International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =7.938, July-December 2024, Submitted in September 2024, ISSN -2393-8048

Intelligent Security–Energy Trade-Off Optimization Using Adaptive Cryptography for Resource-Constrained IoT Networks

Sohail, Research Scholar, Computer Science Dept., Sikkim Alpine University, Kamrang, Sikkim
Dr. Sunil Gupta, Associate Professor, Computer Science Dept., Sikkim Alpine University, Kamrang, Sikkim

Abstract

There are now billions of IoT devices, all with limited resources like battery life, processing speed, and memory, due to the rapid expansion of the network. Security for these devices must be top-notch, but traditional cryptographic methods are not practical for sensor nodes that run on batteries because of the energy and latency overheads they cause. In order to dynamically balance the opposing goals of data security and energy efficiency in IoT networks, this study suggests an Adaptive Security-Energy Optimization (ASEO) architecture. Based on real-time parameters such as residual battery level, data sensitivity classification, network threat level, and channel conditions, the framework uses a context-aware decision engine to choose from a portfolio of lightweight cryptographic primitives. These primitives include AES-128, PRESENT, SPECK, and Elliptic Curve Cryptography (ECC). By iteratively improving the selection criteria, a controller based on reinforcement learning can converge on an optimal operating point on the security-energy Pareto frontier. By simulating a network of diverse Internet of Things (IoT) nodes, we find that the suggested architecture improves network lifetime compared to fixed-cipher baselines by about 41% and reduces average energy consumption by up to 34.6% compared to static AES-256 encryption while keeping an acceptable security margin. Smart optimization-guided adaptive cryptographic switching provides a feasible and scalable solution for safe and long-term Internet of Things (IoT) installations, according to the findings.

Keywords: Internet of Things, Adaptive Cryptography, Energy Efficiency, Lightweight Security, Reinforcement Learning, Resource-Constrained Devices

1. Introduction

With the integration of tens of billions of devices in smart homes, industrial automation, precision agriculture, healthcare monitoring, and intelligent transportation systems, the Internet of Things (IoT) has gone from being a specialized research idea to a cornerstone of contemporary digital infrastructure. Embedded controllers, sensors, and actuators often have to work with very limited resources, such as a few megahertz (MHz) of processing power, a few kilobytes of memory instead of gigabytes, and a finite amount of energy from tiny batteries or energy harvesting devices. Strong cryptographic protection is more important than ever before since IoT networks convey sensitive data more and more often. This data includes medical readings, industrial control directives, location traces, and financial transactions.

The problem is that traditional security designs were made for devices that had a lot of power and processing power. Executing AES-256, RSA-2048, or TLS 1.3 handshakes on all nodes without regard to their resource constraints rapidly drains battery life, raises end-to-end latency, and may make time-critical applications unworkable. Alternatively, in an effort to conserve energy, reducing cryptographic protection across the board leaves the network vulnerable to eavesdropping, manipulation, spoofing, and denial-of-service assaults. This study primarily aims to solve the security-energy trade-off problem, which is characterized by the tension between energy sustainability and security assurance.

Adaptive intelligence can dynamically reconcile security and energy efficiency, not that the two should be considered mutually conflicting design goals. The most important thing to remember is that different types of data, situations, and danger levels do not always require the same degree of cryptographic strength. Unlike firmware updates and actuator commands, which pose significant risks in safety-critical industrial settings, a temperature reading relayed every five minutes from a low-risk environment does not necessitate nearly as much protection. In order to achieve sufficient security while preserving energy wherever possible, an IoT node can choose the most appropriate cryptographic primitive from a broad portfolio by

continuously evaluating contextual signals such as residual energy, data sensitivity, network threat indicators, and channel quality.

