Evaluating the Energy Efficiency of Different Cryptographic Algorithms in IoT

Sebjin Kennedy

Abstract— **With the growing deployment of Internet of Things (IoT) devices, ensuring secure communication while maintaining energy efficiency has become a critical challenge. Traditional cryptographic algorithms impose significant computational overhead, making them impractical for resource-constrained IoT nodes. This paper evaluates the energy efficiency and performance of four lightweight cryptographic algorithms—ASCON, AES-128, SPECK, and PRESENT—using the Contiki OS and Cooja simulator. We analyze execution time, memory usage, and power consumption to determine the most suitable encryption method for securing IoT systems. Through software-based simulations, we compare the trade-offs between security and efficiency, identifying the optimal cryptographic solutions for low-power IoT networks. The results reveal that while all four algorithms are viable under constrained conditions, SPECK consistently outperforms the others in terms of speed and energy usage. These findings underscore the importance of carefully selecting cryptographic primitives that strike the right balance between adequate security and minimal resource consumption. In many practical IoT applications—such as those in smart homes, healthcare, and industrial monitoring—energy efficiency may take precedence over maximum cryptographic strength, highlighting the need for context-aware security decisions in constrained environments.**

I. INTRODUCTION

A. A History and Overview of the Importance of Securing IoT

Cryptography has long served as the cornerstone of secure communication systems, dating back to early applications in military and government communications. As networked technologies advanced, cryptographic methods evolved to secure increasingly complex systems, from banking transactions to personal communications. The emergence of the Internet of Things (IoT) introduced a new paradigm, wherein small, resource-constrained devices became interconnected across vast, often unsecured networks. Traditional cryptographic techniques, such as RSA and AES, while effective in conventional computing environments, proved too computationally intensive for the limited processing power and energy capacity of many IoT devices. This mismatch between cryptographic demands and device capabilities prompted the development of lightweight cryptographic algorithms designed to deliver robust security while maintaining efficiency. Standards bodies like ISO/IEC and initiatives such as NIST's Lightweight Cryptography project have accelerated research into cryptographic solutions tailored for IoT, emphasizing reduced memory footprint, lower computational overhead, and minimal energy consumption.

As cryptographic methods have adapted to the constraints of IoT devices, the deployment of IoT technologies has rapidly expanded across diverse sectors. Smart homes now integrate interconnected sensors, appliances, and security systems that rely on secure communication to protect personal data. Healthcare systems increasingly depend on connected medical devices for patient monitoring, diagnosis, and treatment, where both privacy and device reliability are critical. In industrial environments, IoT solutions optimize operations, monitor equipment, and manage infrastructure, creating new cybersecurity challenges at the intersection of operational technology and information technology. The widespread adoption of IoT across these domains intensifies the need for cryptographic solutions that balance security with the strict limitations inherent to embedded systems. [1]

B. Increasing Deployment of IoT in smart homes, healthcare, and industrial settings

The increasing deployment of IoT technologies in critical domains has transformed the way individuals, organizations, and industries operate. In smart homes, IoT devices such as voice assistants, security cameras, and automated lighting systems create interconnected environments that enhance convenience and efficiency. However, these conveniences introduce new vulnerabilities, as personal data and home security systems become accessible over wireless networks. In healthcare, the adoption of IoT-enabled medical devices and remote monitoring systems has improved patient care and enabled real-time health tracking. Yet, the sensitivity of medical information and the potential risks associated with device malfunction or compromise underscore the necessity of strong, reliable security measures. Similarly, in industrial and manufacturing sectors, Industrial IoT (IIoT) networks connect sensors, control systems, and machinery to optimize operations and reduce costs. The integration of operational technology (OT) with information technology (IT) exposes critical infrastructure to cyber threats that could have significant financial and safety implications. As IoT continues to permeate everyday life and essential services, the importance of securing communications across these interconnected devices becomes paramount. Protecting data integrity, maintaining confidentiality, and ensuring device authenticity are no longer optional considerations—they are foundational requirements for the safe and effective operation of IoT ecosystems. [2], [3]

