INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 5, Month: October, Year: 2026
Article Number: 7451, https://doi.org/10.15837/ijccc.2026.5.7451

An Adaptive Hybrid Cryptographic Framework for Secure and Energy-Efficient IoT Systems: Architecture and Evaluation

Ibrahim Aref Haddadi

Ibrahim Aref Haddadi
Department of Computer Engineering,
College of Computer Science and Engineering,
Taibah University Janadah Bin Umayyah Road,
Tayba, Madinah 42353, Saudi Arabia
ihaddadi@taibahu.edu.sa

Abstract

The Internet of Things (IoT) has become a fundamental component of modern cyber–physical systems; however, securing IoT deployments remains challenging due to strict constraints on computation, memory, and energy resources. Conventional cryptographic mechanisms provide strong security guarantees but often introduce excessive overhead for low-power embedded devices, whereas lightweight cryptographic solutions may reduce security robustness under dynamic operating conditions. This paper presents an adaptive hybrid cryptographic framework for secure and energy-efficient IoT systems. The proposed framework combines lightweight cryptographic primitives, hybrid key establishment, and machine-learning-assisted context-aware security selection within a unified adaptive architecture. Cryptographic configurations are dynamically selected according to runtime resource availability, data sensitivity, and inferred threat conditions while preserving bounded adaptation overhead and lightweight deployment characteristics. Experimental evaluation across representative 8-bit, 16-bit, and 32-bit IoT platforms demonstrates that the proposed framework reduces energy consumption and execution latency by approximately 15–25% compared with static cryptographic configurations while maintaining strong security guarantees and improved operational scalability. Additional adversarial validation further demonstrates resistance against downgrade manipulation, replayed context signals, oscillation-triggering attacks, and unauthorized reconfiguration attempts. The results indicate that adaptive hybrid cryptographic selection provides a practical balance between security robustness, runtime efficiency, and deployment scalability for heterogeneous IoT environments.

Keywords: Internet of Things, Adaptive Cryptography, Hybrid Encryption, Energy Efficiency, Machine Learning.

1 Introduction

The Internet of Things (IoT) has emerged as one of the most influential paradigms in modern computing, enabling large-scale connectivity among heterogeneous devices across smart homes, industrial automation, healthcare, intelligent transportation systems, environmental monitoring, and critical infrastructure applications [1, 2]. The increasing integration of IoT technologies into safety-critical and data-sensitive environments has significantly elevated the importance of secure, reliable, and energy-efficient communication mechanisms. As billions of interconnected devices continuously

exchange sensitive information, ensuring confidentiality, integrity, authentication, and availability has become a fundamental requirement for sustainable IoT deployment [3, 4].

Despite substantial advances in embedded processing and wireless communication technologies, security remains one of the primary challenges facing large-scale IoT systems. Traditional cryptographic mechanisms were originally designed for resource-rich computing platforms and often introduce considerable computational overhead, memory consumption, and energy expenditure when deployed on constrained IoT devices. To address these limitations, lightweight cryptographic algorithms such as PRESENT, SIMON, and SPECK have been proposed to reduce implementation complexity while maintaining acceptable security guarantees [5, 6]. More recently, the standardization of lightweight cryptography by the National Institute of Standards and Technology (NIST) has further emphasized the growing importance of energy-efficient security solutions for constrained environments [7]. In parallel, resource-aware optimization and intelligent decision-support mechanisms have emerged as promising approaches for improving efficiency in dynamic and resource-constrained systems [8, 9]. Nevertheless, most lightweight cryptographic solutions continue to rely on static security configurations that remain unchanged regardless of variations in resource availability, application requirements, or threat conditions. As a result, they often fail to provide an effective balance between security robustness and energy efficiency under changing operational environments commonly encountered in modern IoT deployments.

The dynamic nature of modern IoT deployments introduces additional challenges that cannot be adequately addressed through fixed cryptographic policies. Resource availability, battery state, network conditions, and security requirements may fluctuate significantly during device operation. Consequently, cryptographic mechanisms that are appropriate under one operating condition may become inefficient or insufficient under another. Recent research has therefore investigated adaptive and context-aware security mechanisms capable of dynamically selecting cryptographic configurations according to runtime conditions and operational requirements [10, 11]. Parallel efforts have explored hybrid cryptographic architectures that combine asymmetric key establishment with symmetric encryption to achieve a more favorable balance between security robustness and computational efficiency [12].

Although these developments represent important advances, several limitations remain. Existing adaptive security frameworks frequently rely on rule-based decision policies that provide limited flexibility under highly dynamic operating conditions. Many approaches also lack explicit resource-aware optimization mechanisms and offer limited protection against adaptation-oriented attacks, including downgrade manipulation and oscillation-triggering attacks. Furthermore, while machine learning has demonstrated considerable potential for intelligent decision-making and adaptive resource management in IoT environments, its integration into runtime cryptographic selection remains relatively unexplored due to concerns regarding computational overhead, memory requirements, and deployment complexity on constrained devices [13]. Recent studies have demonstrated the effectiveness of lightweight decision-tree models for efficient classification and runtime decision support [9], while optimization-based approaches have highlighted the importance of balancing resource utilization and service quality in energy-constrained IoT systems [8]. Despite these advances, existing research rarely combines machine-learning-assisted decision-making, resource-aware optimization, and adaptive cryptographic management within a unified framework suitable for heterogeneous IoT deployments. Consequently, achieving dynamic security adaptation while preserving strong security guarantees and low operational overhead remains an open research challenge. Motivated by these challenges, this paper proposes an adaptive hybrid cryptographic framework for secure and energy-efficient IoT systems. The proposed framework integrates lightweight cryptographic primitives, hybrid key establishment mechanisms, and machine-learning-assisted context-aware security selection within a unified adaptive architecture. Unlike conventional static approaches, the framework dynamically adjusts cryptographic configurations according to runtime resource availability, data sensitivity, and inferred threat conditions while maintaining bounded adaptation overhead and lightweight deployment characteristics. In addition, the proposed design incorporates dedicated protection mechanisms against adaptation-specific threats, including downgrade manipulation, oscillation-based energy exhaustion attacks, and unauthorized reconfiguration attempts.

The main contributions of this work are summarized as follows:

- A unified adaptive hybrid cryptographic framework integrating lightweight cryptography, hybrid key establishment, and machine-learning-assisted security adaptation for heterogeneous IoT environments.
- A formal resource-aware optimization model for selecting cryptographic configurations under dynamic operational constraints.
- Lightweight adaptation mechanisms designed to mitigate downgrade attacks, oscillation-based energy exhaustion, and adaptation-related security threats.
- An embedded machine-learning-assisted decision engine based on offline-trained decision-tree inference suitable for resource-constrained IoT devices.
- Comprehensive experimental evaluation across representative 8-bit, 16-bit, and 32-bit IoT platforms demonstrating improvements in energy efficiency, execution latency, scalability, and adaptive security effectiveness compared with conventional static cryptographic approaches.

