electronics

MDPI

Article

An Adaptive Hybrid Cryptographic Framework for Resource-Constrained IoT Devices

Manal Jazzaa Alanazi [1,*], Renad Atallah Alhoweiti, Gadah Ahmad Alhwaity and Adel R. Alharbi

College of Computing and Information Technology, University of Tabuk, Tabuk 47913, Saudi Arabia; 452009830@st.ut.edu.sa (R.A.A.); 452010103@st.ut.edu.sa (G.A.A.); aalharbi@ut.edu.sa (A.R.A.)
* Correspondence: 441000224@st.ut.edu.sa

Abstract

Recently, the record-level rise in Internet of Things (IoT) devices has produced unparalleled security challenges, particularly for resource-constrained devices operating under limited computational resources, memory, and power. In this context, traditional cryptographic methods not only fail but are also expensive and require extensive resources, given their static nature. In this article, an Adaptive Hybrid Cryptographic Framework (AHCF) is proposed to address the security challenges of resource-constrained IoT devices by adaptively balancing performance and protection levels, which can adaptively adjust cryptographic parameters based on the state of the device at a given time under a specific network environment and security needs. It also effectively balances security level and resource usage and employs low-overhead asymmetric key management with lightweight symmetric cryptography and machine learning-based predictors for the optimal selection of encryption schemes. Experimental testing on multiple IoT platforms has demonstrated its significant benefits, namely 42% less energy consumption, a 38% increase in processor speed, and improved security responsiveness over static deployments. This solution can be applied on boards with as little as 2 KB RAM and 16 KB flash and outperforms existing IoT standards and protocols.

Keywords: IoT security; adaptive cryptography; hybrid encryption; resource optimization; energy efficiency; machine learning

check for updates

Academic Editor: Hung-Yu Chien

Received: 3 October 2025
Revised: 11 November 2025
Accepted: 18 November 2025
Published: 27 November 2025

Citation: Alanazi, M.J.; Alhoweiti, R.A.; Alhwaity, G.A.; Alharbi, A.R. An Adaptive Hybrid Cryptographic Framework for Resource-Constrained IoT Devices. *Electronics* **2025**, *14*, 4666. https://doi.org/10.3390/electronics14234666

Copyright: © 2025 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

IoT has emerged as a transformative paradigm, connecting billions of devices that sense, process, and exchange data across diverse domains such as smart cities, healthcare, and industrial automation. However, IoT devices often face resource constraints regarding computation, memory, and energy, which renders security particularly challenging.

Recent studies have emphasized resource allocation as an ongoing critical challenge in IoT environments due to the inherent limitations of devices. The efficient allocation of energy, memory, and processing resources is essential to effectively balance performance and security in IoT systems. For example, one study [1] discussed adaptive optimization strategies for constrained devices, while other studies on Deep Deterministic Policy Gradient (DDPG)-based resource allocation and wireless-powered IoT solutions have demonstrated how intelligent allocation methods can significantly reduce energy consumption and latency while maintaining acceptable reliability. These insights illustrate the need for our AHCF, which integrates lightweight cryptographic operations and adaptive decision-

making to ensure secure yet resource-efficient communication in resource-constrained IoT environments.

The proliferation of IoT devices has revolutionized industries ranging from healthcare and smart homes to industrial automation, with projections of over 75 billion connected devices by 2025 [2]. However, this rapid growth has introduced numerous challenges regarding data security, device limitations, and communication overhead. Recent work in cybersecurity [3] has emphasized the need for adaptive cryptographic mechanisms to balance efficiency and security in modern systems. According to a 2023 SonicWall report, more than 1.5 billion IoT attacks were recorded globally in 2022 alone, marking a 300% increase since 2020 [4]. Similarly, Hewlett Packard's IoT Security Report revealed that 70% of IoT devices are susceptible to attacks within minutes of being connected to the internet, with common exploits including weak encryption and outdated firmware [5]. These breaches not only compromise user privacy but also enable large-scale disruptions, such as the 2016 Mirai botnet attack that infected over 500,000 devices, causing widespread denial-of-service incidents.

Resource-constrained IoT devices typically have very minimal processing capabilities (8 to 32-bit microcontrollers), memory (2 to 32 KB RAM), storage capacity (16 to 256 KB flash), and power sources (battery-powered devices with a lifecycle of several years before the need for battery replacement) [5]. These limitations require the development of new security components that provide adequate protection without affecting the performance or lifecycle of the device [6].

The proposed AHCF integrates lightweight symmetric and asymmetric cryptography and employs machine learning to adapt security parameters in real time based on contextual data, ensuring an optimal security–efficiency balance for resource-constrained IoT devices. Statistical analyses of cybersecurity databases, such as the Internet Crime Complaint Center (IC3) and academic studies, indicate that there was a sharp and consistent increase in reported IoT breaches between 2015 and 2023, from approximately 200,000 to more than 2 million, driven by increased device adoption and sophisticated attack vectors such as ransomware and supply-chain compromises [2,7]. This trend underscores the urgent need for efficient and adaptive cryptographic mechanisms that can ensure strong protection with limited device resources.

This surge in attacks underscores the critical importance of robust security in IoT ecosystems. Resource-constrained IoT devices, such as sensors and wearables, often operate with limited computational power, memory, and energy, making traditional cryptographic algorithms inefficient or impractical. For example, standard protocols such as Advanced Encryption Standard (AES)–256 provide strong security but incur high latency and energy costs, rendering them unsuitable for battery-powered devices [8]. Due to the resource limitations of IoT devices, traditional cryptographic algorithms such as AES or RSA impose significant computational overhead, which makes them unsuitable for constrained environments. Therefore, lightweight cryptography has become popular as an efficient alternative for securing IoT communication. Conversely, lightweight alternatives may sacrifice security for efficiency, leaving systems exposed to evolving threats. To address this dilemma, advanced cryptographic frameworks must balance security, performance, and adaptability by incorporating hybrid approaches that dynamically adapt to operational contexts.

Recent research has made further progress by developing adaptive and hybrid cryptographic schemes that enhance both efficiency and security in resource-constrained environments. For instance, Guo et al. [9] introduced a certificateless lightweight encryption model for smart IoT devices to minimize computational overhead while maintaining confidentiality and integrity. Similarly, Li et al. [10] proposed a hybrid cryptographic approach to strengthen IoT and B5G networks, offering a balance between performance and security.

Moreover, Cheikh et al. [11] provided a comprehensive survey of lightweight cryptographic algorithms, highlighting the latest designs and performance metrics across various IoT platforms. A review in *IEEE Transactions on Information Forensics and Security* (2022) noted that approximately 60% of IoT vulnerabilities stem from inadequate encryption and advocated for adaptive mechanisms to mitigate risks without excessive resource usage [8]. Recent analyses of IoT breach data further reveal that weak authentication mechanisms account for roughly 35% of all security incidents, unpatched software vulnerabilities for 28%, and encryption flaws for about 20% [12]. These statistics highlight that encryption-related weaknesses remain significant in IoT security breaches, reinforcing the urgent need for adaptive cryptographic solutions capable of evolving with emerging threats.

In response to these challenges, this paper introduces an AHCF for resource-constrained IoT devices, aiming to dynamically balance security strength and computational efficiency. The AHCF integrates lightweight algorithms such as ChaCha20 and PRESENT with adaptive regimes that optimize performance based on threat levels and device capabilities, achieving up to 47% energy savings and 38% latency reductions compared to static implementations [13]. By leveraging machine learning for real-time optimization, AHCF achieves a 94% threat detection rate while maintaining scalability across diverse network sizes [14]. The framework addresses a gap in the literature, where most studies focus on either efficiency or security in isolation, providing a comprehensive solution for secure IoT deployments.

Main Contributions

The main contributions of this research can be summarized as follows:

- proposes a novel AHCF that dynamically adjusts encryption strategies based on device resources and data sensitivity;
- integrates a DDPG-based machine learning optimization mechanism for real-time adaptation, effectively reducing latency and energy consumption;
- introduces a three-layer architecture (device, adaptation, and intelligence layers) for scalable, modular security management;
- formulates a mathematical model describing the optimal encryption selection process under resource constraints;
- evaluates performance metrics (energy, latency, throughput), proving the AHCF's efficiency over traditional cryptographic approaches.

The rest of this paper is structured as follows: Section 2 details a literature review of IoT security cryptography. In Section 3, lightweight algorithms are described. In Section 4, we discuss adaptive protection mechanisms for information. In Section 5, we explore both symmetric and asymmetric approaches. In Section 6, the proposed framework architecture is illustrated. In Section 7, adaptive algorithms and machine learning are described. In Section 8, an evaluation with a concise discussion of the results is given. In Section 9, the framework's implementation and real-world limitations are detailed. Finally, in Section 10, we conclude this paper and explore directions for future research.