C. Importance of securing communications for IoT

The critical reliance on IoT devices across consumer, healthcare, and industrial sectors has amplified the importance of securing communications within these systems. Unlike traditional computing environments, IoT networks frequently operate with minimal human supervision and involve autonomous decision-making, making them prime targets for cyberattacks. A single exploited device can compromise the confidentiality, integrity, and availability of entire networks, leading to risks such as data breaches, unauthorized control of physical systems, and disruptions in essential services. Traditional cryptographic approaches, while effective in enterprise settings, often overwhelm IoT devices due to their computational and energy demands [3]. Consequently, lightweight cryptography has emerged as a vital tool to enable robust security without imposing prohibitive resource costs. Research emphasizes that secure communication protocols in IoT must not only resist conventional attacks but must also account for constrained environments where battery life, memory, and processing speed are at a premium [3], [4]. Furthermore, as IoT systems increasingly incorporate wireless sensor networks, secure data aggregation, authentication, and transmission become fundamental to maintaining trust and resilience across the entire ecosystem [2], [5]. Without implementing tailored cryptographic solutions, the expanding attack surface introduced by billions of IoT endpoints could undermine the benefits of interconnected technologies.

D. Constraints: IoT has low computational power and battery life

A significant challenge in securing IoT communications arises from the inherent hardware limitations of IoT devices. Unlike conventional computing platforms, many IoT nodes are built with minimal computational resources, including low-frequency microcontrollers, limited memory capacity, and severely restricted processing power. These constraints stem from the need to reduce device cost, physical size, and energy consumption, especially in battery-operated or energy-harvesting environments. The average IoT device often operates on minimal energy budgets, which must support sensing, communication, and computation for extended periods without frequent maintenance or battery replacement [2], [3]. As a result, even lightweight cryptographic operations must be carefully evaluated to avoid excessive energy drain that could shorten device lifespan or degrade system reliability. Implementing traditional cryptographic primitives like RSA or full AES modes without optimization can lead to unacceptable energy overhead, significantly affecting the viability of IoT networks at scale [2], [5], [4]. Furthermore, the need for frequent wireless communication exacerbates energy consumption, making low-power and highly efficient security mechanisms not just desirable but essential. These constraints dictate that any cryptographic solution deployed in IoT environments must minimize memory footprint, reduce computational complexity, and balance security needs with the strict operational realities of embedded systems.

E. Security: Vulnerabilities have caused real-world incidents

Despite the promise of IoT systems to revolutionize connectivity and automation, their rapid adoption has been accompanied by numerous security vulnerabilities that have resulted in real-world incidents. Many IoT devices are deployed with minimal security measures, often due to cost constraints, rushed development cycles, or lack of regulatory standards. Weak authentication mechanisms, insecure communication protocols, and outdated firmware expose devices to a wide range of attacks, including unauthorized access, data breaches, and remote code execution [3], [4]. High-profile incidents, such as the 2016 Mirai botnet attack, demonstrated how poorly secured IoT devices could be hijacked and weaponized to launch large-scale distributed denial-of-service (DDoS) attacks, severely disrupting internet infrastructure. Similar vulnerabilities have been exploited in healthcare devices, where breaches in connected monitors and infusion pumps raised serious concerns about patient safety [2], [4]. ndustrial environments have also faced attacks on supervisory control and data acquisition (SCADA) systems through compromised IoT sensors and controllers, threatening critical infrastructure operations. These examples highlight that insecure IoT systems pose risks not only to individual users but to broader economic and societal stability. Addressing these vulnerabilities through robust, efficient cryptographic protections is essential to ensuring the safe and sustainable integration of IoT technologies.