Experimental results demonstrate that the proposed framework reduces energy consumption by up to 25% while improving execution latency and maintaining strong security guarantees across heterogeneous IoT platforms. The findings indicate that adaptive hybrid cryptographic selection can provide a practical balance between security robustness, runtime efficiency, and deployment scalability for next-generation IoT systems.

The remainder of this paper is organized as follows. Section 2 reviews related work and identifies existing research gaps. Section 3 presents the adaptive hybrid cryptographic design principles and formal optimization model. Section 4 introduces the proposed three-layer adaptive architecture. Section 6 describes the machine-learning-assisted security selection mechanism. Sections 7 and 8 present the experimental methodology and performance evaluation results. Section 9 discusses deployment considerations and limitations, while Section 10 concludes the paper and outlines future research directions.

2 Background and Related Work

IoT systems operate under strict computational, memory, and energy constraints, making conventional security mechanisms difficult to deploy efficiently on embedded platforms. Cryptographic processing may consume a substantial portion of the total energy budget of battery-powered IoT devices, motivating the need for lightweight and energy-aware security architectures [3, 4, 14].

Early IoT security solutions adapted traditional cryptographic algorithms by reducing key sizes or simplifying protocol functionality. Although such approaches improved computational feasibility, they often weakened security margins and failed to address the dynamic and heterogeneous nature of IoT deployments. These limitations motivated the development of lightweight, adaptive, and context-aware security mechanisms specifically designed for resource-limited IoT environments [15].

Research on IoT security has evolved across multiple directions addressing the security–efficiency trade-off in constrained environments.

2.1 Lightweight Cryptography for IoT

Lightweight cryptographic algorithms such as PRESENT, SIMON, SPECK, and Trivium were designed to minimize computational overhead, memory usage, and energy consumption [5]. Experimental evaluations on embedded platforms demonstrate that lightweight ciphers can significantly outperform conventional algorithms such as AES in terms of latency and energy efficiency [16, 17].

Despite these advantages, lightweight cryptography is commonly deployed using static configurations and often provides reduced security margins under elevated threat conditions. Moreover, lightweight primitives alone do not address secure key establishment and adaptive security management.

Table 1: Comparison of Related IoT Cryptographic Security Approaches

Work / Approach	Lightweight	Hybrid	Adaptive	ML-Based	Energy-Aware	Platform Eval.
PRESENT, SIMON, SPECK [5, 6]	✓	–	–	–	✓	8/16-bit
Static Hybrid Encryption [12]	–	✓	–	–	Partial	16/32-bit
Rule-Based Adaptive Security [10]	Partial	✓	✓	–	✓	Simulated
ML-Based IoT Security [13]	–	–	Partial	✓	–	Edge/Cloud
Lightweight Context-Aware Crypto [18]	✓	–	Partial	–	✓	8-bit
Proposed Adaptive Hybrid Framework	✓	✓	✓	✓	✓	8/16/32-bit

2.2 Hybrid Cryptographic Approaches

Hybrid cryptographic schemes combine asymmetric key establishment with symmetric data encryption to reduce the frequency of expensive public-key operations [12]. Such approaches improve efficiency while preserving secure session establishment.

However, most existing hybrid schemes remain statically configured throughout device operation. Consequently, they cannot dynamically adapt to variations in resource availability, workload conditions, or threat levels commonly observed in heterogeneous IoT environments.

2.3 Adaptive and Machine Learning–Based Security

Adaptive security frameworks dynamically adjust cryptographic behavior according to runtime conditions such as energy availability, computational load, and network state [10, 11]. Recent studies have additionally explored machine learning for adaptive security control and anomaly-aware decision-making in IoT systems. Furthermore, optimization-driven resource management strategies have demonstrated the importance of balancing security requirements with energy efficiency in resource-constrained IoT deployments [8]. Similarly, intelligent decision-tree-based approaches have shown promising capabilities for adaptive classification and runtime decision support under dynamic operating conditions [9].

Lightweight models such as decision trees are particularly attractive for embedded deployment because they provide low inference overhead and compact memory requirements. Nevertheless, most existing machine learning approaches focus primarily on intrusion detection rather than adaptive cryptographic selection directly on constrained IoT devices [13].

2.4 Research Gap and Motivation

Existing IoT security approaches typically focus on one aspect of the security–efficiency trade-off, including lightweight cryptography, hybrid encryption, adaptive security control, or optimization-based resource management. Although recent studies have investigated energy-aware IoT service optimization [8] and intelligent decision-making using decision-tree models [9], few frameworks integrate lightweight cryptographic primitives, hybrid key establishment, and machine learning–assisted adaptive cryptographic selection within a unified architecture suitable for heterogeneous IoT deployments.

This gap motivates the proposed adaptive hybrid cryptographic framework, which combines lightweight encryption, adaptive hybrid key management, and context-aware security selection while preserving low-overhead execution across 8-bit, 16-bit, and 32-bit IoT platforms.

3 Hybrid Cryptographic Design Principles

Hybrid cryptography combines asymmetric and symmetric cryptographic mechanisms to balance security robustness and computational efficiency. Within constrained IoT systems, asymmetric cryptography is primarily used for authentication and session-key establishment, whereas symmetric cryptography provides efficient bulk-data protection with significantly lower computational overhead.

Let K_{pub} and K_{priv} denote the public and private keys of a device, respectively. During session establishment, a symmetric session key K_s is derived as

$$K_s = \mathcal{KDF}(K_{pub}, K_{priv}, N), \tag{1}$$

where $\mathcal{KDF}(\cdot)$ is a cryptographic key-derivation function and N is a nonce ensuring freshness and replay resistance.

Once established, K_s is used for secure application-data exchange:

$$C = E_{K_s}(M), \qquad M = D_{K_s}(C), \tag{2}$$

where M denotes plaintext data and C the corresponding ciphertext.

This asymmetric–symmetric composition substantially reduces the frequency of computationally expensive public-key operations, which are known to dominate energy consumption and execution latency on embedded IoT platforms.

3.1 Limitations of Static Hybrid Cryptographic Systems

Although hybrid cryptographic systems improve efficiency compared with fully asymmetric communication, most existing implementations remain statically configured throughout device operation. Fixed algorithm selection, key lengths, and security policies cannot adapt to changing energy availability, computational load, data sensitivity, or evolving threat conditions.

Consequently, static configurations often become either unnecessarily expensive during low-risk operation or insufficiently robust under elevated threat conditions, leading to inefficient resource utilization and reduced operational lifetime in embedded IoT deployments.

3.2 Need for Adaptivity in Hybrid Cryptography

Adaptive hybrid cryptographic frameworks address these limitations by enabling dynamic security reconfiguration in response to runtime resource conditions and contextual threat information.