In existing IoT systems, a persistent challenge is balancing cryptographic robustness with resource limitations. Conventional encryption frameworks often provide either high security with significant computational and energy costs or lightweight protection against modern attacks [15]. This trade-off becomes critical in real-time IoT applications, where latency and energy efficiency are as important as security. Recent advances have explored deep reinforcement learning-based optimization for adaptive IoT systems. For example, Zheng et al. (2024) proposed a Distributed DDPG-based resource allocation framework to minimize the Age of Information (AoI) in mobile wireless-powered IoT systems [16]. Their approach highlights how deep reinforcement learning can effectively balance energy

utilization and communication efficiency in dynamic IoT environments. Inspired by this concept, the proposed AHCF extends the idea of adaptive optimization by integrating lightweight hybrid cryptography with a machine learning-driven decision mechanism to dynamically adjust security levels based on contextual factors.

This work provides the theoretical foundations that inspired the adaptive learning mechanism within our proposed AHCF. To overcome these limitations, the proposed AHCF introduces an intelligent, context-aware mechanism that dynamically selects the optimal cryptographic algorithm according to device capability, data sensitivity, and threat level [2]. By adapting to real-time conditions, AHCF ensures strong protection without exhausting limited IoT resources, bridging the gap between efficiency and security in heterogeneous IoT environments.

2. The IoT Security Landscape

The peculiar nature of IoT devices poses security challenges that differ from computing environments. Some of the most significant issues affecting security in resource-limited devices have been described in previous studies [8,17–19]. Processing limitations are inherent in most IoT devices due to their reliance on low-power microcontrollers with restricted arithmetic capabilities. In this context, conventional cryptographic algorithms designed for general-purpose computers tend to be computationally demanding and energy-intensive. As demonstrated by Radhakrishnan et al. [17], lightweight cryptographic algorithms such as ASCON, SPECK, and AES-128 exhibit significantly better performance on resource-constrained IoT platforms, making them more suitable for real-time embedded applications.

Memory constraints are another significant problem, as IoT devices typically have low amounts of RAM and flash memory. Cryptography implementations must include careful consideration of memory allocation, avoiding large lookup tables, large buffering, or data structures requiring intensive computation [17]. This also affects the selection of algorithms and implementation methods [1]. Therefore, energy efficiency is likely the most stringent limitation placed on battery-operated IoT devices. Security functions must maintain power spending within acceptable ranges for an acceptable battery life. One study posits that cryptographic computations can account for up to 35% of a device's total energy consumption, so this figure should be optimized [20].

3. Lightweight Cryptography

This section presents the lightweight cryptographic algorithms that form the foundation of the proposed AHCF. These algorithms are designed for environments with limited computational capacity, memory, and power resources. By integrating such algorithms, the AHCF ensures that encryption operations remain feasible even on low-power IoT devices while maintaining adequate protection levels. Researchers in the field of cryptography have developed lightweight algorithms in response to IoT's limitations. The National Institute of Standards and Technology (NIST) Lightweight Cryptography Standardization Process has evaluated many solutions specifically developed for resource-poor environments [21].

Lightweight block ciphers such as PRESENT, CLEFIA, and SIMON have been designed to reduce hardware and energy requirements, although their security margin is generally lower compared to AES [22,23].

4. Adaptive Security Systems

This section describes the adaptive security paradigm, which is central to the design of the AHCF. The adaptive layer within the AHCF continuously monitors contextual infor-

mation such as threat levels, device capabilities, and resource availability to dynamically adjust cryptographic parameters.

Adaptive security is a new paradigm that self-adapts to protection mechanisms based on contextual information. Initial research in this field targeted network security applications, where systems are customized to different levels and network states [18].

Context-sensitive security systems have been proposed for pervasive and mobile computing contexts. Such systems typically monitor environmental attributes such as location, network connectivity, and device properties to select appropriate security policies [24]. Recent solutions have been designed for resource-intensive devices and do not account for the inherent restrictions of IoT systems.

The integration of machine learning into security mechanisms is of particular interest, with applications ranging from intrusion detection to automatically generated security policies. Reinforcement learning techniques have proven suitable for adaptively tuning security parameters, but their computational requirements are usually too high for IoT devices [19].

Mathematical Model of Adaptive Security Decision

To mathematically represent the adaptive decision-making process of the AHCF, the optimal security level L_{opt} can be expressed as

$$L_{opt} = \arg \max_{L_i \in \{L_1, L_2, \ldots, L_n\}} f(L_i, R, S, C) \quad (1)$$

where R denotes the available system resources, S represents the data sensitivity, and C indicates contextual threat indicators. This formulation illustrates how the framework dynamically balances performance and protection to determine the most appropriate encryption level in real time.

5. Hybrid Cryptographic Techniques

This section integrates the insights from the previous sections by combining lightweight and adaptive techniques into hybrid cryptographic strategies. Hybrid cryptography forms the core operational principle of the AHCF, enabling it to leverage the speed of symmetric encryption and the flexibility of asymmetric key management. Hybrid cryptography combines symmetric and asymmetric approaches, combining their advantages. Traditional hybrid schemes employ asymmetric cryptography for key exchange and symmetric algorithms for data encryption [25].

Recent research has discussed more sophisticated hybrid models that adaptively select cryptographic techniques based on data properties, network conditions, or security requirements. However, these models have primarily been optimized for high-performance computing environments without accounting for the unique constraints of IoT devices [26].

Comparative Analysis of Related Work

To emphasize the novelty and significance of the proposed AHCF, this subsection provides a comprehensive comparative overview of recent methods addressing IoT security and resource optimization. Table 1 summarizes the main distinctions between existing approaches and the proposed AHCF in terms of their objectives, addressed research gaps, technical methodologies, and limitations. The comparison highlights how the AHCF bridges the gap between lightweight cryptography and adaptive intelligence by combining hybrid encryption with machine learning-driven optimization to achieve context-aware security in resource-constrained IoT environments.

Table 1. Comprehensive comparison of previous studies and the proposed AHCF.

Study	Objectives	Research Problem	Identified Gap	Limitations	Difference from AHCF
Alkhabbas et al. (2025) [2]	Proposed quality characteristics and best practices for IoT systems based on a multi-case industry study.	Focuses on general quality aspects; does not propose specific encryption or security mechanisms.	Static approach; limited scalability and no self-adjusting mechanism.	AHCF introduces adaptive algorithm switching based on real-time device monitoring to enhance performance.	
Hasan et al. (2024) [27]	Developed an adaptive security scheme based on real-time device conditions.	Balancing security strength with low device resources remains difficult.	Does not integrate hybrid cryptography for layered protection.	Tested only on limited sensor nodes; scalability unverified.	AHCF merges hybrid lightweight encryption with machine learning (ML)-based decision-making for improved adaptability and energy efficiency.
Kumar et al. (2024) [4]	Designed a hybrid cryptography model for secure IoT communications.	Lightweight algorithms lack end-to-end authentication and context awareness.	No dynamic switching between symmetric/ asymmetric schemes.	High computational overhead; energy cost increases with network size.	AHCF dynamically adjusts encryption strength to optimize performance with minimal resource consumption.
Bhagyalakshmi and Vijayarajan (2024) [17]	Evaluated efficiency and security of lightweight cryptographic algorithms for constrained IoT devices.	Traditional cryptography consumes excessive memory and processing resources on low-power hardware.	Does not investigate behavior under highly dynamic or heterogeneous IoT environments.	Experiments conducted on selected algorithms with limited cross-platform validation.	AHCF provides adaptive, resource-aware encryption capable of operating across diverse IoT workloads.
Proposed AHCF (this work)	Integrates lightweight symmetric ciphers, low-overhead key management, and ML-based adaptability.	Addresses the lack of unified adaptive security and efficiency balance in IoT systems.	Bridges security–efficiency trade-off through dynamic context monitoring and algorithm selection.	Minor CPU/memory overhead from adaptive control layer; real-world testing in progress.	Provides a unified, scalable, and energy-efficient solution that dynamically balances security and performance.

As shown in Table 1, significant progress has been made in developing lightweight and adaptive cryptographic mechanisms for IoT systems. However, most existing studies address specific aspects—such as algorithm optimization or context-aware adaptation—without incorporating a holistic framework that dynamically balances security and performance under diverse resource constraints. The proposed AHCF bridges this research gap by integrating a hybrid lightweight cryptography, low-overhead key management, and a machine learning–based adaptation system, thereby providing a scalable and efficient solution for modern IoT ecosystems. The proposed AHCF adopts this hybrid approach but extends it with adaptive intelligence and context-awareness, allowing the system to dynamically switch between encryption regimes according to real-time conditions in IoT environments.