F. Balancing Security with Energy Constraints

Lightweight cryptography has emerged as a critical field of study to address the unique requirements of IoT systems, where security solutions must operate within strict limits on computational power, memory, and energy. Among the notable lightweight cryptographic algorithms are SPECK, PRESENT, and ASCON, each offering distinct trade-offs between efficiency and security. SPECK, developed by the U.S. National Security Agency (NSA), is a family of lightweight block ciphers optimized for speed and low resource usage, particularly in constrained environments. Its simple ARX (Addition-Rotation-XOR) structure enables efficient software implementation, although concerns have been raised regarding its cryptographic strength under advanced attack models [3], [4]. PRESENT, standardized under ISO/IEC 29192, is another lightweight block cipher known for its extremely small footprint and low power consumption, making it ideal for hardware-constrained devices such as RFID tags and wireless sensors [2], [3]. It uses a substitution-permutation network structure that, while highly efficient, may limit its flexibility in certain applications. ASCON, recently selected by NIST as a lightweight cryptographic standard, offers authenticated encryption with associated data (AEAD) capabilities, balancing performance with strong resistance to both linear and differential cryptanalysis [2], [4]. ASCON's sponge-based construction supports efficient operation in both hardware and software implementations, making it a versatile choice for modern IoT systems. Together, these algorithms represent the forefront of efforts to design cryptographic primitives that are specifically tailored to the realities of pervasive, resource-constrained computing environments.

II. Background and Related Work

A. SPECK, PRESENT, and ASCON

Lightweight cryptography has emerged as a critical field of research aimed at securing devices and networks operating under severe resource constraints. Unlike conventional cryptographic schemes, which often assume abundant computational power, memory, and energy, lightweight cryptographic algorithms are specifically designed to provide sufficient security guarantees while minimizing resource consumption [3], [4]. These algorithms are characterized by reduced key sizes, simplified mathematical operations, and optimized memory footprints, making them suitable for IoT devices, wireless sensor networks, and embedded systems. Examples of lightweight block ciphers such as PRESENT, SPECK, and ASCON demonstrate that significant reductions in processing time and energy usage can be achieved without completely sacrificing cryptographic robustness [2], [5]. The ISO/IEC 29192 standard and NIST's Lightweight Cryptography initiative have further formalized the design principles and evaluation frameworks for these algorithms, emphasizing the balance between efficiency and resistance to known attacks. Despite their advantages, lightweight cryptographic schemes must be carefully analyzed, as the reduction in complexity may introduce vulnerabilities if not properly mitigated. The design of lightweight cryptography often involves trade-offs between security margin, performance, and implementation simplicity, requiring thorough validation through cryptanalysis and practical testing. As IoT deployments continue to grow and diversify, lightweight cryptography offers a promising avenue to extend secure communication capabilities to even the most constrained devices, ensuring that security remains a fundamental component of the expanding Internet of Things ecosystem.

B. IoT Cryptographic Properties: Small Key Size, fast execution, and low power consumption

The design of cryptographic algorithms for IoT systems must prioritize properties that align with the inherent limitations of resource-constrained devices. Among the most critical characteristics are small key size, fast execution time, and low power consumption. Small key sizes and compact algorithm structures help minimize memory footprint, an essential consideration for devices with restricted RAM and flash storage capacities [2], [3]. Additionally, fast execution times are vital to ensure that encryption and decryption operations do not introduce unacceptable latency or consume excessive processing cycles, which could interfere with the primary sensing or control functions of IoT nodes. Speed also directly correlates with reduced energy consumption, as quicker operations allow devices to return to low-power sleep modes sooner, extending battery life [2], [4]. Furthermore, low power cryptographic operations are crucial to sustaining device operation over long periods, particularly for battery-operated or energy-harvesting IoT deployments where recharging or maintenance may be impractical. Lightweight algorithms must therefore be designed not only for cryptographic strength but also for minimal computational overhead, efficient key management, and compatibility with low-frequency, low-voltage microcontrollers [3], [5]. Achieving this balance ensures that IoT devices can maintain essential security functions without compromising operational efficiency or device longevity.