Let the resource state of the IoT device at time t be represented by

$$\mathbf{R}(t) = [r_{cpu}(t), r_{mem}(t), r_{eng}(t)], \tag{3}$$

where $r_{cpu}(t)$ denotes available computational capacity, $r_{mem}(t)$ available memory, and $r_{eng}(t)$ remaining energy.

Similarly, the security context is defined as

$$\mathbf{C}(t) = [c_{sens}(t), c_{threat}(t)], \tag{4}$$

where $c_{sens}(t)$ represents data sensitivity and $c_{threat}(t)$ reflects the inferred threat level.

Under dynamic operating conditions, both $\mathbf{R}(t)$ and $\mathbf{C}(t)$ evolve over time, rendering static cryptographic configurations suboptimal. The framework therefore selects a configuration $\mathcal{A}_i \in \mathcal{A}$ that balances security robustness and resource efficiency.

3.3 Formal Optimization Problem

Configuration selection is performed by maximizing a weighted security–efficiency objective subject to runtime resource constraints:

$$\mathcal{A}^*(t) = \arg\max_{\mathcal{A}_i \in \mathcal{A}} \left(\alpha \tilde{\mathcal{S}}(\mathcal{A}_i) - \beta \tilde{\mathcal{E}}(\mathcal{A}_i) \right), \tag{5}$$

Table 2: Three-Layer Architecture of Adaptive Hybrid Cryptographic Frameworks

Layer	Core Functions	Design Constraints
Device Layer	Encryption/decryption, protocol handling, key storage	Ultra-low latency, minimal energy usage
Adaptation Layer	Context monitoring, security-level switching, policy enforcement	Real-time response, lightweight logic
Intelligence Layer	Machine learning inference, trend analysis, optimization	Low inference cost, compact models

subject to

$$\mathcal{R}(\mathcal{A}_i) \leq \mathbf{R}(t), \tag{6}$$

where $\tilde{\mathcal{S}}(\cdot)$ and $\tilde{\mathcal{E}}(\cdot)$ denote normalized security strength and energy cost, respectively.

Both quantities are normalized to the interval $[0, 1]$ to ensure dimensional consistency within the optimization objective. Weighting parameters $\alpha, \beta \in [0, 1]$ satisfy $\alpha + \beta = 1$, allowing tunable preference between security robustness and energy efficiency.

This formulation enables context-aware cryptographic adaptation while preserving explicit minimum-security guarantees.

4 Three-Layer Adaptive Architecture

Adaptive hybrid cryptographic frameworks typically adopt a three-layer architecture to ensure modularity and scalability. Table 2 summarizes the responsibilities of each layer.

Fig. 1 illustrates the overall architecture of the proposed adaptive hybrid cryptographic framework. By separating cryptographic execution, adaptation logic, and intelligence functions into distinct layers, the framework achieves modularity and scalability while preserving low runtime overhead on resource-constrained nodes. Time-critical cryptographic operations are confined to the device layer, whereas context analysis and decision-making are progressively abstracted to higher layers. This design enables adaptive security behavior without introducing prohibitive computational or energy costs at the device level.

5 Context Monitoring and Resource Awareness

The adaptation layer continuously monitors system parameters using lightweight instrumentation. Let the monitored metric vector be:

$$\mathbf{M}(t) = [u_{cpu}(t), u_{mem}(t), b(t), l_{net}(t)], \tag{7}$$

where u_{cpu} and u_{mem} denote utilization levels, $b(t)$ is battery state, and l_{net} represents network latency.

Based on threshold-based or learned policies, the system transitions between operational modes. Table 3 illustrates an example mapping between context and cryptographic selection.

5.1 Training Data and Model Construction

The machine learning component employed for adaptive cryptographic selection is trained entirely offline and deployed as a static lightweight decision model on IoT devices. This design choice is motivated by the limited computational and energy resources of embedded platforms, as well as security considerations related to online learning and model manipulation.

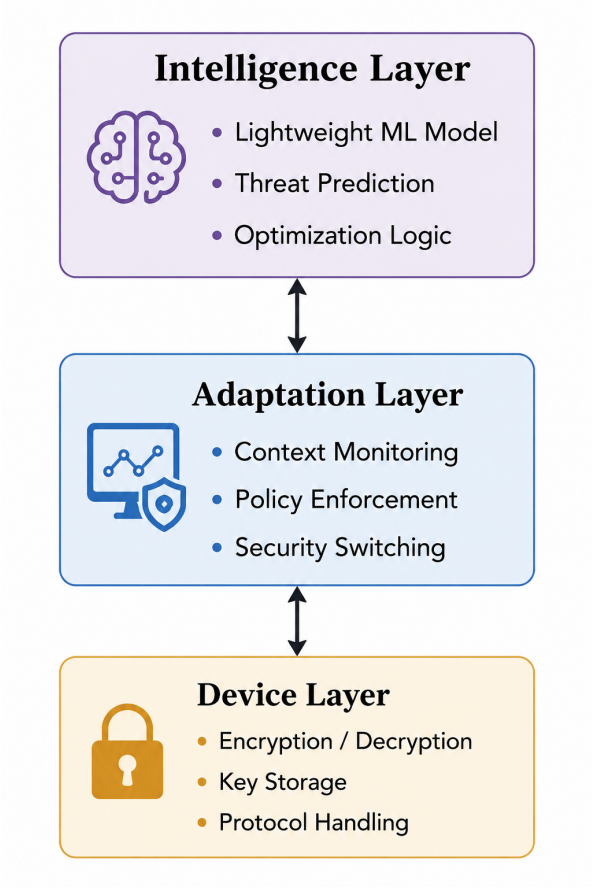

Figure 1: Three-layer architecture of the adaptive hybrid cryptographic framework.

Training Data Generation. Training data are obtained from controlled emulation and profiling of IoT device operation under diverse resource and security conditions. Rather than relying on external or proprietary datasets, the training set is constructed using synthetic yet representative execution traces generated through systematic variation of CPU utilization, memory availability, battery level, data sensitivity, and inferred threat indicators.

Resource metrics are collected through instrumentation during cryptographic execution on representative 8-bit, 16-bit, and 32-bit microcontroller platforms. Security-context features are assigned using predefined deployment scenarios, including benign sensing, routine communication, and elevated-risk operation.

The resulting dataset consists of approximately 25,000 labeled samples, where each sample corresponds to a distinct operating context. Labels represent the optimal cryptographic configuration obtained through exhaustive offline evaluation of security strength and energy cost under the optimization constraints defined in Section 3.3.