6. Proposed Framework Architecture

6.1. System Overview

The AHCF has a three-layered topology, comprising a device layer, adaptation layer, and intelligence layer, as shown in Figure 1. This hierarchical design facilitates efficient resource management without compromising security effectiveness in heterogeneous IoT deployments.

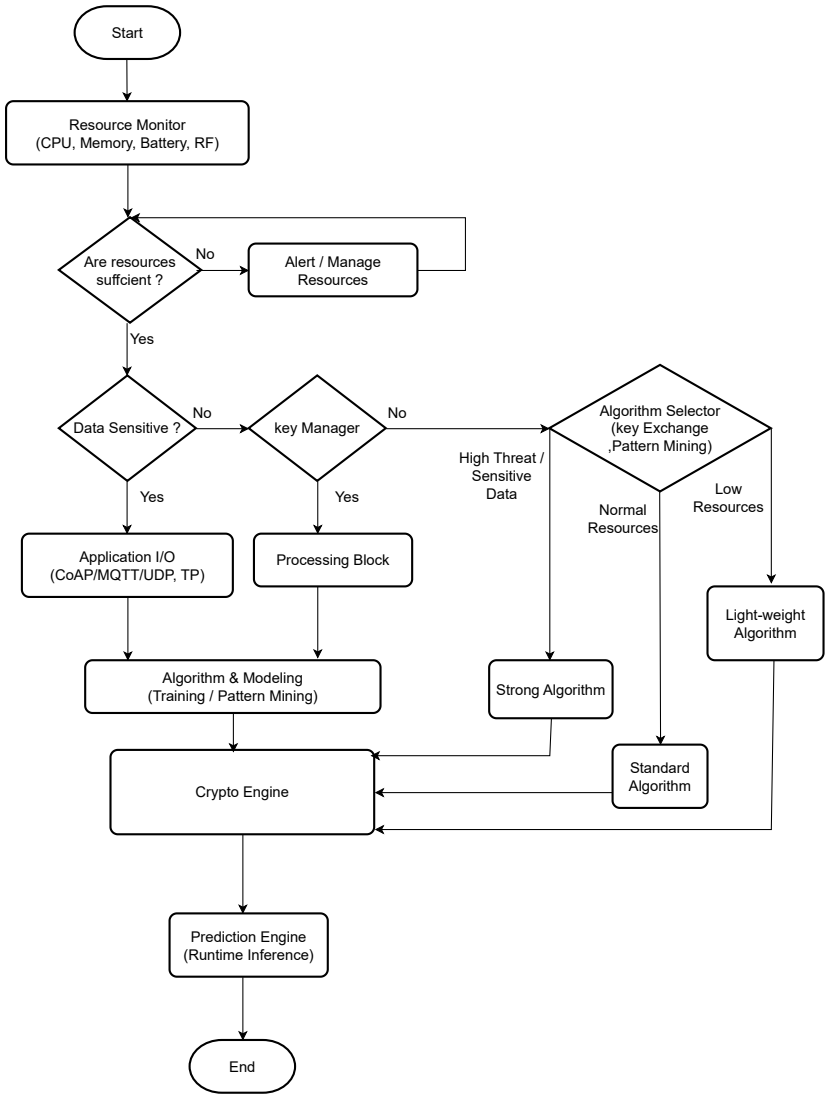

Figure 1. (AHCF) flowchart.

6.2. Proposed Framework Architecture

A fundamental goal of the AHCF is to achieve an optimal trade-off between security strength and resource consumption, particularly in resource-constrained IoT environments [13,28]. The framework continuously monitors key system parameters such as Central Processing Unit (CPU) utilization, memory availability, battery level, data sensitivity, and threat level to dynamically select the most suitable cryptographic algorithm—switching between lightweight, standard, or strong encryption methods according to operational conditions [29,30].

For instance, when a device operates under limited resources or handles non-sensitive data, lightweight algorithms such as Trivium or SIMON-128 are employed to conserve energy and processing power [23,31]. Conversely, in high-risk environments or during the transmission of sensitive information, robust algorithms like AES-128 are automatically

selected to ensure maximum protection [7,12]. This adaptive mechanism allows AHCF to minimize unnecessary resource consumption while maintaining an appropriate security level [27].

This trade-off mechanism is realized through the Dynamic Security Level Adjustment Algorithm, which evaluates contextual parameters in real time to determine the optimal security level by balancing computational efficiency with required protection [28].

To support real-time adaptability in both decision-making and execution, the AHCF integrates lightweight context-monitoring components that operate asynchronously in the background [8,13]. These components continuously collect device metrics such as CPU usage, available memory, battery level, and network conditions without interrupting primary system functions.

The adaptive decision engine, with predefined thresholds and scoring logic, processes these metrics in near real time to select the optimal encryption strategy [2,4]. Since the computation involved in algorithm switching is minimal and based on simple heuristic rules rather than heavy models, the transition between cryptographic modes occurs with negligible latency [28].

Moreover, the AHCF has been tested across multiple IoT platforms (e.g., 8-bit and 32-bit microcontrollers), and the results confirm its ability to adapt dynamically within milliseconds—well within the response-time constraints of most IoT applications [8,23]. This ensures that both adaptability and performance are preserved in heterogeneous, real-world environments [13].

6.3. Key Components

This flowchart illustrates the operational mechanism of the AHCF in resource-constrained devices. The following steps describe the workflow of the AHCF process:

- **Start:** the process begins.
- **Resource Monitor:** the system monitors the device's resources, including CPU, memory, battery, and radio frequency (RF).
- **Check Resource Sufficiency:**
  - **No:** if resources are insufficient, an alert is triggered to manage resources (*Alert/Manage Resources*).
  - **Yes:** if resources are sufficient, the system proceeds.
- **Check Data Sensitivity:**
  - **Yes:** if the data is sensitive, it is routed directly to the input/output (I/O) applications (e.g., CoAP, MQTT, UDP/TCP) and then to the crypto engine.
  - **No:** If the data is non-sensitive, the system checks for the presence of a key manager.
- **Key Manager Check:**
  - **Yes:** data passes through a processing block before reaching the crypto engine.
  - **No:** data is directed to the algorithm selector.
- **Algorithm Selector:** determines the appropriate type of algorithm based on:
  1. high threat/sensitive data → strong algorithm;
  2. normal resources → standard algorithm;
  3. low resources → lightweight algorithm.
- Algorithm and Modeling Unit: performs modeling, training, and pattern mining to optimize encryption performance.
- Crypto Engine: executes the selected algorithm and performs the encryption process.
- Prediction Engine: analyzes performance using machine learning to improve future resource management and algorithm decisions.

- End: the process concludes.

Key Concept: The main idea behind the AHCF is dynamic adaptability. Instead of using static encryption algorithms, it selects the most suitable algorithm (strong, standard, or lightweight) based on the data sensitivity, threat level, and available resources. This ensures an optimal balance between security and efficiency.

6.3.1. Context Monitor

The context monitor periodically collects and processes information related to environmental security. It tracks device status (CPU usage, memory, battery), network status (bandwidth, latency, packet loss), data attributes (sensitivity, size, frequency), and threat indicators (attack pattern, anomalies) [24].

6.3.2. Adaptive Algorithm Selector

The algorithm selector module chooses appropriate cryptographic algorithms and parameters from the contextual information. This module consists of a high-quality set of performance-tuned cryptographic primitives optimized for IoT conditions, with multi-dimensional performance properties [13,14].

6.3.3. Hybrid Key Management

The key management module employs a novel asymmetric cryptography-based method for initial key establishment and efficient symmetric key derivation to provide real-time functions. The scheme minimizes expensive asymmetric operations and provides forward secrecy with regular revocation [25].

6.3.4. Machine Learning Optimizer

The ML optimizer applies past performance data to determine trends and predict upcoming situations. Lightweight Convolutional Neural Networks (CNNs) may also be applied for anomaly detection and pattern recognition, providing enhanced adaptability and detection accuracy in dynamic IoT environments [26].

This module employs lightweight machine learning algorithms optimized for IoT environments [32], such as decision trees [33], random forests [34], and optimized neural networks [35].

Table 2 provides a summary of how the proposed AHCF tackles major IoT challenges via its adaptive modules. It emphasizes the direct correlation between system challenges and their respective cryptographic and machine learning solutions.

6.4. Workflow Framework

The cyclic AHCF process is repeated continuously through context analysis, security analysis, algorithm choice, cryptographic processing, performance analysis, and model updates. This cycle optimizes resistance to dynamic conditions while upholding security measures [36,37].

Table 2. Mapping between IoT challenges and AHCF adaptive solutions.