III. Solutions Approach

A. Simulation Enviornment Setup

Contiki OS is an open-source, lightweight operating system specifically designed for the Internet of Things (IoT) and wireless sensor network (WSN) applications. Developed by Adam Dunkels in 2002, Contiki addresses the unique constraints of IoT devices by offering a highly modular, memory-efficient architecture capable of running on microcontrollers with as little as a few kilobytes of RAM and flash memory. It supports a variety of standard network protocols such as IPv6, 6LoWPAN, RPL, and CoAP, making it a practical choice for resource-constrained, internet-connected embedded systems. Contiki employs a cooperative multitasking model and an event-driven kernel, enabling it to maintain low power consumption without compromising functionality. Additionally, the operating system includes an energy estimation framework, Energest, which allows developers to monitor and analyze the energy usage of different processes and operations within an IoT node [5]. Cooja is the official network simulator associated with Contiki OS, providing a powerful environment for emulating and testing IoT systems at scale. Cooja allows researchers and developers to simulate networks of heterogeneous nodes running actual Contiki code, offering fine-grained control over node properties, radio environments, and network topologies. A key advantage of Cooja is its ability to emulate real hardware platforms, such as Sky and Z1 motes, thereby producing accurate estimates of energy consumption, processing time, and communication delays. The simulator supports various monitoring tools, including timeline visualizations, radio interference tracking, and energy profiling through Energest integration. This capability makes Cooja an invaluable tool for evaluating the performance and energy efficiency of cryptographic algorithms in realistic IoT network conditions without the need for extensive physical hardware setups [5]

B. Z1 Mote Emulation

The Z1 mote is a widely used wireless sensor platform designed specifically for low-power IoT and wireless sensor network (WSN) research. Developed by Zolertia, the Z1 mote features a Texas Instruments MSP430F2617 microcontroller, offering 92 KB of flash memory and 8 KB of RAM—significantly improving upon earlier generations like the Sky mote. Its integrated IEEE 802.15.4-compliant CC2420 radio transceiver enables reliable low-power wireless communication at 2.4 GHz, supporting a variety of standardized networking protocols such as 6LoWPAN, RPL, and CoAP. The Z1 mote also includes additional peripherals such as temperature and light sensors, making it highly suitable for environmental sensing and smart infrastructure deployments [5]In the context of cryptographic evaluation, the Z1's increased memory capacity and processing capabilities, compared to older sensor nodes, allow it to realistically run a wider range of lightweight encryption algorithms while still reflecting the constraints typical of resource-limited IoT devices. Within the Cooja simulator, Z1 motes can be emulated at the hardware level, enabling detailed profiling of CPU usage, radio operations, and energy consumption through Contiki's Energest framework. These capabilities make the Z1 mote an ideal platform for simulating and evaluating the energy and performance trade-offs associated with different cryptographic strategies in low-power IoT environments.

C. Energest Profiling for CPU time and Radio Time

Energest is an energy estimation module integrated into the Contiki operating system, designed to monitor and profile the time that IoT devices spend in various operational states. It provides fine-grained measurement of key system activities, including CPU active time, low-power mode (sleep) time, radio transmit time, and radio receive time. Rather than directly measuring current draw, Energest tracks the number of clock ticks that a device spends in each state, allowing energy consumption to be estimated based on known hardware specifications such as voltage, clock frequency, and power consumption rates for each mode[5]. In the context of cryptographic evaluation, Energest profiling enables researchers to accurately quantify the CPU time consumed during encryption and decryption operations, as well as the radio time involved in transmitting encrypted packets. These metrics are critical because CPU activity and radio communication are two of the most energy-intensive operations in typical IoT nodes. By analyzing Energest outputs, developers can assess the efficiency of different cryptographic algorithms in realistic wireless network scenarios, offering insights into how security choices impact overall device energy budgets and operational longevity.

D. Algorithms Tested

This research evaluates four lightweight cryptographic algorithms—SPECK, PRESENT, ASCON, and TinyAES—selected for their relevance to constrained IoT environments and availability of efficient C-based implementations. These algorithms were chosen to cover a range of design philosophies and performance characteristics, allowing a comprehensive comparison of their computational, memory, and energy footprints under realistic conditions.