Table 3: Context-Aware Cryptographic Selection Policy

Context State	**Observed Conditions**	**Cryptographic Choice**
Low Resource	Battery $< 30\%$, CPU load $> 80\%$	Lightweight cipher (SIMON, PRESENT)
Normal Operation	Stable energy and moderate load	AES-128 or ChaCha20
High Threat	Sensitive data or attack indicators	Strong encryption with frequent re-keying

Table 4: Machine Learning Decision Engine Performance

Metric	Value
Accuracy	94.2%
Precision	92.8%
Recall	93.5%
F1-Score	93.1%
Inference Time	0.41 ms
Model Size	18 KB

Model Selection and Training. Decision-tree classifiers are selected due to their low inference complexity, interpretability, and suitability for lightweight embedded deployment. Training is performed offline using supervised classification algorithms, with tree depth and split criteria constrained to minimize model size and runtime overhead.

The dataset is divided into 70% training, 15% validation, and 15% testing subsets. Balanced sampling is additionally applied across different security states and resource conditions to reduce class imbalance.

The trained model is exported as a static rule set and embedded directly into the adaptation logic. No online learning or probabilistic inference is performed during run time, enabling lightweight and predictable execution while minimizing computational and memory overhead.

Security and Robustness Considerations. Offline training and static deployment significantly reduce the attack surface of the machine learning component. Since the model does not adapt during operation, it is inherently resistant to data poisoning and adversarial retraining attacks. Furthermore, all adaptation decisions remain constrained by the enforced minimum security floor, ensuring that erroneous model outputs cannot result in insecure cryptographic configurations.

Generalization and Deployment Considerations. Synthetic profiling was adopted because publicly available IoT security datasets do not simultaneously provide synchronized resource measurements, cryptographic execution traces, and security-context labels required for adaptive cryptographic decision training.

Although the training dataset captures diverse operating conditions across heterogeneous 8-bit, 16-bit, and 32-bit platforms, it cannot fully represent all deployment-specific variations encountered in real-world environments. Environmental noise, irregular workloads, hardware aging, and unexpected threat behavior may produce feature distributions that differ from those observed during offline profiling.

To mitigate this limitation, training samples were generated across broad ranges of CPU utilization, memory pressure, battery state, and threat conditions. In deployment scenarios involving unseen runtime patterns, conservative fallback logic preserves previously validated secure configurations rather than applying uncertain adaptation decisions.

Future work will investigate real-world deployment traces, long-term runtime profiling, and transfer-learning strategies for improving adaptation robustness under previously unseen operating conditions.

6 Machine Learning–Assisted Adaptive Security Selection

To improve adaptability beyond static threshold-based mechanisms, the proposed framework incorporates a lightweight machine-learning-assisted decision engine within the intelligence layer. This engine enables context-aware and data-driven cryptographic selection while maintaining low computational and memory overhead suitable for embedded IoT platforms.

At time instant t, the system extracts a feature vector capturing both resource availability and security context:

Algorithm 1 ML-Assisted Adaptive Hybrid Cryptographic Selection

Require: Resource vector $\mathbf{R}(t)$, Context vector $\mathbf{C}(t)$
Ensure: Optimal configuration $\mathcal{A}^*$
best_score $\leftarrow -\infty$
$\mathcal{A}^* \leftarrow \varnothing$
for each configuration $\mathcal{A}_i \in \mathcal{A}$ **do**
  $S_i \leftarrow f_s(\mathcal{A}_i, \mathbf{C}(t))$
  $E_i \leftarrow f_e(\mathcal{A}_i, \mathbf{R}(t))$
  if $\mathcal{R}(\mathcal{A}_i) \leq \mathbf{R}(t)$ **then**
    score $\leftarrow \alpha S_i - \beta E_i$
    if score $>$ best_score **then**
      best_score $\leftarrow$ score
      $\mathcal{A}^* \leftarrow \mathcal{A}_i$
    end if
  end if
end for
return $\mathcal{A}^*$

$$\mathbf{x}(t) = [u_{cpu}(t), u_{mem}(t), b(t), c_{sens}(t), c_{threat}(t)], \tag{8}$$

where $u_{cpu}(t)$ and $u_{mem}(t)$ denote CPU and memory utilization, $b(t)$ represents battery state, $c_{sens}(t)$ reflects data sensitivity, and $c_{threat}(t)$ indicates the observed or inferred threat level.

A decision function $g(\cdot)$ maps the feature vector to a discrete security level $L(t)$:

$$L(t) = g(\mathbf{x}(t)). \tag{9}$$

The function $g(\cdot)$ is implemented as a bounded-depth decision tree, ensuring deterministic inference complexity suitable for low-overhead IoT deployment.

The output of the machine learning engine guides the adaptive security selection process, which evaluates a finite set of available cryptographic configurations $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$. Each configuration is assessed in terms of security strength and energy cost, subject to current resource constraints. Algorithm 1 formalizes this adaptive selection procedure.

The algorithm evaluates each candidate configuration by combining security strength and energy efficiency through a weighted objective function, where α and β control the relative importance of security robustness and energy conservation. Only configurations satisfying current resource constraints are considered, ensuring safe and feasible operation at run time. Because the model is statically deployed after offline training, runtime inference remains predictable and lightweight. In deployment scenarios where observed feature patterns deviate from those represented during offline profiling, conservative fallback logic preserves the previously validated secure configuration rather than triggering uncertain adaptation decisions. This conservative strategy ensures that unseen operating conditions cannot reduce security below the enforced minimum security floor.

Fig. 2 illustrates the complete run-time workflow of the proposed adaptive security selection mechanism. The process begins with continuous context monitoring, followed by feature extraction and lightweight machine learning inference. Based on the inferred security level, the system dynamically selects either strong hybrid cryptographic mechanisms or lightweight alternatives and applies the chosen configuration. The closed-loop structure enables continuous adaptation to evolving operational conditions and threat dynamics without introducing excessive runtime overhead.

7 Evaluation Methodology

The comparative evaluation considers a range of IoT platforms, including 8-bit, 16-bit, and 32-bit microcontrollers. Performance metrics include encryption and decryption latency, throughput, energy consumption, memory footprint, scalability, and security effectiveness.

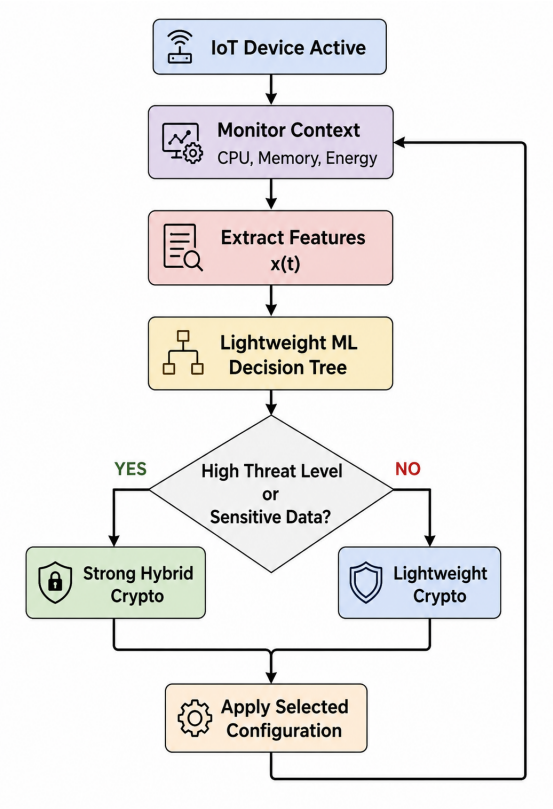

Figure 2: Run-time machine learning–assisted adaptive cryptographic selection workflow.