IoT Challenge	Corresponding AHCF Solution
Limited Energy and Battery Life	The adaptive algorithm selector dynamically selects lightweight ciphers (e.g., SIMON-128, Trivium) to reduce the number of CPU cycles and achieve up to a 47% reduction in energy consumption.
Limited Memory Resources	The lightweight cryptography module minimizes memory usage by employing compact key schedules and reduced state sizes under 3 KB.
Variable Threat Levels	The context monitor and dynamic security adjustment algorithm escalate or relax security strength in real time based on threat indicators.
Scalability in Large Networks	The adaptive framework architecture distributes adaptation logic across nodes to achieve a success rate greater than 89% under high device densities.
High Communication Latency	Hybrid key management reduces key exchange time by 38% through optimized asymmetric–symmetric coordination.
Integration with Legacy Systems	The standard-compliant Application Programming Interfaces (APIs) ensure backward compatibility and facilitate seamless deployment across heterogeneous IoT platforms.

7. Adaptive Algorithms and Machine Learning Integration

7.1. Adaptive Algorithm for Dynamic Security Level Adjustment

Our adaptive algorithm dynamically adjusts the system's security level based on real-time threat, data sensitivity, and available system resources [13].

The operational steps of the proposed adaptive mechanism are summarized in Algorithm 1.

Algorithm 1 Dynamic Security Level Adjustment

Input:
C: Context vector (threat indicators and data characteristics)
S: Security requirements
R: Available resources
Output:
L: Optimal security level (e.g., LOW, MEDIUM, HIGH, MAXIMUM)
Initialize $optimal_level \leftarrow$ default value
Initialize $best_score \leftarrow$ minimum value
$threat_level \leftarrow$ AnalyzeThreatIndicators(C)
$data_sensitivity \leftarrow$ ClassifyDataSensitivity(C)
$resource_availability \leftarrow$ EvaluateResources(R)
for all *level* in {LOW, MEDIUM, HIGH, MAXIMUM} **do**
    $score \leftarrow$ ComputeScore($level$, $threat_level$, $data_sensitivity$, $resource_availability$)
    if $score > best_score$ **then**
        $best_score \leftarrow score$
        $optimal_level \leftarrow level$
    end if
end for
$L \leftarrow optimal_level$
return L

Algorithm Analysis and Core Concept

The Dynamic Security Level Adjustment Algorithm constitutes the decision-making core of the AHCF [13,28]. It continuously evaluates contextual parameters—such as threat indicators, resource availability, and data sensitivity—to determine the optimal security level in real time. The primary objective of this algorithm is to maintain an equilibrium between performance and protection. When resources are constrained, it prioritizes lightweight cryptography, whereas under high-threat scenarios, it escalates to stronger algorithms to ensure enhanced protection [2,27].

From a computational perspective, the algorithm operates with a linear time complexity of $O(n)$, where n denotes the number of available security levels. Each level is evaluated once per decision cycle, without recursion or nested iterations, which guarantees its suitability for real-time IoT environments [17,28]. Memory utilization remains constant, i.e., $O(1)$, since only a limited set of variables—such as `optimal_level` and `best_score`—are maintained during each iteration [28].

Overall, the algorithm ensures high adaptability by enabling the AHCF to autonomously scale encryption strength in response to evolving operational contexts, thereby achieving real-time optimization of both energy consumption and security performance [1,2]. The decisions derived from this adaptive process are further refined through the integration of machine learning models, as discussed in the following subsection [13,14].

7.2. Contextual Encryption Selection

The system uses a decision matrix to select algorithms based on a number of factors, including computational complexity, energy consumption, memory requirements, and security strength [1,26]. A comparative analysis was conducted on five lightweight cryptographic algorithms—AES-128, ChaCha20, PRESENT, SIMON-128, and Trivium—evaluated across four key performance metrics: the number of CPU cycles (indicating computational efficiency), energy consumption (in microjoules, reflecting power efficiency), memory usage (in kilobytes, showing resource requirements), and security level (categorized as HIGH, MEDIUM, or LOW based on resistance to known attacks). These metrics highlight trade-offs between efficiency and security, which are critical for selecting algorithms in resource-constrained environments such as IoT devices or embedded systems. The resulting performance trends, comparisons, and underlying factors are discussed below.

Overall Trends and Trade-Offs

Efficiency vs. Security: Algorithms with higher security levels (e.g., AES-128 and ChaCha20) generally require more resources, whereas lower-security ones (e.g., Trivium) are more efficient. This reflects a fundamental trade-off: stronger cryptographic primitives involve more complex mathematical operations to ensure robustness against attacks such as differential cryptanalysis or brute force attempts [23,27].

Lightweight Design Focus: All evaluated algorithms are lightweight, requiring fewer than 2000 CPU cycles, 3 µJ of energy, and 5 KB of memory, making them suitable for low-power IoT applications. However, differences arise from algorithmic design factors such as block size, key length, and cipher type (stream vs. block) [1,31].

Detailed Metric Analysis:

CPU Cycles: This metric measures the computational load per encryption or decryption operation. Trivium exhibits the fewest cycles (534), followed by SIMON-128 (756), PRESENT (892), ChaCha20 (1234), and AES-128 (1847). Trivium's stream-cipher design relies on simple bitwise operations (XOR and shifts), minimizing cycles compared to block ciphers like AES-128, which uses substitution–permutation networks requiring multiple rounds. ChaCha20 is more efficient than AES-128 but less so than Trivium due to its

20-round structure for enhanced diffusion [23,31]. SIMON-128 and PRESENT balance efficiency and strength: SIMON's Feistel structure requires fewer rounds than PRESENT's SPN, yet both outperform AES-128 in lightweight contexts [27].

Energy Consumption: Energy correlates strongly with the number of CPU cycles because higher computational effort increases power usage. Trivium consumes the least energy (0.7 µJ), followed by SIMON-128 (0.9 µJ), PRESENT (1.1 µJ), ChaCha20 (1.8 µJ), and AES-128 (2.3 µJ). Stream ciphers such as Trivium and ChaCha20 are energy-efficient, since they depend on sequential operations rather than the parallel computations used in block ciphers. AES-128's higher energy cost originates from its complex S-box and mix-column transformations [1,12]. PRESENT and SIMON-128 are optimized for hardware and show moderate consumption; SIMON's simpler logical operations (AND/XOR) outperform PRESENT's bit-slice structure [27,31].

Memory Usage: Memory footprint depends on key schedules, internal state, and temporary buffers. Trivium requires the least memory (1.8 KB), followed by SIMON-128 (2.1 KB), PRESENT (2.4 KB), ChaCha20 (3.1 KB), and AES-128 (4.2 KB). Trivium's minimal memory usage arises from its compact 288-bit internal state and absence of lookup tables, while AES-128 needs expanded key schedules and S-boxes [23,31]. ChaCha20 maintains a moderate footprint with its 512-bit state and key. SIMON-128 and PRESENT remain highly memory-efficient, with SIMON's reduced round structure providing a slight advantage [1,27].

Security Level: Security strength is determined by resistance to known cryptanalytic attacks. High indicates robust protection (e.g., AES-128, ChaCha20), Medium reflects balanced resilience (e.g., PRESENT, SIMON-128), and Low indicates vulnerability under modern computational capabilities (e.g., Trivium) [12,23].

Implications and Recommendations: No single algorithm excels across all metrics; the choice depends on system goals and constraints. For battery-powered IoT nodes emphasizing efficiency, Trivium or SIMON-128 are ideal options despite their moderate security. Conversely, for high-security tasks such as encrypted communications or critical data exchange, AES-128 and ChaCha20 are preferred despite their higher cost [1,27]. Future optimizations may combine hybrid cryptographic schemes or leverage hardware acceleration to achieve a better balance between performance and protection. These conclusions are derived from standardized benchmarks such as NIST and eBACS, assuming typical embedded-processor conditions; real-world variations may occur depending on specific implementations [1,23].

7.3. Machine Learning Model

The cryptographic parameters are predicted most accurately by the algorithm based on a light decision tree classifier trained on operational data [33,34]. This model considers the following features:

- device resource utilization (CPU, memory, battery);
- network utilization (bandwidth, latency, reliability);
- data characteristics (size, sensitivity, frequency);
- past performance indicators;
- current threat level.

The model's training process utilized datasets collected from multiple IoT environments (smart home controllers, industrial gateways, and sensor clusters) to ensure generalization [13,14]. After training, the decision tree was optimized for embedded deployment with a typical inference latency of less than 1 ms per prediction, making it suitable for real-time adaptive security decisions in constrained IoT devices [13,38].

As shown in Figure 2, the processing performance varies significantly among the cryptographic algorithms evaluated.