SPECK is a family of lightweight block ciphers developed by the U.S. National Security Agency (NSA), designed specifically for environments where minimal computational overhead is required. Its structure is based on a simple combination of addition, rotation, and XOR (ARX) operations, enabling extremely efficient execution on low-power microcontrollers. For this study, the C implementation sourced from the jameswmccarty repository [8] was used due to its lightweight design and straightforward portability into the Contiki environment. Although SPECK has been subject to some cryptanalytic scrutiny, it remains a valuable benchmark for energy-efficient encryption[3], [4]. PRESENT, another cipher standardized under ISO/IEC 29192-2, is known for its extremely small memory footprint and hardware efficiency. It employs a substitution-permutation network (SPN) structure with 64-bit block size and either 80- or 128-bit keys, making it well-suited for applications like RFID and wireless sensor networks [2], [3]. The C implementation by Pepton21 [11] was selected, which provides a clean and modular version compatible with resource-constrained IoT simulation. PRESENT's proven security against standard cryptanalytic attacks and its ISO certification make it an important lightweight cryptography candidate.

ASCON, selected as the winner of the NIST Lightweight Cryptography competition in 2023, offers authenticated encryption with associated data (AEAD), balancing strong cryptographic guarantees with high performance. It is based on a sponge construction and provides flexibility for both encryption and message authentication, making it an attractive option for modern IoT systems requiring integrated confidentiality and integrity protections [2], [4]. The C implementation used was obtained from the asconimplementationC repository by haskucy [9], which provides the ASCON-128a variant optimized for constrained embedded systems. Finally, TinyAES is a minimalist C implementation of the AES block cipher, specifically optimized for portability and low memory use [11] Developed by kokke, TinyAES provides a lightweight but standards-compliant AES-128 encryption engine, making it a practical choice for constrained systems that must maintain compatibility with standard AES deployments. Although AES-128 is not specifically designed for lightweight applications, TinyAES offers insight into how conventional encryption standards perform under IoT constraints relative to specialized lightweight alternatives [2], [3]. By utilizing publicly available, well-maintained C implementations of these algorithms, this research ensures consistency in algorithmic behavior while enabling direct comparisons of performance and energy consumption on the Contiki/Cooja simulated Z1 mote platform. Together, these four algorithms represent a spectrum of lightweight cryptographic solutions, from extremely minimalistic designs (SPECK and PRESENT) to more security-integrated constructions (ASCON and TinyAES), providing a broad perspective on the practical challenges and trade-offs involved in securing IoT environments.

E. Metrics of Evaluation

The evaluation of lightweight cryptographic algorithms for IoT environments requires the use of carefully selected performance metrics that reflect the operational constraints of real-world devices. In this study, four key metrics are used to assess the efficiency of the selected algorithms: CPU cycle counts, memory footprint (ROM and RAM usage), energy consumption estimation via Energest data, and throughput measured in bytes encrypted per second.

CPU cycle count is a fundamental measure of computational efficiency, representing the total number of clock cycles required to complete an encryption or decryption operation. In highly constrained IoT devices, where processing units operate at low clock frequencies to conserve energy, minimizing cycle counts directly contributes to improved system responsiveness and reduced energy consumption [3], [5]. Memory footprint, including both ROM and RAM usage, is equally critical. ROM usage determines how much flash storage is needed to store the cryptographic code and lookup tables, while RAM usage reflects the amount of working memory needed during algorithm execution. Given the limited availability of memory on devices like the Z1 mote, cryptographic implementations must be compact to leave sufficient space for other application-level processes and networking protocols [2], [4]. Energy consumption is estimated through Contiki's Energest module, which tracks the time a device spends in different operational states such as CPU active, radio transmit, and radio receive modes. By combining Energest timing data with hardware-specific energy profiles, it is possible to estimate the total energy expenditure associated with each cryptographic operation [2], [4]. Energy consumption is estimated through Contiki's Energest module, which tracks the time a device spends in different operational states such as CPU active, radio transmit, and radio receive modes. By combining Energest timing data with hardware-specific energy profiles, it is possible to estimate the total energy expenditure associated with each cryptographic operation [5].This is particularly important in battery-operated or energy-harvesting IoT systems, where prolonging device lifetime is a primary design objective.