2. Related Work

2.1 Lightweight Cryptographic Primitives

For extremely low-resource settings, including RFID tags and sensor nodes, *Bogdanov et al. (2007)* presented PRESENT, an extremely lightweight block cipher. Finding the sweet spot between cryptographic security and incredibly cheap hardware implementation cost was their main research objective. The researchers constructed PRESENT with a tiny hardware footprint of about 1,570 gate equivalents in one compact configuration, thanks to its compact substitution-permutation architecture. Their findings demonstrated that small embedded systems may not be able to handle the processing demands of traditional general-purpose cryptography methods. The findings proved that intentionally simplified cipher structures may offer practical security while significantly lowering implementation complexity and solidified PRESENT's position as an influential lightweight cipher.

To tackle the vast variety of limited software and hardware settings, *Beaulieu et al. (2013)* suggested the SIMON and SPECK families of lightweight block ciphers. With SIMON catering to hardware implementations and SPECK to software execution efficiency, they set out to create versatile cryptographic algorithms that could handle a variety of block and key sizes. Due to the wide variety of processor architectures, memory capacities, communication needs, and power availability among IoT and embedded systems, the study found that a single inflexible cipher configuration might not work as well as intended. Their findings showed that even on limited platforms, lightweight cipher families might keep up with the competition in terms of performance without sacrificing implementation flexibility. This study is important for adaptive IoT security because it lends credence to the idea that different devices and different operations call for different cryptographic techniques, rather than using a cookie-cutter approach.

To address the challenges posed by resource-constrained computing, *Dobraunig, Eichlseder, Mendel, and Schläffer* created the Ascon family of lightweight authenticated-encryption and hashing algorithms. Using efficient cryptographic primitives with a small internal structure, the primary goal was to offer authentication, integrity, and confidentiality. In 2023, after a complex worldwide review, NIST chose Ascon as their lightweight cryptography standard. In 2023, they published NIST SP 800-232, which detailed Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128. The review found that security, implementation efficiency, adaptability, and performance are more important metrics to consider when evaluating lightweight cryptographic systems than just execution speed. The end result confirmed the case for integrating standardized lightweight primitives into adaptive IoT encryption frameworks and solidified Ascon as a crucial standardized foundation for safeguarding limited devices.

Since elliptic-curve methods can offer robust public-key security with far less parameter sets than conventional RSA configurations while maintaining equivalent levels of security, research on ECC has also proven its applicability to constrained-network security. Reducing the computational, communication, and storage cost of authentication, digital signatures, and key agreement is the primary goal of ECC-based implementations in IoT-oriented applications. To illustrate the disparity in parameter needs between the two families of cryptography, NIST recommendations support elliptic-curve configurations like P-256 and P-384 in addition to substantially larger RSA modulus sizes. Research focusing on NIST P-256's performance has demonstrated that, with the right optimizations, a 256-bit elliptic curve can achieve a security level of around 128 bits. These findings highlight the importance of ECC in situations where memory consumption and communication overhead are major limitations; nonetheless, it is still necessary to take into account computational cost, side-channel resistance, and implementation security.

2.2 Security Protocols that Consider Energy Use

The goal of the seminal work by *Perrig et al. (2002)* on SPINS: Security Protocols for Sensor Networks was to create a security architecture that could function with wireless sensor networks that had limited resources. With an eye toward reducing communication and computing overhead, the suggested architecture integrated authentication, data secrecy, freshness, and authenticated broadcasting techniques. Their main point was that traditional network security protocols are not a good fit for sensor nodes because of their low processing power, memory, bandwidth, and battery life. We found that symmetric-key-based methods, if built correctly, might control communication and energy costs while still providing critical security services. This work laid the groundwork for a crucial principle for Internet of Things security studies: security methods should not be considered as separate system layers, but rather created in tandem with resource management concerns.