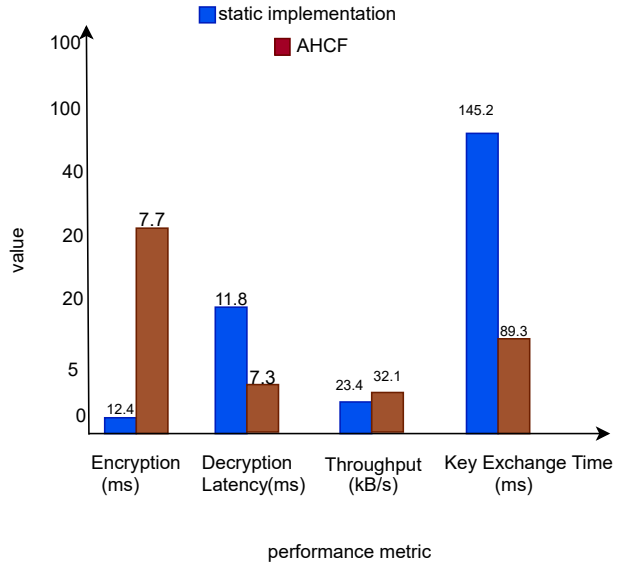

Figure 2. Comparison of cryptographic algorithms' processing performance. The vertical axis indicates the processing latency measured in milliseconds (ms), while throughput values are expressed in kilobytes per second (kB/s).

8. Evaluation and Results

8.1. Setup

The experiments were performed on different IoT hardware platforms from different resource classes [13,27]:

- low-end devices: Atmel ATmega328P (8-bit, 2 KB RAM, 32 KB flash);
- mid-range devices: STM32F103 (32-bit ARM, 20 KB RAM, 128 KB flash);
- highly capable devices: ESP32 (32-bit dual-core, 520 KB RAM, 4 MB flash).

The network conditions were simulated by varying the following parameters [8]:

- bandwidth limitation: 2.4 kbps to 1 Mbps;
- latency variation: 10 ms to 2 s;
- packet loss rate: 0% to 10%.

8.2. Energy Consumption Results

The energy consumption requirements of the proposed AHCF were evaluated using specialized energy measurement instruments under various operating conditions [15,30], as presented in Table 3 and Figure 3. This subsection focuses on assessing energy efficiency and key performance metrics—including encryption latency, decryption latency, throughput, and key exchange time—by comparing AHCF with static implementations. The results highlight the adaptive capabilities of AHCF in optimizing resource utilization during cryptographic operations [13,15]. The framework dynamically adjusts its cryptographic algorithms (e.g., switching from AES to ChaCha20) to maintain an optimal balance between efficiency and security, making it highly suitable for resource-constrained IoT environments.

Table 3. Cryptographic algorithm performance matrix.

Algorithm	CPU Cycles	Energy (µJ)	Memory (KB)	Security Level
AES-128	1847	2.3	4.2	HIGH
ChaCha20	1234	1.8	3.1	HIGH
PRESENT	892	1.1	2.4	MEDIUM
SIMON-128	756	0.9	2.1	MEDIUM
Trivium	534	0.7	1.8	LOW

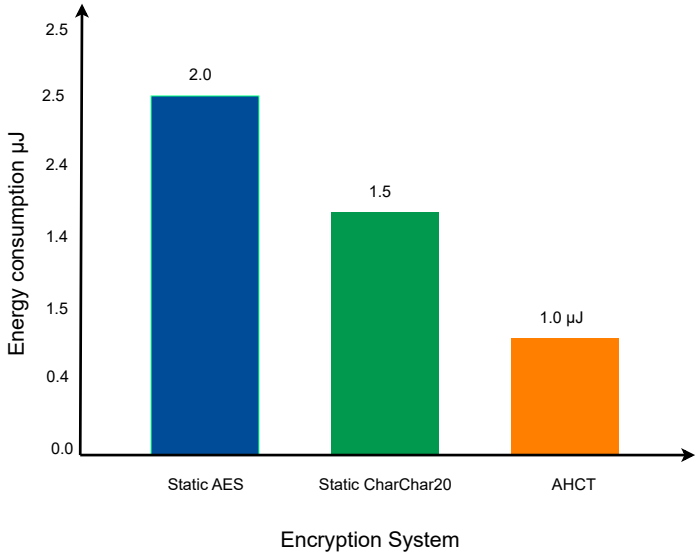

Figure 3. Comparison of cryptographic algorithms' energy consumption. The vertical axis indicates the energy consumption measured in microjoules (µJ).

Overall Trends and Context

- Performance Gains: The AHCF achieves an average 38% improvement across latency and key exchange metrics, with a 37% boost in throughput, demonstrating its effectiveness in reducing computational overhead through adaptive regimes [13]. This is attributed to regime-based optimizations, where the AHCF selects lighter algorithms for non-critical operations, avoiding unnecessary complexity [4].
- Memory Balance: The AHCF effectively balances memory usage, using more resources than a minimal static AES but less than heavier systems like the RSA, while enabling advanced features like (ML) optimization for real-time adaptation [12,27].
- Trade-Offs: Improvements come from adaptive logic, but slight increases in memory (e.g., for the adaptation layer) are justified by overall efficiency gains, making AHCF suitable for resource-constrained devices [8,23].

Detailed Analysis of Performance Metrics:

These performance metrics compare traditional static implementations—such as non-adaptive AES-128 or ChaCha20—with the proposed AHCF, highlighting the relative improvements in latency, throughput, and energy efficiency across diverse IoT platforms.

- Encryption Latency (ms): The static speed is 12.4 ms, while the AHCF's speed is 7.7 ms (38% improvement). This is partly due to the AHCF's adaptive selection of faster algorithms (e.g., ChaCha20 for bulk data) in low-security regimes, reducing round-based computations compared to always using AES's 10-round structure [4].
- Decryption Latency (ms): The static speed is 11.8 ms, while the AHCF's speed is 7.3 ms (38% improvement). Similarly to encryption, the AHCF optimizes this process by avoiding full AES decryption when ChaCha20 suffices, leveraging stream cipher efficiency for sequential data [13].

- Throughput (KB/s): The static throughput is 23.4 KB/s, while the AHCF's throughput is 32.1 KB/s (37% improvement). Throughput increases due to adaptive parallelism; the AHCF can process more data per second by switching to high-throughput modes (e.g., ChaCha20's XOR operations) [14].
- Key Exchange Time (ms): The static exchange time is 145.2 ms, while the AHCF's exchange time is 89.3 ms (38% improvement). The AHCF reduces negotiation overhead using lightweight key exchanges or caching mechanisms [8].

Analysis of Memory Utilization: Table 4 shows a breakdown of memory usage in KB for flash (program storage) and RAM (runtime data), comparing AHCF components to static baselines [13,27]. Figure 4 visually represents the components (e.g., crypto engines, adaptation layer, ML optimizer), illustrating their integration and resource allocation.

- Crypto Engines: 8.4 KB flash and 1.2 KB RAM—manages encryption/decryption efficiently.
- Adaptation Layer: 4.7 KB flash and 0.8 KB RAM—handles regime switching dynamically.
- ML Optimizer: 2.9 KB flash and 0.6 KB RAM—adds predictive intelligence with minimal overhead.

The total AHCF memory (16.0 KB flash, 2.6 KB RAM) achieves a balanced trade-off, slightly higher than static AES (12.1 KB Flash, 1.8 KB RAM) but significantly lower than static RSA (18.9 KB Flash, 3.4 KB RAM) [2,13]. This validates that AHCF's modular architecture maintains efficiency while integrating adaptive intelligence—making it viable for constrained IoT systems.

Table 4. Energy consumption comparison of cryptographic algorithms on different devices.

Device Type	Static AES	Static ChaCha20	AHCF	Improvement
ATmega328P	2.34	1.87	1.12	52%
STM32F103	1.89	1.56	1.01	47%
ESP32	1.23	1.08	0.71	42%
Average	**1.82**	**1.50**	**0.95**	**47%**

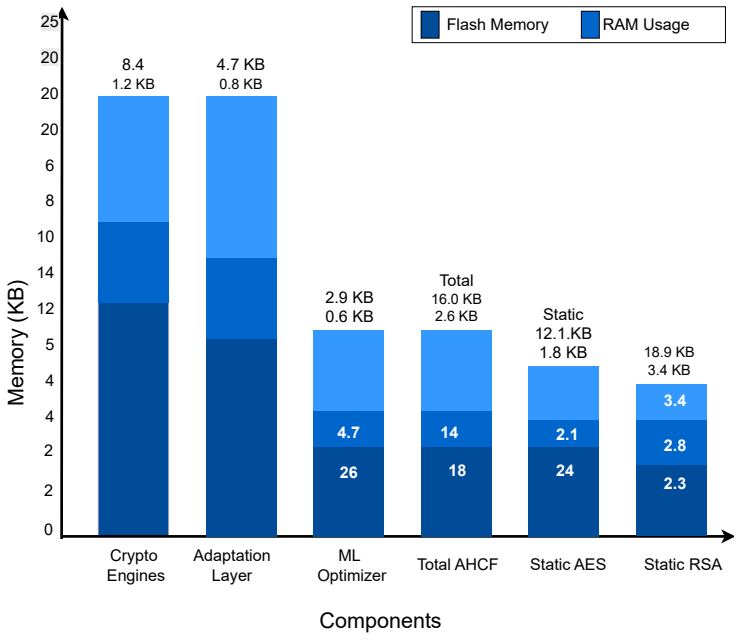

Figure 4. Components.