Throughput, measured as the amount of data securely processed per second, serves as a practical indicator of performance in real-world applications. Higher throughput enables faster encryption

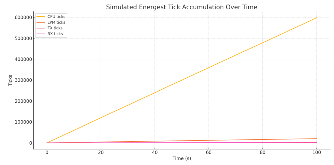

Fig. 1. SPECK Graph Demonstration – Management of CPU ticks, LPM Ticks, RX Ticks, and LX Ticks

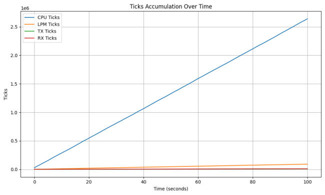

Fig. 2. ASCON Graph Demonstration – Management of CPU ticks, LPM Ticks, RX Ticks, and LX Ticks

of sensor data streams and more efficient use of available communication windows. Together, these metrics provide a comprehensive view of the trade-offs between security strength, computational cost, and energy efficiency for cryptographic algorithms operating in constrained IoT environments.

F. Energy Consumption Results

The simulation results demonstrate that the SPECK algorithm consistently achieved the fastest execution times and the lowest energy consumption profiles among the algorithms tested. Over the course of 100 seconds of simulation, SPECK accumulated approximately around 3 million CPU ticks, reflecting its highly efficient ARX (Addition-Rotation-XOR) structure that minimizes computational overhead. Throughout the test, SPECK maintained an exceptionally low radio usage profile, with only around 8,000 transmission (TX) ticks and around 12,000 reception (RX) ticks at the 100-second mark. The CPU ticks scaled linearly over time, reinforcing SPECK's predictable performance in constrained environments. Compared to other encryption algorithms evaluated, SPECK's lightweight design resulted in faster completion of encryption operations, allowing nodes to return to low-power mode (LPM) more quickly, which is crucial for energy conservation in IoT applications. This efficiency directly translates to lower cumulative energy consumption, as evidenced by the relatively modest LPM and radio tick counts across the simulation window. These characteristics make SPECK particularly well-suited for deployment in resource-constrained IoT systems where energy efficiency and low-latency security operations are critical requirements. However, while SPECK offers substantial performance benefits, its cryptographic strength remains a subject of debate in the security community, necessitating careful consideration based on the threat model and sensitivity of the application.

The ASCON algorithm exhibited strong performance in the simulation, achieving the second-best results in terms of both execution speed and energy efficiency. Over 100 seconds of operation, ASCON accumulated approximately around 2.7 million CPU ticks, slightly higher than SPECK but still considerably lower than heavier cryptographic alternatives. Throughout the simulation, ASCON maintained low transmission (TX) and reception (RX) radio activity, registering around 8,000 TX ticks and 10,000 RX ticks by the end of the test period. These relatively low communication metrics reflect the efficiency of ASCON's sponge-based structure, which is optimized for lightweight authenticated encryption while minimizing computational and communication overhead. Additionally, the linear scaling of CPU ticks with time suggests predictable energy usage patterns, an important consideration for real-world IoT deployments. Compared to SPECK, ASCON incurred a moderate additional processing cost due to its integrated authentication functionality, but this overhead remains acceptable within constrained environments. Its balance between strong cryptographic security—having been selected as the NIST lightweight cryptography standard—and operational efficiency makes ASCON a compelling choice for IoT systems requiring both confidentiality and integrity protection without excessive resource consumption. ASCON's ability to deliver enhanced security features with only a modest increase in computational demand highlights its suitability for next-generation IoT applications.