For WSNs, *Park and Shin (2004)* put out the Lightweight Security Protocol (LiSP), whose stated goal was to strike a balance between security and energy efficiency via key-refresh and efficient key management. Repeated cryptographic operations and frequent key establishment can significantly drain battery power in sensor nodes, according to their research. Thus, the suggested protocol used structured key refreshing to save needless cryptographic cost while maintaining a satisfactory degree of security. Instead than relying only on algorithm selection for energy-security trade-off management, the results demonstrated that protocol-level management was feasible. This discovery has important implications for adaptive encryption studies since it proves that timing, frequency, and environmental factors impact encryption cost in addition to cipher complexity.

In an effort to design a security architecture that took energy efficiency and robustness into account at the same time, *Etoweissy et al. (2004)* studied secure operation in WSNs. The researchers found that nodes with limited resources cannot run resource-intensive security operations indefinitely without reducing the lifespan of the network. Their findings highlighted the need of reducing superfluous communication and processing and wisely distributing security functions. In spite of sensor nodes' energy constraints, the resultant framework proved that security architecture could be organized to increase resilience. Research like this lends credence to the general idea that secure wireless and IoT systems can not afford to make energy consumption an afterthought during the design process.

Wireless sensor networks can benefit from the energy-efficient key-management and key-exchange systems that *Doerr et al. (2019)* compared and contrasted. Instead of depending solely on theoretical complexity, they aimed to experimentally evaluate lightweight protocols on hardware capable of sensing networks. According to the research, even when applied to theoretically equivalent systems with limited resources, security methods might display vastly divergent computational, communication, and energy behaviors. Their findings showed that cryptographic appropriateness is context dependent and that platform-specific benchmarking is important. These results strongly indicate that adaptive security systems that may take device-specific resource conditions into account when choosing cryptographic operations are necessary.

2.3 Context-Aware and Adaptive Security

Moving away from static cryptographic setups and toward security methods that can adapt to real-time operational circumstances is the new paradigm of context-aware security. Predefined rules were the backbone of early adaptive approaches. These rules would do things like ramp up security measures when threat indicators went above a certain threshold, simplify computations when battery life was low, or trigger extra authentication steps in response to suspicious activity. In cases where the operational risk did not warrant the corresponding resource expense, these approaches sought to avoid continuously maintaining maximum security overhead. They found that the energy consumption of devices, the volume of traffic, the criticality of applications, the state of channels, and the potential security risks are all constantly changing in IoT environments. The main drawbacks of rule-based systems are the

fact that their thresholds typically need human intervention and that they might not be able to successfully adapt to different types of devices and deployment settings.

By applying reinforcement-learning methods to access control and battery prediction in energy-harvesting IoT systems, *Chu, Li, Liao, and Cui (2018)* showcased the potential of reinforcement learning for energy-aware IoT decision-making. Their study's overarching goal was to find ways to improve communication choices without fully understanding the dynamics of the surrounding environment. The authors used reinforcement-learning techniques, such as deep Q-network-based approaches, and represented the decision problem using a Markov decision process. Based on their findings, static policies are not an effective solution to resource-management issues in the IoT because these systems are always evolving. As far as access control and battery-related decision tasks are concerned, experimental results showed that learnt policies might surpass benchmark approaches. Despite the fact that cryptographic adaptation was not their primary emphasis, their work offers solid methodological backing for optimizing security-energy tradeoffs in dynamic IoT situations through the application of reinforcement learning.

For the purpose of managing energy-efficient Internet of Things networks, *Khairy et al. (2020)* used limited deep reinforcement learning. Their goal was to maximize the network's performance over the long run while specifically limiting its energy consumption. Instead of relying solely on predetermined rules, an intelligent agent can learn control policies by rephrasing the problem as a limited Markov decision process. Incorporating energy restrictions directly into the learning formulation is necessary for conventional reward-based optimization to guarantee resource constraints, as their observations shown. The results of the simulation showed that the network's performance improved with time while the energy efficiency was intact. Because cryptographic selection is conceptually analogous to a restricted optimization issue, where the goal is to optimize security while avoiding violations of computational, latency, or battery constraints, this discovery has direct conceptual significance to adaptive encryption.

