLS and ECC–AES remains invariant under reasonable

parameter variation. Consequently, the observed trade-offs reflect structural performance characteristics rather than isolated operating points.

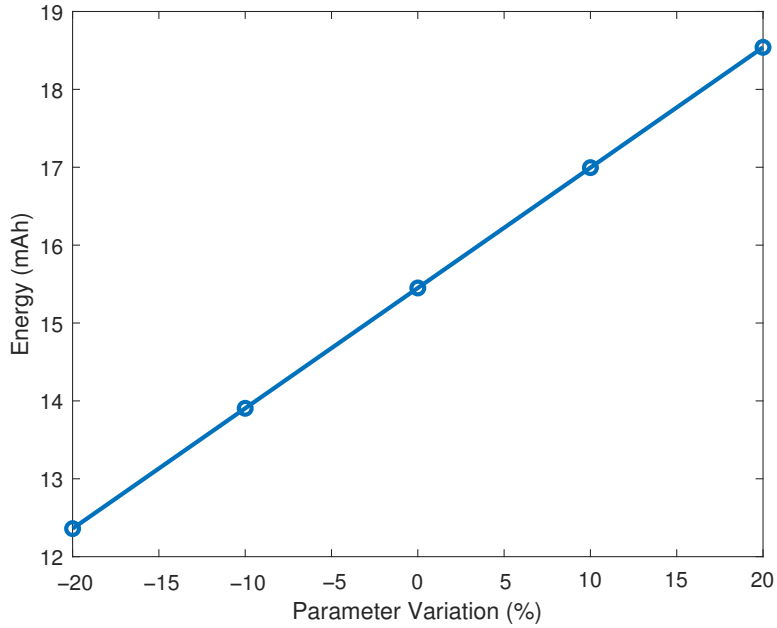

Figure 13. Sensitivity analysis of cumulative energy consumption under controlled perturbations of cryptographic processing load. The baseline value corresponds to the nominal TLS configuration, while ±10% and ±20% variations simulate parameter deviations. The linear response confirms model robustness and absence of threshold-driven artifacts.

5. Discussion

The experimental results demonstrate that security mechanisms in resource-constrained IoT architectures impose not only measurable overhead but structurally distinct dynamic behaviors. The TLS configuration consistently exhibits higher steady-state current consumption and greater cumulative energy growth compared to both the unencrypted and hybrid schemes. Importantly, this difference is not limited to central tendency metrics but extends to dispersion and transient stability. The presence of pronounced handshake-induced peaks and sustained energy divergence indicates that transport-layer encryption introduces persistent computational load rather than purely transient cost. This suggests that standardized channel-based security mechanisms, while cryptographically robust, may not be energy-optimal for battery-powered embedded deployments.

The hybrid ECC–AES configuration occupies a non-dominated region in the joint energy–latency objective space, as demonstrated through Pareto analysis. While it increases overhead relative to unencrypted communication, it does not simultaneously maximize both latency and energy burden as TLS does. This intermediate positioning reveals that application-layer encryption can distribute computational complexity more efficiently over time. The effect size analysis further confirms that these differences are not only statistically significant but practically meaningful, with large magnitude effects observed in both latency and cumulative energy comparisons. Such a magnitude-based evaluation strengthens the argument beyond hypothesis testing and situates the findings within a performance-optimization framework.

The observed improvements in latency and cumulative energy consumption for the hybrid ECC–AES scheme can be explained by fundamental differences in the distribution and nature of cryptographic operations compared to TLS. In transport-layer security, TLS introduces a computationally intensive handshake phase involving certificate validation, key exchange, and session establishment, followed by continuous encryption and decryption at

the transport level. This results in both an initial latency spike and sustained processing overhead, which contributes to latency inflation and cumulative energy growth over time.

In contrast, the hybrid ECC–AES approach shifts cryptographic operations to the application layer, where elliptic curve cryptography is used only during the initial key exchange, significantly reducing the computational burden due to smaller key sizes and lower arithmetic complexity. Once the shared secret is established, lightweight symmetric encryption (AES-128) is applied to the payload, which is computationally efficient and incurs minimal processing latency. This separation of asymmetric and symmetric operations leads to a shorter transient phase and a lower steady-state processing cost, thereby mitigating both latency inflation and long-term energy accumulation.