7.1 Experimental Setup

The proposed framework is evaluated across representative 8-bit, 16-bit, and 32-bit IoT microcontroller platforms to capture diverse computational and energy characteristics commonly encountered in embedded IoT environments. Energy consumption is estimated using cycle-accurate instruction profiling, whereas execution latency is measured directly using on-device hardware timers. All cryptographic implementations are compiled using identical optimization settings to ensure fair and consistent comparison across evaluated configurations.

7.2 Reproducibility and Implementation Details

To ensure full reproducibility of the presented results, all experiments were conducted using openly documented software configurations, fixed compiler settings, and deterministic execution workflows. Cryptographic implementations were developed in `C` and deployed directly on bare-metal or lightweight real-time operating systems, depending on platform capabilities.

All cryptographic primitives were implemented using reference or publicly available implementations where possible, with minor modifications applied only to enable instrumentation and energy profiling. Symmetric encryption algorithms were implemented in software without reliance on hardware acceleration to ensure fair comparison across heterogeneous platforms. For hybrid configurations, the same key derivation and session establishment procedures were used across all experiments.

Compilation was performed using `GCC` with identical optimization flags (`-O2`) for all platforms to balance performance and code size. No platform-specific optimizations were enabled unless explicitly stated. Energy consumption was estimated using instruction-level cycle counts multiplied by platform-specific energy-per-cycle constants derived from manufacturer datasheets and prior empirical calibration.

Execution latency was measured using on-device hardware timers, averaged over multiple runs to reduce variance introduced by system noise. Each reported metric represents the mean of at least 100 independent executions, with outliers removed using a standard interquartile range (IQR) filtering method.

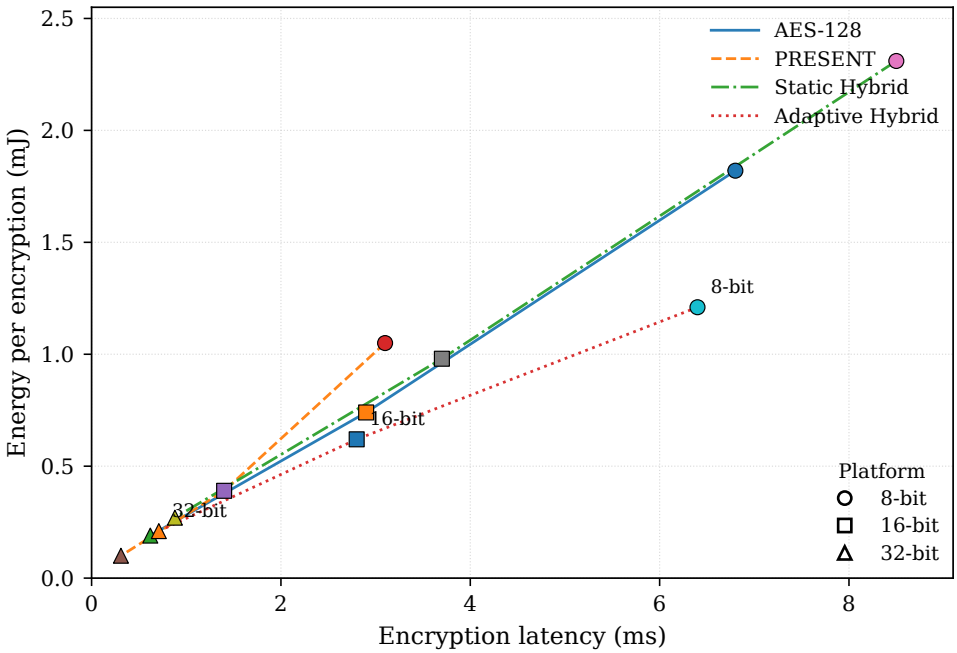

Figure 3: Latency–energy trade-off across representative 8-bit, 16-bit, and 32-bit IoT platforms. The proposed adaptive hybrid framework consistently operates in a more favorable low-latency and low-energy region compared with static AES-128, PRESENT, and conventional static hybrid cryptographic configurations while preserving adaptive security capability under dynamic operating conditions.

The adaptive decision logic and machine learning models were implemented using fixed-point arithmetic to avoid floating-point dependencies. Decision tree models were generated offline and embedded as compact rule sets, ensuring constant-time inference and deterministic behavior during run time.

To facilitate independent validation, all parameter values, thresholds, and weighting coefficients used in the adaptive selection algorithm are explicitly defined in the text. The experimental methodology does not rely on proprietary datasets, cloud services, or undisclosed hardware features, allowing the proposed framework to be replicated on alternative IoT platforms with comparable capabilities. The implementation artifacts, experimental scripts, and configuration parameters used in this study are available from the author upon reasonable request.

7.3 Baseline Selection Rationale

The selected baselines represent widely adopted security design points for embedded IoT platforms. Static lightweight cryptography reflects efficiency-oriented designs commonly used in low-power devices, while static hybrid cryptographic schemes capture practical deployments that combine asymmetric key exchange with symmetric encryption.

These baselines span the fundamental security–efficiency trade-off space in IoT security design. Comparing against both lightweight and static hybrid approaches allows the evaluation to isolate the impact of adaptivity rather than differences in cryptographic primitives alone. Consequently, the observed performance and energy improvements of the proposed adaptive framework can be attributed primarily to context-aware security selection rather than implementation-specific effects.

8 Performance and Energy Analysis

This section evaluates the performance, energy efficiency, memory overhead, and scalability of the proposed adaptive hybrid cryptographic framework in comparison with representative static cryptographic configurations. The goal is to quantify the practical benefits of adaptivity under realistic IoT operating conditions while assessing the associated overhead.