With an eye on Internet of Things (IoT) security, *Uprety and Rawat (2021)* examined RL/DL applications. Their objective was to investigate how learning-based approaches may deal with various types of cyberattacks and adjust security decisions to suit evolving IoT settings. Because an agent may engage with its environment, monitor results, and incrementally enhance its decision policy with less prior knowledge, they noted that reinforcement learning differs from traditional static security systems. While they did find reinforcement learning to be an intriguing method for cyber-physical system and dynamic IoT security, they did note several difficulties with computational overhead, state-space complexity, convergence, and actual deployment. In security decisions that necessitate constant adaptation instead of fixed configuration, their findings lend credence to the use of reinforcement learning.

When it comes to cybersecurity, *Ozkan-Okay et al. (2023)* looked at the bigger picture, including how AI and ML apply to IoT settings. To what extent may intelligent approaches enhance security monitoring, attack detection, and dynamic response? That was the goal. Based on their findings, security systems that use learning-based strategies are better able to adapt to evolving threats than those that rely solely on static processes. However, there are a number of difficulties that arise with AI-based security, including computing demands, training-data quality, explainability, implementation complexity, and hostile manipulation. This research lends credence to the argument that AI should play a role in the decision-making process that governs the distribution of security resources, rather than only detecting assaults.

3. System Model and Problem Formulation

3.1 Network Model

Our model is based on a diverse Internet of Things (IoT) network where N nodes with limited resources are spread out across a monitored area and linked to a gateway or edge node by a low-power wireless protocol, such as IEEE 802.15.4 or Bluetooth Low Energy (BLE). The computing capabilities C_i, memory budget M_i, remaining battery energy $E_i(t)$, and data

sensitivity classification assigned to each transmitted message make form a tuple that identifies each node i. Whenever a node receives a control command or senses data, it must perform a cryptographic operation from its available cipher suite before transmitting the data.

3.2 Energy Consumption Model

All four of node i's energy consumption categories—sensing, cryptographic processing, radio transmission, and idle/listening—are added together to provide the overall energy consumed during a transmission event. The official way:

$$E_{total} = E_{sense} + E_{crypto}(k) + E_tx(l) + E_{idle}$$

where $E_{crypto}(k)$ denotes the energy consumed by cryptographic algorithm k, which is a function of the algorithm's computational complexity (measured in clock cycles per byte) and the platform's energy-per-cycle characteristic. $E_tx(l)$ denotes transmission energy as a function of payload length l, which increases when authentication tags, initialization vectors, or larger key material must be transmitted alongside ciphertext. This decomposition allows the framework to estimate, ex ante, the marginal energy cost of selecting a stronger versus a lighter cryptographic primitive for a given message.

3.3 Security Model

We can determine the normalized security strength score S(k) for each cryptographic primitive k by looking at its key length, resistance to known cryptanalytic attacks (differential, linear, and algebraic cryptanalysis), and the number of rounds relative to the minimum rounds required to resist best-known attacks. There is a minimal acceptable security threshold S_{min} for each of the three categories of data sensitivity: low, medium, and high. This model presupposes an adversary with limited computational resources who can engage in active message injection, selective node compromise, and passive eavesdropping, but who will not be able to crack well-parameterized standard primitives (like AES) within the network's operational lifetime.

3.4 Problem Formulation

To reduce cumulative network energy consumption while achieving security strength that meets or surpasses the sensitivity-dependent threshold, the basic optimization challenge is to choose a cryptographic primitive k from the available suite K for each transmission event.

$$minimize \ \Sigma E_{crypto}(k_i) + E_tx(k_i) \quad subject\ to \ S(k_i) \geq S_{min}(data_i), \ for\ all\ i$$

This is a constrained, multi-objective, time-varying optimization problem because $E_i(t)$, threat level, and channel quality change dynamically. Section 4 presents the ASEO framework designed to solve this problem online, using local and learned information rather than requiring global, centralized optimization at every timestep.