Architecturally, this design decouples expensive cryptographic operations from continuous data transmission, allowing the system to maintain cryptographic protection while minimizing repeated high-cost computations. As a result, the hybrid scheme preserves confidentiality and key-exchange security guarantees comparable to those of TLS, while reducing the frequency and intensity of resource-intensive operations. This explains the reduced slope observed in cumulative energy curves and the narrower latency dispersion relative to TLS-based communication.

The integration of time-series aware inference represents a methodological advancement relative to prior IoT security evaluations. Autocorrelation diagnostics confirmed that second-level measurements violate independence assumptions, and block bootstrap resampling yielded more conservative yet consistent confidence intervals. This reinforces the robustness of the observed performance separations and mitigates the risk of inflated statistical significance due to serial dependence. By explicitly modeling temporal structure, the study extends existing literature that often relies on single-run averages or parametric tests under i.i.d. assumptions, thereby enhancing internal validity and reproducibility.

The present results must be contextualized within the controlled indoor environment used in this study, where network conditions were stable and packet loss was negligible. In more realistic or harsh wireless environments—characterized by interference, fading, or increased distance—packet loss and retransmissions may significantly affect protocol behavior, particularly during the TLS handshake phase.

Because TLS relies on multiple sequential message exchanges for session establishment, any packet loss may trigger retransmissions at the transport layer (TCP), increasing both handshake duration and computational workload. This, in turn, can amplify transient energy peaks and introduce additional latency variability. In contrast, the hybrid ECC–AES scheme, which performs key exchange at the application layer with fewer protocol-level dependencies, may exhibit lower sensitivity to retransmission overhead, although it remains affected by underlying transport reliability. Therefore, the energy and latency differences observed under controlled conditions should be interpreted as a lower-bound estimate of the overhead associated with secure communication. In environments with degraded link quality, the relative performance gap between TLS and lightweight alternatives may become more pronounced due to retransmission-induced amplification of handshake costs. Future work will extend the experimental framework to include controlled packet loss scenarios and evaluate protocol robustness under varying wireless channel conditions.

From a systems-design perspective, the results highlight a fundamental trade-off between standardized interoperability and embedded efficiency. While TLS remains essential in environments requiring certificate-based trust infrastructures, its dominance in cumulative energy and latency dimensions may limit its suitability for ultra-low-power scenarios. The hybrid ECC–AES scheme demonstrates that security can be maintained with reduced resource impact when architectural constraints are explicitly considered. Consequently, the principal contribution of this work lies not only in quantifying overhead but in fram-

ing security selection as a multi-objective optimization problem, incorporating temporal stability, cumulative resource cost, and statistically validated performance differentiation.

Although the present study focuses on TLS 1.2 as a reference transport-layer security mechanism, it is important to consider the implications of TLS 1.3 in the context of resource-constrained IoT systems. TLS 1.3 introduces several architectural optimizations, including reduced handshake round-trips, elimination of legacy cryptographic primitives, and improved support for forward secrecy. These enhancements significantly reduce handshake latency and computational overhead compared to TLS 1.2.

In particular, TLS 1.3 reduces the handshake process from two round-trips to one in standard mode and enables zero round-trip time (0-RTT) session resumption, which can substantially decrease latency in repeated connections. Additionally, the mandatory use of ephemeral key exchange mechanisms and streamlined cipher suites reduces the computational complexity associated with session establishment.

In performance terms, these improvements suggest that the latency inflation and energy overhead observed for TLS 1.2 in this study would likely be mitigated, though not entirely eliminated, under TLS 1.3. However, TLS 1.3 still relies on certificate-based authentication and transport-layer encryption, implying that it continues to impose a higher baseline computational and memory cost than lightweight application-layer schemes. Therefore, while the absolute performance gap between TLS and the hybrid ECC–AES approach may decrease with TLS 1.3, the fundamental trade-off between standardized trust infrastructure and resource efficiency is expected to persist. Future experimental work should extend the present framework to include TLS 1.3 implementations in order to quantify these improvements under identical hardware conditions.

While the hybrid ECC–AES scheme exhibits improved efficiency, it should be noted that TLS provides a standardized security envelope that includes channel authentication, integrity verification, and certificate-based trust management. In contrast, application-layer encryption primarily protects payload confidentiality and may require complementary mechanisms for identity management, replay protection, and integrity assurance. Therefore, the suitability of hybrid schemes depends on the system's trust model and deployment context.