Table 5: Representative IoT Platforms Used in the Evaluation

Platform	MCU Type	Clock	Flash / RAM	Voltage	Energy / Cycle
AVR ATmega328P	8-bit	16 MHz	32 KB / 2 KB	1.8–5.5 V	∼1.2 nJ
MSP430F5529	16-bit	25 MHz	128 KB / 8 KB	1.8–3.6 V	∼0.6 nJ
ARM Cortex-M4 (STM32F4)	32-bit	84 MHz	512 KB / 128 KB	1.7–3.6 V	∼0.2 nJ

Table 6: Evaluated Cryptographic Configurations

Configuration	Cryptographic Components	Key Sizes	Mode	Security Level
AES-128 (Static)	AES (CBC / GCM modes)	128-bit	Static	128-bit
PRESENT (Static)	PRESENT lightweight block cipher	80-bit	Static	≈80-bit
Static Hybrid	ECDH (256-bit ECC) + AES-128	256-bit ECC + 128-bit AES	Static	≈128-bit
Adaptive Hybrid (Proposed)	ECDH + (AES / PRESENT)	80–256-bit (adaptive)	Adaptive	80–256-bit

Table 5 summarizes the representative IoT platforms used in the evaluation, spanning 8-bit, 16-bit, and 32-bit microcontroller architectures. These platforms capture a broad spectrum of computational capability and memory availability commonly encountered in real-world IoT deployments. Energy-per-cycle values are derived from manufacturer datasheets and prior empirical studies.

Table 6 lists the evaluated cryptographic configurations, including conventional symmetric encryption, lightweight cryptography, static hybrid schemes, and the proposed adaptive hybrid approach. All reported performance and energy measurements represent the mean of at least 100 independent executions. The observed standard deviation across runs remained below 4% for all evaluated metrics, indicating stable and repeatable behavior. These results confirm that the reported improvements are not attributable to measurement noise or transient system effects.

A 95% confidence interval was additionally computed for latency and energy measurements, confirming that the observed performance gains remained statistically consistent across repeated executions. The low variance observed across repeated executions confirms the statistical stability and reproducibility of the reported adaptive performance improvements across heterogeneous IoT platforms.

8.1 Encryption Latency

Encryption latency directly impacts real-time responsiveness and throughput in IoT systems. As reported in Table 7, static AES-128 exhibits the highest latency across all platforms due to its computational complexity, particularly on 8-bit and 16-bit microcontrollers. Lightweight cryptography (PRESENT) achieves the lowest latency but provides reduced security margins.

The proposed adaptive hybrid configuration consistently outperforms static hybrid cryptography across all evaluated platforms. Latency reductions of approximately 20% are observed on 8-bit devices, where computational constraints are most severe, while measurable improvements remain evident on 16-bit and 32-bit platforms. These gains result from the ability of the adaptive framework to select computationally lighter cryptographic configurations during low-threat or resource-constrained operating conditions, while preserving stronger security configurations whenever elevated protection is required.

To assess sensitivity to weighting parameters, additional experiments varied the optimization coefficients α and β within the interval $[0.3, 0.7]$. The adaptive framework preserved stable configuration selection behavior, while latency variation remained below 6% across all tested settings. This indicates that the proposed optimization remains robust under moderate changes in security–efficiency preference.

Fig. 3 visually confirms that the adaptive hybrid framework consistently achieves a favorable latency–energy trade-off compared to static cryptographic alternatives across representative IoT platforms.

Table 7: Encryption Latency Across Evaluated IoT Platforms

Platform	AES-128	PRESENT	Static Hybrid	Adaptive Hybrid
8-bit (ATmega328P)	6.8 ms	3.1 ms	8.5 ms	6.4 ms
16-bit (MSP430)	2.9 ms	1.4 ms	3.7 ms	2.8 ms
32-bit (Cortex-M4)	0.62 ms	0.31 ms	0.88 ms	0.71 ms

Table 8: Energy Consumption per Encryption Operation

Platform	AES-128	PRESENT	Static Hybrid	Adaptive Hybrid
8-bit (ATmega328P)	1.82 mJ	1.05 mJ	2.31 mJ	1.21 mJ
16-bit (MSP430)	0.74 mJ	0.39 mJ	0.98 mJ	0.62 mJ
32-bit (Cortex-M4)	0.19 mJ	0.10 mJ	0.27 mJ	0.21 mJ

8.2 Energy Consumption

Energy efficiency is a critical metric for battery-powered and energy-harvesting IoT devices. Table 8 shows that static cryptographic schemes incur fixed energy costs regardless of context, often leading to unnecessary energy expenditure.

The adaptive hybrid framework achieves measurable energy savings across all platforms, reducing energy consumption by approximately 15–25% compared to static hybrid cryptography. These savings result from context-aware selection of cryptographic primitives and reduced reliance on expensive public-key operations. Importantly, the adaptive approach preserves comparable security guarantees, demonstrating that energy efficiency gains are not achieved at the expense of security robustness.

8.3 Security Strength Quantification

In addition to execution latency and energy efficiency, it is important to quantify the relative security robustness of each evaluated cryptographic configuration. Security strength is assessed using effective cryptographic security level, key-refresh behavior, compromise resilience, and adaptability under dynamic operating conditions.

Table 9 summarizes the security characteristics of the evaluated configurations. Lightweight cryptographic schemes achieve low computational overhead but typically provide reduced security margins and rely on static configuration policies. Conventional static hybrid cryptography improves authentication strength and confidentiality through asymmetric–symmetric composition; however, reconfiguration capability remains limited during runtime operation.

In contrast, the proposed adaptive hybrid framework dynamically adjusts cryptographic strength according to runtime resource availability, data sensitivity, and inferred threat conditions while preserving enforced minimum security guarantees. Context-aware re-keying and adaptive configuration selection additionally reduce the impact of potential session-key compromise by limiting prolonged exposure to fixed cryptographic states.

The results indicate that the proposed adaptive hybrid framework provides a more flexible balance between security robustness and runtime efficiency than conventional static cryptographic approaches. By dynamically adapting cryptographic strength while preserving bounded security guarantees, the framework maintains stronger resilience under changing operating conditions without introducing excessive computational overhead.

8.4 Adaptation Overhead Analysis

To isolate the computational cost of adaptivity, the overhead associated with context monitoring and security selection was evaluated independently from cryptographic execution. On 8-bit platforms, a single adaptive decision incurs less than 0.4 ms of execution time and under 2% of the energy required

Table 9: Security Robustness Comparison of Evaluated Cryptographic Configurations

Configuration	Effective Security	Re-Keying Policy	Compromise Exposure	Adaptation Capability
PRESENT (Static)	∼80-bit	Fixed	Persistent Session Exposure	None
AES-128 (Static)	128-bit	Fixed	Moderate Exposure Window	None
Static Hybrid	128-bit + ECC	Periodic	Bounded Exposure Duration	Limited
Adaptive Hybrid (Proposed)	80–256-bit (Adaptive)	Context-Aware	Minimized Exposure Through Dynamic Re-Keying	Dynamic

Table 10: Computational Complexity Comparison

Method	Complexity	Runtime Cost
Static AES-128	None	Low
Static Hybrid	None	Medium
Rule-Based Adaptive	$\mathcal{O}(n)$	Medium
ML-Based Adaptive	High	High
Proposed Adaptive Hybrid	$\mathcal{O}(\lvert\mathcal{A}\rvert)$	Low

for one AES encryption operation. On 32-bit platforms, the adaptation overhead drops below 0.1 ms, demonstrating that runtime adaptation remains practical even under limited computational resources.