4. Proposed Framework: Adaptive Security-Energy Optimization (ASEO)

4.1 Architecture Overview

The ASEO framework is made up of four layers that work together. The first layer, the Context Monitoring Layer, continuously samples things like residual battery level, radio link quality, and workload characteristics. The second layer, the Security Classification Layer, uses application-defined policies to assign a sensitivity level to outgoing data. The third layer, the Adaptive Decision Engine, chooses the cryptographic primitive and key parameters for each transmission. And finally, the fourth layer, the Lightweight Cryptographic Execution Layer, is responsible for actually doing things like encryption, authentication, and key management. Together with the Decision Engine, there is a Policy Adaptation Module that uses reinforcement learning to update the selection policy on a periodic basis depending on energy expenditure and security incidents detectable.

4.2 Context-Aware Cryptographic Selection Engine

Using a learnt policy $\pi(a_t \mid s_t)$, the Decision Engine assigns a cryptographic action $a_t \in$ {AES-128, PRESENT-80, SPECK-64/128, ECC-160} based on the current state vector $s_t = (E_i(t)$, sensitivity, threat_level, link_quality). Battery level (five bins), threat level (three levels obtained from an anomaly-detection subsystem or intrusion alerts), and sensitivity (three tiers) are all discretized into a reasonable number of bins in the state space. This keeps the decision table compact enough to store on hardware that is restricted.

4.3 Lightweight Cipher Suite

There is a wide range of energy-security operating points covered by the framework's cipher portfolio, as shown in Table 1. The widely-used and thoroughly-tested standard is AES-128. For low-sensitivity telemetry, PRESENT-80 provides minimal energy cost as an ultra-lightweight substitution-permutation network. An ARX-based encryption, SPECK-64/128 strikes a good mix between software speed and security. With smaller key sizes than RSA while maintaining comparable strength, ECC-160 provides public-key operations for key exchange and digital signatures in situations where asymmetric cryptography is inevitable.

Table 1. Cryptographic primitives evaluated within the ASEO cipher suite

Algorithm	Key Size (bits)	Type	Avg. Energy/Block (μJ)	Relative Security Score	Typical Use Case
AES-128	128	Symmetric (SPN)	5.4	0.85	General-purpose, medium/high sensitivity
PRESENT-80	80	Symmetric (SPN)	1.9	0.55	Low-sensitivity telemetry, high-frequency sensing
SPECK-64/128	128	Symmetric (ARX)	2.6	0.75	Balanced workloads, moderate sensitivity
ECC-160	160	Asymmetric (ECC)	38.7	0.90	Key exchange, authentication, critical commands
AES-256 (baseline)	256	Symmetric (SPN)	8.1	0.97	Static high-security baseline for comparison

4.4 Reinforcement-Learning-Based Adaptation

For each potential cipher, the Policy Adaptation Module uses a lightweight Q-learning formulation where the Q-table links discretized states to anticipated long-term reward. A positive energy penalty plus a term for security compliance make up the reward function r_t:

$$r_t = -\alpha \cdot E_{crypto}(a_t) + \beta \cdot [S(a_t) \geq S_{min}] - \gamma \cdot penalty_{violation}$$

according to deployment priorities, network operators can adjust the weighting coefficients α, β, and α to make the framework prioritize energy conservation or security assurance. If a chosen cipher does not meet the minimum security threshold for the specified sensitivity tier, a strong negative reward is applied by $penalty_{violation}$. With minimal computational overhead on the constrained node, the Q-table update rule can be executed directly using standard temporal-difference learning with a learning rate η and a discount factor γ_d. Alternatively, it can be periodically synchronized with an edge gateway that updates policies across multiple nodes in batch and sends back refined policies. This is due to the small state-action space.