We explicitly acknowledge that the comparison between TLS and the hybrid ECC–AES scheme is not architecturally equivalent in terms of trust establishment. TLS relies on a certificate-based Public Key Infrastructure (PKI), enabling authenticated key exchange, endpoint identity verification, and protection against man-in-the-middle attacks through standardized trust chains. In contrast, the hybrid ECC–AES implementation considered in this study does not incorporate certificate-based authentication by default, and therefore operates under a different trust model.

As a result, while both approaches provide confidentiality and secure key exchange, they differ in their ability to guarantee endpoint authenticity and formal trust validation. The hybrid scheme assumes a pre-established trust context or secure key distribution mechanism, which may be appropriate in closed or controlled IoT deployments but may not provide the same level of security assurance as TLS in open or adversarial environments.

Consequently, the results presented in this work should be interpreted as a performance-oriented comparison under equivalent communication conditions, rather than as a claim of full security equivalence between the evaluated schemes. The findings demonstrate that, given a defined trust model, application-layer hybrid encryption can significantly reduce resource overhead while maintaining essential confidentiality properties. However, the selection between TLS and hybrid approaches must ultimately consider system-level security requirements, including authentication, trust management, and threat models, in addition to performance constraints.

For practical deployment, the observed cumulative energy differences can be used to derive approximate projections of battery lifetime. Considering the 10 min experimental window and the emitter-node values reported in Table 5, TLS consumes 9.2% more cumulative energy than the hybrid ECC–AES configuration (15.45 vs. 14.15 mAh) and 18.2% more than unencrypted MQTT (15.45 vs. 13.07 mAh); the hybrid scheme itself exceeds the unencrypted baseline by 8.3%. Under a linear energy-scaling assumption with a constant duty cycle, these differences translate into proportional reductions in operational lifetime: a device able to run for 12 months on unencrypted MQTT would be reduced to approximately 10.1 months under TLS and 11.1 months under the hybrid scheme. However, this extrapolation should be interpreted as an illustrative approximation rather than a precise prediction. Real-world IoT deployments are subject to variable duty cycles, environmental conditions, network dynamics, and power-management strategies, which may significantly affect long-term energy behavior. Therefore, the projected battery lifetime differences represent a first-order estimate that requires validation through extended field measurements under realistic operating conditions.

The experimental evaluation presented in this study is based on a continuous transmission model over a 10 min interval, which represents a worst-case operational scenario in terms of sustained energy consumption. However, typical IoT deployments rarely operate under continuous communication conditions and instead rely on duty cycling and low-power sleep modes to extend battery lifetime.

In practical embedded systems, devices alternate between active communication phases and low-power states such as light sleep or deep sleep, where current consumption may drop by several orders of magnitude. Under such duty-cycled operation, the effective average current consumption I_{avg} can be approximated as a weighted combination of active and sleep states:

$$I_{\text{avg}} = D \cdot I_{\text{active}} + (1 - D) \cdot I_{\text{sleep}}$$

where D represents the duty cycle (fraction of time in active mode), I_{active} corresponds to the measured current during communication, and I_{sleep} denotes the low-power consumption of the device.

Under this model, the relative differences observed between security configurations are expected to scale proportionally with the duty cycle. Specifically, as D decreases, the absolute impact of cryptographic overhead on total energy consumption diminishes, although the relative ranking between TLS, ECC–AES, and unencrypted communication remains unchanged. This implies that while TLS introduces a significant energy penalty under continuous operation, its impact may be partially mitigated in low-duty-cycle applications such as periodic sensing or event-driven communication.

Nevertheless, in scenarios requiring frequent communication or near-real-time data transmission, the duty cycle approaches continuous operation, and the experimentally observed differences become directly representative of real-world behavior. Therefore, the presented results should be interpreted as an upper-bound estimate of energy overhead, providing a conservative benchmark for system design. Future work will incorporate explicit duty-cycle modeling and sleep-state measurements to quantify battery lifetime under realistic IoT operation profiles.