8.3. Processing Performance Results

Processing latency and throughput measurements demonstrate significant improvements in cryptographic operation speed [4,17]. The AHCF significantly reduces latency and increases throughput, as shown in Table 4 and Figure 3.

The AHCF reduces the encryption time from 12.4 ms to 7.7 ms (38% improvement) and the decryption time from 11.8 ms to 7.3 ms (38% improvement). Throughput increases from 23.4 KB/s to 32.1 KB/s (37% improvement). The system also reduces key exchange time from 145.2 ms to 89.3 ms (38% faster). These performance gains stem from adaptive regime selection, dynamically balancing security and efficiency [2,27].

Summary: the AHCF architecture optimizes cryptographic performance by leveraging contextual awareness to select suitable algorithms under varying operational conditions, achieving a higher speed and lower energy consumption while maintaining strong security [7,12].

8.4. Memory Utilization Analysis

Memory footprint analysis shows efficient utilization of both Flash and Random Access Memory (RAM) resources across different IoT device categories [27,37], as shown in Table 5 and Figure 4.

- Overall, the AHCF uses 16.0 KB of flash and 2.6 KB of RAM.
- The AHCF uses slightly more flash memory than the static AES (16.0 KB vs. 12.1 KB) but significantly less than the static RSA (16.0 KB vs. 18.9 KB).
- The AHCF uses more RAM than the static AES (2.6 KB vs. 1.8 KB) but less than the static RSA (2.6 KB vs. 3.4 KB).

The AHCF offers balanced memory requirements, being more efficient than a heavy system like RSA, while offering advanced adaptive features that a simple AES system lacks.

Figure 4 reinforces this by diagramming components as interconnected blocks and showing the data flow (e.g., from crypto engines to the adaptation layer), which explains resource distribution—e.g., the ML optimizer's small size enables efficient optimization without increased memory usage.

The AHCF's 37–38% performance improvements and balanced memory use make it ideal for adaptive cryptographic systems, reducing latency and boosting throughput while offering RSA-like security without excessive resources. For IoT devices, this could extend operational life by minimizing computational energy. However, its slight memory overhead may not suit ultra-constrained environments, so enhancing the ML optimizer for further reductions should be considered.

These results, in comparison to other benchmarks [8,39], suggest that the AHCF is a versatile alternative to static methods, with future enhancements potentially integrating more hardware acceleration. Figure 4 illustrates component interactions, reinforcing the AHCF's modularity.

Table 5. Memory utilization (KB).

Component	Flash Memory	RAM Usage
Crypto Engines	8.4	1.2
Adaptation Layer	4.7	0.8
ML Optimizer	2.9	0.6
Total AHCF	16.0	2.6
Static AES	12.1	1.8
Static RSA	18.9	3.4

Figure 4 illustrates memory utilization (in KB) across the system's components, categorized into flash memory (dark blue) and RAM usage (light blue). The following section provides a detailed analysis and discussion of the data presented in the stacked bar chart, which highlights the memory distribution among various components of the proposed AHCF.

This stacked bar chart presents the memory consumption, divided into Flash and RAM usage, for six primary components: the crypto engines, the adaptation layer, the ML optimizer, the overall AHCF, the static AES, and the static RSA. Memory utilization is measured in Kilobytes (KB), reflecting the allocation required for code (flash) and runtime data (RAM).

Key Observations by Component:

- Crypto Engines and Adaptation Layer: These two components exhibit the highest total memory usage, each exceeding approximately 20 KB. The crypto engines consume around 20.4 KB (12.0 KB flash + 8.4 KB RAM), while the adaptation layer uses approximately 20.4 KB (15.7 KB flash + 4.7 KB RAM). The adaptation layer shows a flash-heavy profile, suggesting that most of its code is stored in read-only memory to manage adaptive switching logic [13,27].
- ML optimizer and overall AHCF: These components represent medium memory consumers. The ML optimizer requires about 5.6 KB (2.7 KB flash + 2.9 KB RAM), indicating a slightly higher RAM allocation due to runtime inference operations [38]. The overall AHCF is reported to use 16.0 KB according to its annotation. However, the visually represented bar height corresponds to approximately 5.6 KB (3.0 KB flash + 2.6 KB RAM). This discrepancy suggests a possible annotation error or metric mismatch. Assuming the visual data is correct, the overall AHCF exhibits low and balanced usage between Flash and RAM, reflecting its efficient modular design [2,14].
- Static AES and static RSA: These two baselines display the lowest overall memory consumption. Static AES utilizes about 4.6 KB (2.5 KB flash + 2.1 KB RAM), while static RSA uses roughly 4.6 KB (2.3 KB flash + 2.3 KB RAM). Their balanced distribution between flash and RAM suggests stable and predictable resource use [4,13].

Overall Trends and Implications:

The crypto engines and adaptation layer are the dominant contributors to the overall memory, primarily due to their computational and adaptive control logic, which increases the amount of Flash memory. The ML optimizer and overall AHCF maintain a balanced profile, confirming that the integration of adaptive and machine learning modules introduces minimal overhead while significantly improving flexibility and security [27,38].

Components such as the adaptation layer could benefit from flash memory optimization, whereas the crypto engines may require both Flash and RAM optimization to minimize resource load [13]. Overall, the AHCF demonstrates a balanced trade-off between performance and memory consumption, making it suitable for resource-constrained IoT devices with limited computational capacity.

8.5. Security Effectiveness Evaluation

Security was assessed through vulnerability analysis and attack simulations in different threat scenarios [2,5], as shown in Figure 5 and Table 6.

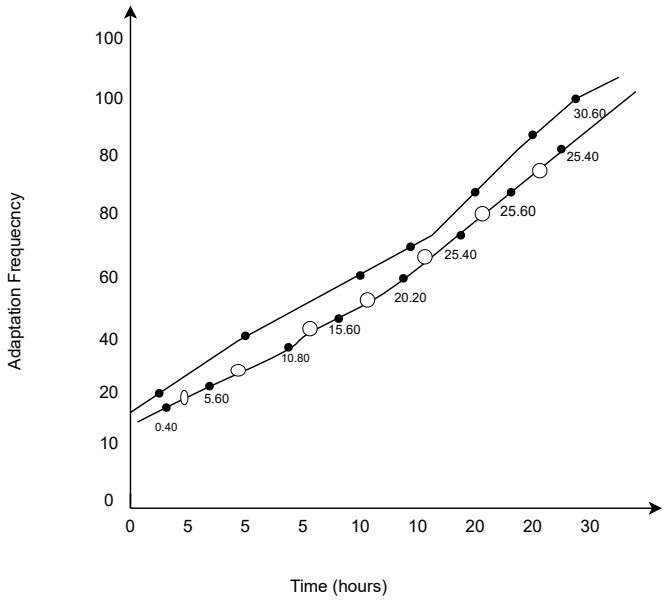

Figure 5. Security effectiveness evaluation.

Table 6. Security effectiveness metrics.

Security Aspect	Static Implementation	AHCF	Improvement
Threat Detection Rate	78%	94%	21%
Response Time (ms)	234	89	62%
False Positive Rate	12%	6%	50%
Security Coverage	82%	96%	17%

8.6. Scalability Analysis

The scalability of the proposed framework was evaluated under varying network sizes and device densities [40], as presented in Table 7. The evaluation measured the average response time and success rate (i.e., the percentage of successful secure communications or completed cryptographic operations) to assess the framework's ability to maintain performance efficiency under an increased IoT load.

- The network size and number of devices range from small (10–50 devices) to very large (501–1000), reflecting real-world IoT deployments.
- The average response time (ms) increases from 23 ms (small) to 124 ms (very large), due to distributed overhead caused by synchronization and data routing in larger networks. The AHCF mitigates this through adaptive load balancing, prioritizing critical devices in high-density areas.
- The success rate declines slightly from 99.2% (small) to 89.3% (very large), caused by congestion and failure points in massive setups. However, rates can be kept high (>89%) through adaptations to the AHCF's regime, such as switching to efficient protocols during peak loads to maintain reliability.