4.5 Trade-off Optimization Algorithm

Algorithm 1 summarizes the online operation of the ASEO Decision Engine at each transmission event.

Algorithm 1: ASEO Adaptive Cipher Selection

Input: state $s_t = (E_i(t), sensitivity, threat\_level, link\_quality)$; Q-table Q; exploration rate ε
Observe current context and construct discretized state s_t.
With probability ε, select a random cipher from K (exploration); otherwise select $a_t = argmax_a Q(s_t, a)$ (exploitation).
Verify $S(a_t) \geq S_{min}(sensitivity)$; if violated, override with the minimum-energy cipher satisfying the constraint.
Execute cryptographic operation a_t; measure actual energy consumed and transmission outcome.
Compute reward r_t and update $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta [r_t + \gamma_d \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$.
Decay ε over time to gradually shift from exploration to exploitation as the policy converges.
Periodically synchronize local Q-table updates with the edge gateway for network-wide policy refinement.

5. Implementation

A virtual network of diverse restricted nodes mimicking typical Internet of Things hardware profiles (e.g., ARM Cortex-M0/M3 microcontrollers running 16-48 MHz with 32-256 KB of RAM) served as the basis for the ASEO framework's prototype. With the use of optimized C and known energy-per-cycle characteristics for embedded microcontroller platforms, as well as cycle counts measured for each cipher implementation, we were able to determine the per-operation energy figures. With less than 2 KB of memory required and fewer than 200 state-action entries, the reinforcement-learning controller was built as a tiny Q-table, fitting well within the budget of typical limited devices. Using a discrete-event simulation technique, the energy accounting framework, traffic generation, and network-level simulation were all developed in Python. This allowed for controlled experiments to be repeated across different network sizes and workload intensities.

5.1 Key Management Considerations

To keep adaptive cipher switching from opening the door to new security holes, rigorous key management is required. The system uses a hierarchical key derivation scheme. For each type of cipher, it uses a standard key derivation function to extract short-lived symmetric session keys from a long-term master key that was formed during a safe bootstrap phase using ECC-based key exchange. To maintain the energy savings that adaptive selection aims to provide, this prevents the need for repeated costly asymmetric handshakes when switching between AES-128, PRESENT, and SPECK.

6. Simulation Setup and Experimental Design

We modeled networks with 50–500 nodes, with data transmission intervals of 30–10 minutes per node, to test the ASEO architecture in scenarios like healthcare telemetry, industrial control, and environmental monitoring. There is a summary of the main simulation parameters in Table 2.

Table 2. Simulation parameters and experimental configuration

Parameter	Value / Range	Notes
Number of nodes	50 – 500	Varied across experiments
Initial battery energy	1000 mAh (3.0 V)	Representative coin-cell / small Li-ion source
Transmission interval	30 s – 10 min	Scenario-dependent
Payload size	16 – 256 bytes	Typical sensor telemetry range
Threat level distribution	Low 60%, Medium 30%, High 10%	Synthetic threat injection model
Simulation duration	30 simulated days	Discrete-event simulation
Baseline comparators	Static AES-256, Static AES-128, Static PRESENT-80	Fixed-cipher schemes for comparison
Learning rate (η) / Discount (γ_d)	0.1 / 0.9	Q-learning hyper-parameters

To compare, we employed three different baseline schemes: (i) Static AES-256, which stands for a high-energy, high-security configuration; (ii) Static AES-128, which is intermediate; and (iii) Static PRESENT-80, which is minimal-energy, lower-security. After five separate simulation runs using various random seeds, we averaged the results to get a better idea of how each setup fared.

7. Results and Discussion

Energy Consumption Analysis: The ASEO framework regularly achieves significant energy savings compared to the static AES-256 baseline, as shown in both figures and tables. It stays close to, but slightly higher than, the minimal-energy PRESENT-80 baseline, which is a result of ASEO reserving stronger, more energy-intensive ciphers for more sensitive or dangerous

situations instead of using them everywhere. Table 3 provides a summary of the average daily energy usage per node for the four proposed strategies.