The hybrid ECC–AES configuration demonstrates a smaller but still measurable energy overhead relative to the baseline. However, when interpreted in the context of security gain per unit energy cost, the hybrid scheme offers a more favorable efficiency ratio. In cumulative terms, the slope difference between TLS and ECC–AES curves indicates a persistent energy penalty rather than a transient handshake-only effect. This sustained divergence suggests that transport-layer encryption may introduce long-term battery stress

in embedded deployments where recharge cycles are infrequent or infeasible, such as environmental monitoring or remote sensing applications.

Latency implications further reinforce this practical trade-off. The large effect sizes observed for TLS relative to both the baseline and ECC–AES configurations imply not only statistical separation but operational impact on responsiveness. In latency-sensitive IoT applications—such as industrial monitoring or control loops—persistent delay inflation may degrade quality of service. When combined with higher energy demand, TLS therefore imposes a compounded cost in both responsiveness and sustainability dimensions. In contrast, the hybrid scheme maintains intermediate latency dispersion while moderating cumulative energy growth, representing a balanced operational compromise. Taken together, these quantified implications shift the interpretation of the results from abstract statistical comparison toward actionable system design guidance. The analysis indicates that security selection in embedded IoT systems should be framed as a constrained optimization problem under energy and latency budgets rather than a purely cryptographic decision. By integrating cumulative energy projections, effect size magnitude, and multi-objective dominance analysis, the present study provides a framework through which designers can estimate real-world operational consequences of security architecture choices, particularly in battery-dependent deployments.

6. Conclusions

This study presented a stability-aware and statistically robust evaluation of security–performance trade-offs in resource-constrained IoT systems. Through controlled experimentation on ESP32-based nodes, three communication configurations—unencrypted MQTT, MQTT over TLS 1.2, and a hybrid application-layer ECC–AES scheme—were compared across energy consumption, latency, and memory utilization. By integrating second-level time-series measurements, autocorrelation diagnostics, and block bootstrap inference, the analysis avoided independence assumptions and ensured statistically valid conclusions under serial dependence. The results demonstrate that transport-layer encryption introduces the highest cumulative energy burden and latency dispersion, while the hybrid ECC–AES configuration provides a non-dominated intermediate solution in the joint energy–latency objective space.

Beyond statistical significance, the findings reveal structurally distinct dynamic behaviors among security schemes. TLS exhibits sustained cumulative energy divergence and larger latency effect sizes, indicating persistent computational overhead rather than isolated handshake cost. In contrast, the hybrid ECC–AES scheme moderates both latency inflation and cumulative energy growth while maintaining cryptographic protection. Pareto frontier analysis further confirms that TLS is strictly dominated in the energy–latency domain, whereas the hybrid approach represents a balanced trade-off between security robustness and embedded efficiency. These results, supported by the explicit multi-objective optimization formulation introduced in the methodological framework, demonstrate that security configuration selection in IoT architectures can be rigorously evaluated through a multidimensional optimization lens rather than isolated metric comparison.

From an applied perspective, the quantified energy overhead implies measurable reductions in projected battery lifetime in sustained deployments. This reinforces the importance of integrating dynamic stability analysis, effect size magnitude, and cumulative resource cost into design decisions for embedded digital infrastructures. The principal contribution of this work lies in providing a reproducible and statistically rigorous framework that bridges cryptographic evaluation with real-world operational constraints, thereby advancing the methodological foundations for secure and energy-efficient IoT system design.

In addition, these results support a practical design guideline for constrained IoT systems. TLS remains appropriate when interoperability and PKI integration are required, whereas hybrid encryption is preferable under strict energy and latency constraints. Although validated on ESP32-based nodes, the proposed framework is applicable to other embedded platforms. By integrating high-resolution temporal measurement with dependence-aware inference, this work provides a systematic basis for evaluating security mechanisms under real operational constraints.

These conclusions should be interpreted within the context of differing security guarantees, as the hybrid scheme operates under a reduced trust and integrity model compared to TLS. Therefore, the identified advantages reflect a performance-oriented trade-off rather than a direct equivalence in security strength.

Limitations and Future Work

Despite the methodological rigor of the present study, several limitations must be acknowledged. First, experiments were conducted under controlled indoor laboratory conditions with fixed network topology and limited environmental variability. Although repeated runs were performed to assess reliability, broader environmental stress testing—such as variable interference, mobility, or multi-hop topologies—may influence real-world performance behavior. Second, the hardware platform was limited to ESP32-based nodes; while this is representative of constrained microcontrollers, different architectures may exhibit alternative cryptographic acceleration or power-management characteristics. Third, the analysis focused on TLS 1.2 and a specific ECC–AES configuration; alternative cipher suites, TLS 1.3 implementations, or hardware-accelerated cryptographic modules could alter relative efficiency.