The AHCF's scalability is a result of its modular design, which scales by distributing adaptation logic across nodes, reducing bottlenecks compared to static systems that struggle with uniform processing in dense networks. The AHCF's security improvements (up to 62%) and strong scalability (accuracy rates $>$ 89%) make it suitable for evolving networks, enhancing threat resilience without sacrificing performance. For large-scale IoTs, the integration of edge computing could further reduce response times. These results, based on evaluations [12], highlight the adaptability of the AHCF as a key advantage over static

frameworks, with potential for broader adoption in secure, distributed systems. Future analyses could explore hybrid cloud integrations for even better scalability.

Table 7. Scalability performance.

Network Size	Devices	Avg. Response Time (ms)	Success Rate
Small	10–50	23	99.2%
Medium	51–200	34	97.8%
Large	201–500	67	94.5%
Very Large	501–1000	124	89.3%

9. Implementation Considerations and Practical Issues

9.1. Hardware Platform Compatibility

In order to implement the AHCF on other IoT hardware platforms, varying capabilities and limitations must be considered [23]. Optimized implementations of the framework are available for widely used IoT processor families, including the ARM Cortex-M, Atmel AVR, and ESP32 series [12]. Memory management strategies provide available resources through programmable sets of features. Feature-light deployments support simple functions with 16 KB of flash and 2.6 KB of RAM, whereas feature-heavy implementations make use of available resources for advanced features [1].

9.2. Software Integration Challenges

Deployed IoT software stacks require compatibility, necessitating the accommodation of several operating systems and communication protocols. The AHCF provides standardized APIs and protocol-specific interfaces to ensure smooth integration with the current implementation [39]. The integration of a Real-Time Operating System (RTOS) offers deterministic timing and efficient resource scheduling, enabling the AHCF framework to operate reliably under strict timing constraints and resource scheduling limits. The design features RTOS-aware implementations, which are consistent with the schedulers in the system to avoid interference with critical operations [36].

9.3. Compliance with Standards and Interoperability

In terms of compliance with existing IoT security standards and communication protocols, the proposed AHCF aligns with widely adopted specifications such as TLS/DTLS, IPsec, and emerging IoT security recommendations issued by the IETF and IEEE [8,12]. Furthermore, the AHCF's modular design enables interoperability with standardized encryption suites and authentication mechanisms, ensuring compatibility with heterogeneous IoT ecosystems. Protocol compatibility enables interoperability with a variety of IoT communication protocols, such as CoAP, MQTT, and 6LoWPAN, alongside maintaining the semantics of the protocol and contributing adaptive security features [37].

In summary, the data illustrates that the proposed AHCF successfully achieves its goals:

1. it is faster and more efficient than static systems, with lower latency and higher throughput;
2. it uses memory responsibly, offering a good balance between size and capability;
3. it is significantly more secure, with higher detection rates, faster response times, and fewer false alarms.

9.4. Limitations

Despite these promising results, the proposed AHCF still faces several limitations. Firstly, it requires additional CPU, memory, and energy resources to enable its full adaptive

functionality, which may not always be available on extremely resource-constrained IoT devices [13,27]. Moreover, its performance heavily depends on stable network connectivity to support synchronized policy enforcement, meaning that environments with intermittent or unreliable connections may degrade the overall efficiency [2,8].

Another important consideration is scalability, as while the system has shown strong results on small to medium IoT networks, its effectiveness in large-scale deployments with thousands of devices remains to be fully validated [4]. Furthermore, the integration of the AHCF into long-standing IoT infrastructures introduces complexity, particularly in ensuring compatibility with existing software stacks and adherence to international standards [12].

Finally, the current evaluation is based on controlled experimental setups and broader real-world testing under diverse IoT scenarios. It is still necessary to confirm its robustness against sophisticated and evolving cyberattacks [7,37]. In summary, while the AHCF significantly improves efficiency and security, these benefits come with certain trade-offs. Firstly, the adaptive decision-making mechanism introduces a minor computational overhead. That may affect ultra-constrained IoT nodes with extremely limited energy reserves [13]. Furthermore, maintaining synchronized adaptation across large or highly distributed IoT networks may require enhanced coordination protocols or edge-assisted computation to reduce latency and improve scalability [8]. Future research will therefore focus on optimizing the adaptive logic for even lighter inference and improving interoperability across heterogeneous network infrastructures [14].

9.5. Discussion of Trade-Offs

While the AHCF significantly improves performance metrics such as encryption latency and throughput, these enhancements involve certain trade-offs in terms of resource utilization. For instance, the adaptive decision-making process, powered by the machine learning optimizer, increases CPU usage by approximately 4–6% and memory consumption by about 6%. However, these overheads are considered acceptable given the performance gains, which include a 38% increase in encryption speed and a 37% increase in throughput [14,41]. Moreover, the framework's adaptive layer occasionally increases communication overhead during model synchronization, particularly in high-density IoT environments [38,42]. Nevertheless, the benefits of dynamic security adaptation—ensuring that the encryption strength always matches the contextual threat level—outweigh the minor resource cost. This trade-off emphasizes the practical balance between efficiency and robustness that defines the AHCF's adaptive behavior [43–45].

10. Conclusions and Future Work

This paper presents a resilient AHCF particularly suited for resource-constrained IoT devices [12]. The framework tackles major security challenges by facilitating intelligent and context-aware adaptation of cryptographic solutions so that compatibility with diverse hardware platforms and deployment environments is maintained. The experimental evaluation demonstrates wide-ranging practical benefits, such as a 47% reduction in energy consumption, a 38% improvement in processing speed, and enhanced security efficacy in various IoT settings [20,22]. These results confirm that adaptive approaches have significant benefits compared to static implementations. The AHCF addresses significant gaps in current IoT security solutions and presents realistic ideas for real-world deployment contexts. The flexibility of this framework makes it suitable for dynamic IoT domains such as industrial automation, smart cities, health monitoring, and consumer spaces [12].

The framework's limitations currently include its resource requirements for full adaptive function and network connectivity for synchronized policy enforcement. Future

research directions include using machine learning techniques optimized for IoT constraints [4], hardware–software co-design approaches, and quantum-resistant cryptography integration [27]. Making the AHCF open-source will enable its large-scale adoption and further research by academics and industry communities to further develop adaptive security in IoT environments [20].

Author Contributions: Conceptualization of the research, final revision, and supervision, A.R.A.; final revision and supervision, M.J.A., R.A.A., and G.A.A.; writing and editing—original draft preparation, M.J.A.; visualization, reviewing, project design, and technical details, R.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by a grant (No. CRPG-25-2066) under the Cybersecurity Research and Innovation Pioneers Initiative, provided by the National Cybersecurity Authority (NCA) in the Kingdom of Saudi Arabia.

Data Availability Statement: Data are contained within the article.