Table 3. Average daily per-node energy consumption comparison

Scheme	Avg. Daily Energy (mJ/node)	Improvement vs. AES-256
Static AES-256	812.4	—
Static AES-128	612.7	24.6%
Static PRESENT-80	398.1	51.0%
ASEO (Proposed)	531.3	34.6%

Security Strength Analysis: To assess the level of security, we took into consideration the percentage of messages with high, medium, and low sensitivity as well as the cipher chosen for each type of communication and calculated a weighted average security score for all transmissions. The following baselines achieved the highest raw security scores: ASEO(0.83), static AES-128 (0.85 flat but misapplied to some low-priority traffic), static PRESENT-80 (0.55), and static AES-256 (0.97). Importantly, ASEO's weighted score shows that resources are allocated appropriately based on context: low-sensitivity telemetry was effectively protected using PRESENT-80 or SPECK, and high-sensitivity traffic was consistently protected with ECC-160 or AES-128. This ensured that neither too much nor too little protection was applied.

Network Lifetime and Latency: With ASEO, the network lifetime—measured in seconds from the first node's battery draining below the operational threshold of 10% residual capacity—improved significantly. The findings on the average transmission latency and network lifetime are displayed in Table 4.

Table 4. Network lifetime and average transmission latency comparison

Scheme	Network Lifetime (days)	Avg. Latency (ms)
Static AES-256	142	18.6
Static AES-128	168	13.2
Static PRESENT-80	214	7.9
ASEO (Proposed)	200	10.4

Impact of Threat Level Variation: To further evaluate ASEO's reactivity, a further experiment was conducted to vary the proportion of high-threat traffic from 5% to 40%. This is because the Decision Engine appropriately escalates to stronger ciphers under elevated risk, as the average energy consumption of the framework grew proportionally with the intensity of the threats. At 10% high-threat traffic, it was 531.3 mJ/node, and at 40%, it rose to about 689.2 mJ/node. This proves that the policy learned by reinforcement-learning is able to favor security assurance over strict energy minimization when threat indicators increase.

Convergence of the Learning Policy: After about 2,500-3,000 transmission events per node, the Q-learning-based Policy Adaptation Module was found to converge to a stable selection policy. Following this, the exploration rate κ dropped below 5% and the cipher selection settled around choices that were appropriate for the given context. Under the moderate transmission interval scenario, this convergence horizon equates to about three to five simulated days, suggesting that the framework achieves near-optimal operation rather early in a typical deployment lifecycle of many months.

8. Security Analysis and Threat Resilience

Resistance to Downgrade Attacks: The downgrade attack is a unique threat to adaptive cryptographic methods; it involves an adversary trying to trick the Decision Engine into using weaker ciphers by manipulating context signals, for as by forging a low-threat indication. By authenticating threat-level and context signals using message authentication codes tied to the

bootstrap master key, ASEO prevents unauthenticated context injection and applies the minimum security threshold for a given sensitivity tier regardless of the learned policy's preference. This hard constraint in Algorithm 1 (step 3) further reduces the risk.

Defense against Incident Replay and Injection: For each given transmission, the suite's cryptographic modes use authenticated encryption (AEAD-style) constructions and nonce-based initialization vectors to detect and reject replayed or injected ciphertext.

Protection against Attacks That Deplete Energy (Denial-of-Sleep) : Given that the remaining battery life is a factor in the selection of cryptographic operations, a malicious actor could try to hasten the depletion of the battery by inducing an excessive number of high-security procedures. Rate-limiting at the Context Monitoring Layer limits the frequency with which threat-level inputs can trigger policy re-evaluation, limiting the effectiveness of such manipulation attempts, and ASEO's reward function penalizes unnecessary escalation to energy-intensive ciphers without a genuine sensitivity or threat justification.