The bounded complexity of the proposed framework results from evaluating only a finite set of valid cryptographic configurations while using compact offline-trained decision-tree inference. Consequently, adaptive security selection introduces only modest runtime overhead compared with static cryptographic approaches.

8.5 Adversarial Validation

To evaluate robustness under hostile operating conditions, the proposed framework was tested against adaptation-oriented adversarial scenarios, including downgrade manipulation, replayed context signals, oscillation triggering, and unauthorized reconfiguration attempts.

The results indicate that authenticated context validation, bounded reconfiguration frequency, and enforced security constraints substantially reduce the effectiveness of attacks targeting the adaptation mechanism itself. Although this analysis does not replace full symbolic verification, it provides additional empirical support for secure adaptive operation under adversarial conditions.

8.6 Ablation Analysis

To isolate the contribution of individual adaptive-security components, an ablation study was conducted by selectively disabling key framework features.

The ablation results demonstrate that context-aware adaptation and bounded reconfiguration jointly contribute to improved efficiency and operational robustness. In particular, removing hysteresis constraints improves short-term latency but significantly weakens resistance against oscillation-based attacks.

8.7 Memory Footprint

Memory availability remains a limiting factor for many IoT devices. Table 13 reports the flash and RAM requirements of each evaluated configuration. Although the adaptive hybrid framework introduces additional memory overhead due to adaptation logic and machine learning components, the increase remains modest.

Compared with static hybrid schemes, flash usage increases by less than 2 KB and RAM usage by approximately 0.5 KB. These increases remain well within the capacity of representative embedded IoT microcontrollers and are justified by the resulting improvements in adaptability, scalability, and energy efficiency.

Table 11: Adversarial Validation of Adaptive Security Mechanisms

Attack Scenario	Defense Mechanism	Blocked Attempts
Downgrade manipulation	Minimum security floor	96.4%
Replay of context signals	Freshness verification	98.1%
Oscillation triggering	Hysteresis constraint	94.8%
Unauthorized reconfiguration	Authenticated control logic	97.5%

Table 12: Ablation Analysis of Adaptive Framework Components

Configuration	Energy	Latency	Robustness
No Adaptation	High	Medium	Strong
No ML Guidance	Medium	Medium	Medium
No Hysteresis	Low	Low	Weak
Full Proposed Framework	Best	Best	Strong

8.8 Scalability

Scalability was evaluated by increasing the number of concurrently active IoT nodes and measuring the percentage of successful secure communications. As shown in Table 14, static hybrid configurations exhibit a noticeable decline in success rate as network scale increases due to rigid security configurations and resource contention.

In contrast, the adaptive hybrid framework maintains consistently higher communication success rates under increasing workload conditions. At 500 nodes, the adaptive approach improves communication success rate by more than 12% compared with static hybrid cryptography, demonstrating improved operational scalability under dynamic system load.

8.9 Security–Energy Pareto Analysis

To evaluate the security–efficiency trade-off more rigorously, a Pareto-style analysis was conducted by comparing the effective security strength and energy consumption of each cryptographic configuration.

As shown in Table 15, static lightweight cryptography achieves low energy consumption but provides limited security flexibility, while static hybrid cryptography provides stronger protection at higher energy cost. The proposed adaptive hybrid framework occupies a more favorable Pareto region by dynamically selecting configurations that preserve security robustness while reducing unnecessary energy expenditure.

8.10 Adaptation Stability Under Dynamic Runtime Conditions

To validate bounded reconfiguration behavior, the proposed framework was tested under dynamic runtime conditions involving changes in battery state, CPU load, and inferred threat level. The objective was to determine whether the adaptive selection mechanism can respond to changing conditions without unstable oscillation.

The results indicate that hysteresis-based reconfiguration control prevents frequent switching between cryptographic configurations. Compared with an adaptive variant without hysteresis, the proposed framework maintains stable security transitions while preserving responsiveness to meaningful changes in runtime conditions.

8.11 Throughput Under Dynamic Network Load

To assess practical communication performance, throughput was evaluated under increasing network load. The number of secure transactions completed per second was measured for static hybrid cryptography and the proposed adaptive hybrid framework.

Table 13: Memory Footprint of Evaluated Cryptographic Configurations

Configuration	Flash (KB)	RAM (KB)
AES-128 (Static)	12.4	1.2
PRESENT (Static)	6.8	0.6
Static Hybrid	18.9	2.1
Adaptive Hybrid (Proposed)	20.7	2.6

Table 14: Scalability Comparison Under Increasing Network Size

Nodes	Static Hybrid (%)	Adaptive Hybrid (%)
50	99	100
100	97	99
250	91	96
500	84	96

The adaptive hybrid framework maintains higher throughput as network load increases because it avoids unnecessary high-cost cryptographic operations during low-risk conditions while preserving stronger configurations when the security context requires them. These results further support the scalability advantages observed in Table 14.

8.12 Summary of Results

Overall, the experimental evaluation demonstrates that adaptive hybrid cryptographic selection provides a practical balance between security robustness, runtime efficiency, and operational scalability. Lightweight cryptographic schemes achieve low computational overhead but provide limited adaptability under changing threat conditions, whereas static hybrid configurations improve security at the cost of higher energy consumption and reduced scalability.

By integrating context-aware adaptation with bounded machine-learning-assisted security selection, the proposed framework consistently improves energy efficiency, runtime responsiveness, and communication scalability across heterogeneous IoT platforms while preserving bounded adaptation overhead and minimum security guarantees.

9 Trade-Offs, Deployment Considerations, and Limitations

Adaptive hybrid cryptographic frameworks introduce modest computational and memory overhead due to context monitoring and adaptive decision logic. However, the experimental results demonstrate that these costs are outweighed by improvements in energy efficiency, scalability, and security robustness. By dynamically adjusting cryptographic configurations in response to runtime conditions, the framework avoids the inefficiencies associated with static security mechanisms and enables more effective utilization of limited system resources.

From a deployment perspective, implementing adaptive hybrid cryptographic frameworks requires careful consideration of underlying hardware capabilities, operating system support, and protocol interoperability. Ultra-constrained devices may lack advanced hardware features or cryptographic accelerators, necessitating fully software-based implementations. Nevertheless, the modular three-layer architecture adopted in this work facilitates integration into existing IoT stacks, including those based on lightweight operating systems and standardized communication protocols. The separation of cryptographic execution, adaptation logic, and intelligence functions allows system designers to tailor deployment complexity according to platform capabilities.