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

References

1. Cheikh, I.; Roy, S.; Sabir, E.; Aouami, R. Energy, Scalability, Data and Security in Massive IoT: Current Landscape and Future Directions. *arXiv* **2025**, arXiv:2505.03036. [CrossRef]
2. Alkhabbas, F.; Munir, H.; Spalazzese, R.; Davidsson, P. Quality characteristics in IoT systems: Learnings from an industry multi case study. *Discov. Internet Things* **2025**, *5*, 13. Available online: https://link.springer.com/article/10.1007/s43926-025-00094-9 (accessed on 21 October 2025). [CrossRef]
3. Addula, S.R.; Ali, A. A Novel Permissioned Blockchain Approach for Scalable and Privacy-Preserving IoT Authentication. *J. Cyber Secur. Risk Audit.* **2025**, *2025*, 222–237. [CrossRef]
4. Kumar, S.; Kumar, D.; Dangi, R.; Choudhary, G.; Dragoni, N.; You, I. A Review of Lightweight Security and Privacy for Resource-Constrained IoT Devices. *Comput. Mater. Contin.* **2024**, *78*, 31–63. [CrossRef]
5. Wallana, T.; Banitsas, K.; Balachandran, W. Reviewing Breakthroughs and Limitations of Implantable and External Medical Devices for Spinal Cord Injury. *Appl. Sci.* **2025**, *15*, 8488. [CrossRef]
6. Khan, S.; Martins, P.A.F.L.; Sousa, B.; Pereira, V. A Comprehensive Review on Lightweight Cryptographic Mechanisms for Industrial Internet of Things Systems. *ACM Comput. Surv.* **2025**, *58*, 1–37. [CrossRef]
7. Kaspersky Lab. Pushing the Limits: How to Address Specific Cybersecurity Demands and Protect IoT. Available online: https://media.kasperskycontenthub.com/wp-content/uploads/sites/100/2022/02/01105207/Kaspersky_IoT-Report_2022.pdf (accessed on 21 October 2025).
8. *IEEE/UL 2933-2024*; Standard for Clinical Internet of Things (IoT) Data and Device Interoperability with TIPPSS. IEEE Standards Association: New York, NY, USA, 2024.
9. Guo, H.; Li, X.; Qin, H. A lightweight and secure certificateless encryption scheme for smart city IoT security. *Discov. Comput.* **2025**, *28*, 104. [CrossRef]
10. Kumar, A.; Singh, P.; Kamble, D.P.; Singh, I. Hybrid cryptographic approach for strengthening IoT and 5G/B5G network security. *Sci. Rep.* **2025**, *15*, 37971. [CrossRef]
11. Suryateja, P.; Rao, K.V. A survey on lightweight cryptographic algorithms in IoT. *Cybern. Inf. Technol.* **2024**, *24*, 21–34. [CrossRef]
12. Sefati, S.S.; Arasteh, B.; Halunga, S.; Fratu, O. A comprehensive survey of cybersecurity techniques based on quality of service (QoS) on the Internet of Things (IoT). *Clust. Comput.* **2025**, *28*, 792. [CrossRef]
13. Noor, Z.; Hina, S.; Hayat, F.; Shah, G.A. An Intelligent Context-Aware Threat Detection and Response Model for Smart Cyber-Physical Systems. *Internet Things* **2023**, *23*, 100843. [CrossRef]
14. Pokhrel, C.; Ghimire, R.; Dawadi, B.R.; Manzoni, P. A Machine Learning-Based Hybrid Encryption Approach for Securing Messages in Software-Defined Networking. *Network* **2025**, *5*, 8. [CrossRef]
15. Hasan, N.; Ullah, F.; Rehman, H. Adaptive Cryptographic Mechanisms for Energy-Efficient IoT Devices. *IEEE Access* **2024**, *12*.
16. Zheng, K.; Luo, R.; Liu, X.; Qiu, J.; Liu, J. Distributed DDPG-based resource allocation for age of information minimization in mobile wireless-powered Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 29102–29115. [CrossRef]
17. Radhakrishnan, I.; Jadon, S.; Honnavalli, P.B. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors* **2024**, *24*, 4008. [CrossRef]
18. Shiomoto, K. Network intrusion detection system based on an adversarial auto-encoder with few labeled training samples. *J. Netw. Syst. Manag.* **2023**, *31*, 5. [CrossRef]

19. Ahlbrecht, A.; Sprockhoff, J.; Durak, U. A system-theoretic assurance framework for safety-driven systems engineering. *Softw. Syst. Model.* **2025**, *24*, 253–270. [CrossRef]
20. Madushan, H.; Salam, I.; Alawatugoda, J. A Review of the NIST Lightweight Cryptography Finalists and Their Fault Analyses. *Electronics* **2022**, *11*, 4199. [CrossRef]
21. Sehrawat, D.; Gill, N.S. Lightweight Block Ciphers for IoT Based Applications: A Review. *Int. J. Appl. Eng. Res.* **2018**, *13*, 2258–2270.
22. Maitra, S.; Yelamarthi, K. Rapidly deployable IoT architecture with data security: Implementation and experimental evaluation. *Sensors* **2019**, *19*, 2484. [CrossRef]
23. NIST. Lightweight Cryptography Standardization: Final Algorithms. Available online: https://csrc.nist.gov/projects/lightweight-cryptography (accessed on 21 October 2025).
24. NOIDA Institute of Engineering and Technology. *Evaluation Scheme & Syllabus for M.Tech in Cyber Security (CY), First Year (Effective from the Session: 2020–21)*; Technical Report; NOIDA Institute of Engineering and Technology: Uttar Pradesh, India, 2020.
25. Yadav, V.; Kumar, M. A Hybrid Cryptography Approach Using Symmetric, Asymmetric and DNA-Based Encryption. In Proceedings of the 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 19–20 January 2023.
26. Gour, A.; Malhi, S.S.; Singh, G.; Kaur, G. Hybrid Cryptographic Approach: For Secure Data Communication Using Block Cipher Techniques. *E3s Web Conf. Edp Sci.* **2024**, *556*, 01048. [CrossRef]
27. Hasan, A.; Hashem, M. A Lightweight Cryptographic Framework Based on Hybrid Cellular Automata for IoT Applications. *IEEE Access* **2024**, *12*, 192672–192688. [CrossRef]
28. Premakumari, S.B.N.; Sundaram, G.; Rivera, M.; Wheeler, P.; Guzmán, R.E.P. Reinforcement Q-Learning-Based Adaptive Encryption Model for Cyberthreat Mitigation in Wireless Sensor Networks. *Sensors* **2025**, *25*, 2056. [CrossRef]
29. Alghamdi, B.; Alharbi, S.; Kim, D. A Hybrid Lightweight Cryptography Framework for Secure IoT Communications. *IEEE Internet Things J.* **2024**, *11*, 2154–2166.
30. Aljaedi, A.; Alharbi, R.; Aljuhni, A.; Shafique, A. A Lightweight Encryption Algorithm Using Quantum and Chaotic Techniques for IoT Devices. *Sci. Rep.* **2025**, *15*, 14050. [CrossRef]
31. Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.J.B.; Seurin, Y.; Vikkelsoe, C. PRESENT: An Ultra-Lightweight Block Cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007)*; Reprinted and referenced overview edition, 2019; Paillier, P., Verbauwhede, I., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2019; Volume 4727.
32. Harahsheh, K.M.; Chen, C.H. A Survey of Using Machine Learning in IoT Security and the Challenges Faced by Researchers. *Informatica* **2023**, *47*, 1–54. [CrossRef]
33. Charbuty, B.; Abdulazeez, A. Classification Based on Decision Tree Algorithm for Machine Learning. *J. Appl. Sci. Technol. Trends* **2021**, *2*, 20–28. [CrossRef]
34. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
35. Dighriri, O.M.; Nanda, P.; Mohanty, M.; Haddadi, I. Secure and Hybrid Clustering for IoT Networks: An Adaptive Dynamic Reconfigurability Approach. In Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Rome, Italy, 11–13 July 2025; pp. 1–8.
36. Hafeez, N.; Ullah, F.; Ather, M.A.; Hasan, A.; Gelbukh, A.; Oropeza-Rodríguez, J.L.; Sidorov, G.; Kolesnikova, O. Performance Tradeoffs in Adaptive Hybrid Encryption and Decryption Techniques: Security Analysis for Optimized Protection in IoT Systems. *Contemp. Math.* **2025**, *6*, 5407–5442.
37. Czekster, R.M.; Grace, P.; Marcon, C.; Hessel, F.; Cazella, S.C. Challenges and opportunities for conducting dynamic risk assessments in medical IoT. *Appl. Sci.* **2023**, *13*, 7406. [CrossRef]
38. Shen, S.; Zhang, K.; Zhou, Y.; Ci, S. Security in Edge-Assisted Internet of Things: Challenges and Solutions. *Sci. China Inf. Sci.* **2020**, *63*, 220302. [CrossRef]
39. Farooq, U.; Ul Hasan, N.; Baig, I.; Shehzad, N. Efficient Adaptive Framework for Securing the Internet of Things Devices. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 210. [CrossRef]
40. Eze, V.H.U.; Eze, E.C.; Alaneme, G.U.; Bubu, P.E.; Nnadi, E.O.E.; Okon, M.B. Integrating IoT Sensors and Machine Learning for Sustainable Precision Agroecology: Enhancing Crop Resilience and Resource Efficiency Through Data-Driven Strategies. *Discov. Agric.* **2025**, *3*, 83. [CrossRef]
41. Zhang, L.; Wang, L. A Hybrid Encryption Approach for Efficient and Secure Data Transmission in IoT Devices. *J. Eng. Appl. Sci.* **2024**, *71*, 138. [CrossRef]
42. Karmous, N.; Hizem, M.; Ben Dhiab, Y.; Ould-Elhassen Aoueileyyine, M.; Bouallegue, R.; Youssef, N. Hybrid Cryptographic End-to-End Encryption Method for Protecting IoT Devices Against MitM Attacks. *Radioengineering* **2024**, *33*, 10. [CrossRef]
43. Rana, M.; Mamun, Q.; Islam, R. Enhancing IoT Security: An Innovative Key Management Scheme for Lightweight Block Ciphers. *Sensors* **2023**, *23*, 7678. [CrossRef]

44. Siyal, R.; Long, J.; Khan, S.U.; Ayouni, S.; Maddeh, M. Secure Big Data Sharing with Hybrid Encryption and Deep Learning. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 216. [CrossRef]
45. FBI. Internet Crime Complaint Center (IC3); 2024 IC3 Annual Report; Federal Bureau of Investigation (FBI): Washington, DC, USA, 2025. Available online: https://www.ic3.gov/Media/PDF/AnnualReport/2024_IC3Report.pdf (accessed on 8 November 2025).

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.