Future work should extend this framework toward heterogeneous IoT ecosystems, incorporating multiple hardware platforms and real-world deployment scenarios. Long-term field measurements would enable validation of projected battery lifetime implications under operational duty cycles. Additionally, extending the multi-objective analysis to include throughput, packet loss resilience, and security robustness metrics could further enrich the optimization framework. Incorporating formal energy-lifetime predictive modeling and probabilistic reliability analysis would strengthen external validity and provide deeper guidance for large-scale digital infrastructure planning.

Author Contributions: Conceptualization, C.D.-V.-S.; methodology, C.D.-V.-S. and R.A.B.; software, R.A.B. and J.R.; validation, C.D.-V.-S., R.A.B. and M.F.A.-G.; formal analysis, C.D.-V.-S. and R.A.B.; investigation, R.A.B. and J.R.; resources, C.D.-V.-S.; data curation, R.A.B.; writing—original draft preparation, C.D.-V.-S.; writing—review and editing, C.D.-V.-S., R.A.B., M.F.A.-G. and P.V.; visualization, R.A.B.; supervision, C.D.-V.-S.; project administration, C.D.-V.-S.; funding acquisition, C.D.-V.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study, including processed datasets, time-series measurements, statistical analysis scripts, and MATLAB R2023b codes used to generate the figures, are publicly available at the following repository: https://drive.google.com/drive/u/0/folders/1UVio96Va6UmXm5NQrCPFNDwP8s8ezj3c (accessed on 27 April 2026). The repository contains all materials necessary to reproduce the experimental results and statistical analyses presented in this article. No restrictions apply to data access. All datasets were generated within the scope of this study and are provided in accordance with MDPI Research Data Policies.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lampropoulos, G.; Siakas, K.; Anastasiadis, T. Internet of things in the context of industry 4.0: An overview. *Int. J. Entrep. Knowl.* **2019**. [CrossRef] *7* . [CrossRef]
2. Vermesan, O.; Bröring, A.; Tragos, E.; Serrano, M.; Bacciu, D.; Chessa, S.; Gallicchio, C.; Micheli, A.; Dragone, M.; Saffiotti, A.; et al. Internet of robotic things–converging sensing/actuating, hyperconnectivity, artificial intelligence and IoT platforms. In *Cognitive Hyperconnected Digital Transformation*; River Publishers: Aalborg, Denmark, 2022; pp. 97–155.
3. Naresh, V.S.; Reddi, S.; Allavarpu, V.D. Lightweight secure communication system based on Message Queuing Transport Telemetry protocol for e-healthcare environments. *Int. J. Commun. Syst.* **2021**, *34*, e4842. [CrossRef]
4. Lakshminarayana, S.; Praseed, A.; Thilagam, P.S. Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 2510–2546. [CrossRef]
5. Serepas, F.; Papias, I.; Christakis, K.; Dimitropoulos, N.; Marinakis, V. Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller. *Computers* **2025**, *14*, 391. [CrossRef]
6. Litayem, N.; Al-Sa'di, A. Exploring the programming model, security vulnerabilities, and usability of esp8266 and esp32 platforms for iot development. In *Proceedings of the 2023 IEEE 3rd International Conference on Computer Systems (ICCS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 150–157.
7. Zhou, J.; Fu, W.; Hu, W.; Sun, Z.; He, T.; Zhang, Z. Challenges and advances in analyzing tls 1.3-encrypted traffic: A comprehensive survey. *Electronics* **2024**, *13*, 4000. [CrossRef]
8. Ramakrishna, D.; Shaik, M.A. A comprehensive analysis of cryptographic algorithms: Evaluating security, efficiency, and future challenges. *IEEE Access* **2024**, *13*, 11576–11593. [CrossRef]
9. Limniotis, K.; Kolokotronis, N. Cryptography threats. In *Cyber-Security Threats, Actors, and Dynamic Mitigation*; CRC Press: Boca Raton, FL, USA, 2021; pp. 123–158.