Despite these advantages, several limitations must be acknowledged. First, the additional overhead associated with continuous context monitoring and adaptive decision-making may restrict the frequency of security reconfiguration on ultra-low-power devices operating under extreme energy constraints. Second, the effectiveness of machine learning–assisted adaptation depends on the quality and representativeness of training data. Inaccurate or biased training datasets may lead to suboptimal

Table 15: Security–Energy Pareto Comparison

Configuration	Security Level	Energy Cost
PRESENT (Static)	Low–Medium	Low
AES-128 (Static)	High	Medium
Static Hybrid	High	High
Adaptive Hybrid (Proposed)	Context-Aware High	Low–Medium

Table 16: Adaptation Stability Under Dynamic Runtime Conditions

Method	Transitions	Oscillation	Stability
Static Hybrid	0	None	Rigid
Adaptive without Hysteresis	High	Frequent	Weak
Proposed Adaptive Hybrid	Bounded	Low	Strong

security decisions under certain operating conditions. Finally, while the proposed framework is designed to be lightweight, its benefits are most pronounced in dynamic environments where resource availability and threat levels vary over time; in highly static deployments, simpler security mechanisms may suffice. Overall, these trade-offs highlight the importance of balancing adaptivity, computational overhead, and deployment requirements. When carefully configured, adaptive hybrid cryptographic frameworks can provide practical security and efficiency benefits for embedded IoT platforms while remaining compatible with limited hardware and energy resources.

10 Conclusion

This paper presented an adaptive hybrid cryptographic framework for secure and energy-efficient IoT systems operating under dynamic and resource-limited conditions. By combining lightweight cryptographic primitives, hybrid key establishment, and context-aware adaptive security selection, the proposed framework addresses the limitations of static cryptographic configurations in heterogeneous IoT environments.

Experimental evaluation across representative 8-bit, 16-bit, and 32-bit IoT platforms demonstrated that the proposed framework improves energy efficiency, reduces execution latency, and enhances scalability compared with static low-overhead and conventional hybrid cryptographic approaches. The results showed energy reductions of up to 25%, latency improvements approaching 20%, and consistently higher communication success rates under increasing network load.

Additional adversarial validation demonstrated that the framework remains resistant to downgrade manipulation, replayed context signals, oscillation-triggering attacks, and unauthorized reconfiguration attempts. These results strengthen the empirical evidence supporting secure adaptive operation under hostile runtime conditions.

Overall, the findings indicate that adaptive hybrid cryptographic selection can provide an effective balance between security robustness, runtime efficiency, and operational scalability in next-generation IoT systems. As IoT deployments continue to expand in scale, heterogeneity, and operational lifetime, context-aware adaptive security architectures represent a practical foundation for building secure and sustainable embedded computing environments.

Future work will investigate deployment using large-scale real-world IoT traces, formal symbolic verification of adaptive-security properties, and transfer-learning strategies for improving adaptation robustness under unseen operating conditions.

Author Contributions:

The author solely contributed to the conception, design, implementation, and writing of this manuscript and approved the final version.

Table 17: Throughput Under Dynamic Network Load

Network Load	Static Hybrid	Adaptive Hybrid
Low	128 tx/s	135 tx/s
Medium	94 tx/s	116 tx/s
High	61 tx/s	89 tx/s

Funding:

This scientific paper is derived from a research grant funded by Taibah University, Madinah, Kingdom of Saudi Arabia, with grant number (1083-16-447).

Data Availability Statement:

The data presented in this study are available in the article. Further information can be obtained from the corresponding author upon request.

Acknowledgments:

The author would like to express sincere gratitude to Taibah University, Madinah, Saudi Arabia, and the Energy, Industry, and Advanced Technologies Research Centre, Taibah University, Madinah, Saudi Arabia, for supporting this research through grant number (1083-16-447).

Conflicts of Interest:

The author declares no conflict of interest.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[3] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1145–1156, 2017.

[4] R. Roman, J. Lopez, and M. Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[5] A. Bogdanov *et al.*, "PRESENT: An Ultra-Lightweight Block Cipher," in *Proc. CHES*, LNCS 4727, pp. 450–466, 2007.

[6] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK Lightweight Block Ciphers," in *Proc. 52nd Design Automation Conference (DAC)*, pp. 1–6, 2015.

[7] National Institute of Standards and Technology (NIST), "Ascon-Based Lightweight Cryptography Standards for Constrained Devices," *NIST Special Publication 800-232*, 2025. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/232/final

[8] G. A. Montoya, C. Lozano-Garzón, C. Paternina-Arboleda, and Y. Donoso, "A Mathematical Optimization Approach for Prioritized Services in IoT Networks for Energy-Constrained Smart Cities," *International Journal of Computers Communications & Control*, vol. 20, no. 1, article 6912, 2025. doi: 10.15837/ijccc.2025.1.6912.

[9] T. Zhou, X. Gao, X. Sun, and L. Han, "Split Difference Weighting: An Enhanced Decision Tree Approach for Imbalanced Classification," *International Journal of Computers Communications & Control*, vol. 19, no. 6, article 6702, 2024. doi: 10.15837/ijccc.2024.6.6702.

[10] N. Hasan and F. Ullah, "Adaptive Cryptographic Mechanisms for Secure IoT Communications," *IEEE Access*, vol. 12, pp. 28901–28915, 2024.

[11] A. Hamarsheh, M. Alazzam, and K. Elgazzar, "An Adaptive Security Framework for Internet of Things Using Software-Defined Networking and Machine Learning," *Applied Sciences*, vol. 14, no. 11, p. 4530, 2024.

[12] N. Gura *et al.*, "Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPUs," in *Proc. CHES*, LNCS 3156, pp. 119–132, 2004.

[13] S. R. Pokhrel, Y. Liu, and J. Park, "Machine Learning for Intelligent IoT Security: A Survey," *IEEE Communications Surveys & Tutorials*, early access, 2025.

[14] T. Nakamura and K. Sato, "Energy-Aware Secure Communication Mechanisms for Distributed IoT-Based Automation Systems," *International Journal of Automation Technology*, vol. 15, no. 6, pp. 842–851, 2021.

[15] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy and Trust in Internet of Things: The Road Ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[16] S. Radhakrishnan and P. Balasubramanian, "Performance Evaluation of Lightweight Cryptographic Algorithms for IoT Platforms," *IEEE Access*, vol. 12, pp. 34512–34526, 2024.

[17] M. Alshahrani, A. Alharbi, and S. Alotaibi, "A Lightweight Framework to Secure IoT Devices with Limited Resources," *Scientific Reports*, vol. 15, 2025.

[18] S. Kumar, A. Verma, and R. Singh, "Lightweight Cryptographic Solutions for Resource-Constrained IoT Devices," *Computers, Materials & Continua*, vol. 78, no. 2, pp. 2451–2470, 2024.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Haddadi, I. A. (2026). An Adaptive Hybrid Cryptographic Framework for Secure and Energy-Efficient IoT Systems: Architecture and Evaluation, *International Journal of Computers Communications & Control*, 21(5), 7451, 2026. https://doi.org/10.15837/ijccc.2026.5.7451

https://doi.org/10.15837/ijccc.2026.5.7451