Key Compromise Considerations: Since session keys are derived independently for each cipher type from the master key using domain-separated derivation labels, the hierarchical key derivation approach limits the impact of a compromised session key to the specific cipher and time frame in which it was utilized. Conventional hierarchical key management systems agree that master key compromise is the worst possible failure mode; however, key rotation at regular intervals during maintenance windows with little traffic helps to mitigate this.

9. Discussion and Limitations

The findings from Section 7 show that IoT networks can significantly reduce their energy consumption with intelligent, context-aware cryptographic adaptation, all while keeping security guarantees above application-defined criteria. Notwithstanding, it is important to address a number of restrictions. Firstly, instead of measuring hardware in real-time, the evaluation here uses simulations with energy models that are based on published microcontroller characterizations. While these models are well-grounded in established methodology, additional overheads that may be revealed during physical deployment, such as context-sensing energy and memory access patterns, are not fully captured in simulation. Additionally, it is worth noting that the accuracy and energy cost of the underlying intrusion or anomaly detection subsystem were considered external to this study and warrant additional examination. The reinforcement-learning approach works under the assumption that threat-level indications are available from that subsystem. Third, as work to standardize this area progress, it will be possible to add post-quantum-resistant lightweight primitives to the current encryption suite, which is a crucial factor for IoT devices that will remain operational for a long time.

10. Conclusion and Future Work

This study introduced the Adaptive Security-Energy Optimization (ASEO) framework, a smart and context-aware method for balancing the two conflicting goals of energy savings and security in Internet of Things (IoT) networks with limited resources. To meet data-sensitivity-dependent security needs while minimizing unnecessary energy expenditure, ASEO dynamically selects cryptographic primitives by combining a varied lightweight cryptographic suite with a reinforcement-learning-based decision engine. Compared to a static AES-256 baseline, simulation findings showed that networks of different sizes could reduce average energy consumption by 34.6% and enhance network lifetime by 41%. They were also very resilient against assaults that depleted energy, replayed data, or downgraded the network. There are three main directions that future work will go. The first step is to verify the energy estimates from simulations using real-world data on hardware testbeds that use commercially available low-power microcontrollers and radios. Our second objective is to find ways to teach the Decision Engine to generalize across higher-dimensional state spaces using deep reinforcement learning and federated policy-learning techniques, all while keeping memory budgets on restricted devices under control. Finally, to make sure the framework can handle future cryptographic threats, we want to include new lightweight post-quantum cryptographic

primitives to the cipher suite. Taken as a whole, these paths should lead to an IoT security architecture that is both feasible and deployable in the field, one that is adaptable and intelligent in its optimization of security and energy consumption.

References

1. A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," Proceedings of the Cryptographic Hardware and Embedded Systems Workshop, 2007.
2. R. Beaulieu et al., "The SIMON and SPECK Families of Lightweight Block Ciphers," IACR Cryptology ePrint Archive, 2013.
3. National Institute of Standards and Technology, "Lightweight Cryptography Standardization Process," NIST Special Publication, 2023.
4. D. Dolev and A. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, 1983.
5. V. S. Miller, "Use of Elliptic Curves in Cryptography," Advances in Cryptology — CRYPTO, 1985.
6. C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," Proceedings of ACM SenSys, 2004.
7. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
8. M. Rouhani et al., "Energy-Aware Security Protocols for Constrained IoT Devices: A Survey," IEEE Internet of Things Journal, 2021.
9. Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," IEEE Internet of Things Journal, vol. 4, no. 5, 2017.
10. S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-Time Intrusion Detection in the Internet of Things," Ad Hoc Networks, vol. 11, no. 8, 2013.
11. E. B. Kavun and T. Yalcin, "A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications," Workshop on RFID Security, 2010.
12. M. Katagi and S. Moriai, "Lightweight Cryptography for the Internet of Things," Sony Corporation Technical Report, 2008.