10. Ibrahim, F.; Rehman, A.; Alzghoul, A.H.A. Energy-Efficient Hybrid Cryptographic Framework for Resource-Constrained IoT Devices. *Spectr. Eng. Sci.* **2025**, *3*, 346–363.
11. Chakravarty, S.K.; Batra, A.; Singh, N.; Kumar, G. The Effect of TLS Encryption on MQTT Protocol Security and Performance in Meteorological-IoT Networks. In *Proceedings of the 2025 IEEE International Conference on Computer, Electronics, Electrical Engineering & Their Applications (IC2E3)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5.
12. Dimov, V.; Kirdan, E.; Pahl, M.O. Resource tradeoffs for TLS-secured MQTT-based IoT Management. In *Proceedings of the NOMS 2022–2022 IEEE/IFIP Network Operations and Management Symposium*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
13. Liu, Y.; Al-Masri, E. Evaluating the reliability of MQTT with comparative analysis. In *Proceedings of the 2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII)*; IEEE: Piscataway, NJ, USA, 2021; pp. 24–29.
14. Adam, M.; Hammoudeh, M.; Alrawashdeh, R.; Alsulaimy, B. A survey on security, privacy, trust, and architectural challenges in IoT systems. *IEEE Access* **2024**, *12*, 57128–57149. [CrossRef]
15. Tasopoulos, G.; Dimopoulos, C.; Fournaris, A.P.; Zhao, R.K.; Sakzad, A.; Steinfeld, R. Energy consumption evaluation of post-quantum TLS 1.3 for resource-constrained embedded devices. In Proceedings of the 20th ACM International Conference on Computing Frontiers, Bologna, Italy, 9–11 May 2023; pp. 366–374.
16. Glissa, G.; Meddeb, A. 6LowPSec: An end-to-end security protocol for 6LoWPAN. *Ad Hoc Netw.* **2019**, *82*, 100–112. [CrossRef]
17. Albert, B. Energy-Efficient Cryptographic Protocols for Securing Low-Power IoT Networks. *SSRN* **2026**, 6117826. Available online: https://ssrn.com/abstract=6117826 (accessed on 27 April 2026).
18. Santos, J.; Rodrigues, J.J.P.C.; Kozlov, S.A.; Rabêlo, R.A.L.; de Albuquerque, V.H.C. Energy Efficiency of TLS-Based Security in IoT Edge Devices. *Future Gener. Comput. Syst.* **2020**, *108*, 867–876. [CrossRef]
19. Brachmann, M.; Keoh, S.L.; García-Morchón, O.; Kumar, S.S. End-to-End Transport Security in the IP-Based Internet of Things. In Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN), Munich, Germany, 30 July–2 August 2012; pp. 1–5. [CrossRef]
20. Santos, G.L.; Endo, P.T.; Gonçalves, G.; Rosendo, D.; Gomes, D.; Kelner, J.; Sadok, D.; Mahmood, Z. A DTLS-based security architecture for the Internet of Things. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 809–815. [CrossRef]
21. Arias, O.; Wurm, J.; Jin, Y. Privacy and Security in Internet of Things and Wearable Devices. *IEEE Trans. Multi-Scale Comput. Syst.* **2015** , *4*, 99–109. [CrossRef]
22. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
23. Rizvi, S.; Kurtz, A.; Pfeffer, J.; Rizvi, M. Securing the Internet of Things (IoT): A Security Taxonomy for IoT. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 163–168. [CrossRef]

24. Herrero, R. Network and Transport Layers. In *Fundamentals of IoT Communication Technologies*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 77–109.
25. Brasser, F.; El Mahjoub, A.; Sadeghi, A.R.; Wachsmann, C. TyTAN: Tiny Trust Anchor for Tiny Devices. In Proceedings of the ACM/IEEE International Conference on Embedded Software, Turin, Italy, 30 September–5 October 2018; pp. 1–10. [CrossRef]
26. Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors* **2018**, *18*, 11734–11753. [CrossRef]
27. Malina, L.; Hajny, J.; Fujdiak, R.; Hosek, J. On perspective of security and privacy-preserving solutions in the Internet of Things. *Comput. Netw.* **2016**, *102*, 83–95. [CrossRef]
28. Künsch, H.R. The jackknife and the bootstrap for general stationary observations. *Ann. Stat.* **1989**, *17*, 1217–1241. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.