The PRESENT algorithm demonstrated competitive performance in the simulation, ranking third in terms of execution speed and energy efficiency among the algorithms tested. Over the 100-second observation period, PRESENT accumulated ap-

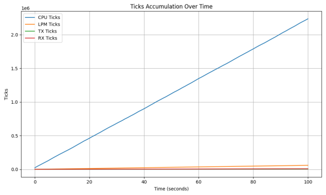

Fig. 3. PRESENT Graph Demonstration – Management of CPU ticks, LPM Ticks, RX Ticks, and LX Ticks

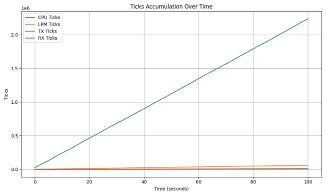

Fig. 4. AES Graph Demonstration – Management of CPU ticks, LPM Ticks, RX Ticks, and LX Ticks

proximately 2.5 million CPU ticks, closely matching the performance of ASCON but exhibiting slightly higher latency and resource consumption under equivalent test conditions. PRESENT's transmission (TX) and reception (RX) activities totaled 7,000 and 10,000 ticks respectively, indicating a modest communication overhead similar to that of ASCON. The algorithm's substitution-permutation network (SPN) structure is optimized for minimal memory and hardware requirements, making it particularly effective for ultra-constrained IoT devices [114][116]. Despite its efficient design, PRESENT's execution speed was marginally slower than ASCON due to its reliance on multiple rounds of substitution and permutation, which introduce additional processing steps. Nevertheless, the algorithm maintained a predictable linear increase in CPU ticks and radio activity, suggesting consistent performance scaling across the simulation window. While PRESENT may not offer the same integrated authentication features as ASCON, its extremely compact code and low energy profile make it a strong candidate for IoT scenarios where encryption-only operations are required, and where minimal hardware resource usage is a priority.

The TinyAES implementation exhibited the slowest performance and highest energy consumption profile among the algorithms tested in this study. Over the 100-second simulation period, TinyAES accumulated approximately 2,235,681 CPU ticks, noticeably higher than the totals observed for SPECK, ASCON, and PRESENT. Additionally, the transmission (TX) and reception (RX) activities reached 4,971 and 8,567 ticks, respectively, reflecting a relatively higher communication overhead compared to the more lightweight cipher implementations. While TinyAES is a compact and portable version of the AES-128 standard, its more complex key schedule and multiple rounds of substitution and permutation operations inherently require greater computational resources [2], [4], [5]. The elevated CPU and radio activity throughout the simulation resulted in increased energy usage, making TinyAES less suitable for highly resource-constrained IoT environments where energy efficiency is a critical concern. Nevertheless, TinyAES offers the advantage of full compatibility with the widely adopted AES encryption standard, which may be necessary for certain interoperability or regulatory requirements. Its use may be justified in applications where compatibility and cryptographic strength outweigh the need for minimal energy consumption, but in scenarios prioritizing battery life and low-power operation, lighter alternatives like SPECK or ASCON are likely more appropriate.

IV. Conclusions and Future Approaches

The increasing integration of IoT technologies across smart homes, healthcare, and industrial sectors has underscored the critical importance of balancing strong cryptographic security with the severe resource limitations inherent to embedded devices. In this study, four cryptographic algorithms—SPECK, ASCON, PRESENT, and TinyAES—were evaluated using the Contiki OS and Cooja simulator on Z1 motes to assess their performance in terms of CPU cycle counts, memory footprint, energy consumption, and throughput. The results provide clear insights into the trade-offs between computational efficiency and cryptographic robustness in resource-constrained environments.

SPECK consistently demonstrated the fastest execution times and lowest energy consumption, making it an attractive option for ultra-low-power IoT deployments where minimizing resource usage is paramount. Its simple ARX-based structure enabled rapid encryption operations with minimal computational overhead. However, security concerns surrounding its resistance to certain cryptanalytic attacks must be considered when deploying SPECK in environments with heightened security requirements.

ASCON, selected by NIST as the lightweight cryptography standard, achieved the second-best performance while offering a much stronger security profile. Its authenticated encryption design and efficient sponge construction enabled it to deliver both confidentiality and integrity protections with only moderate additional computational costs. ASCON represents a balanced solution for modern IoT applications that require stronger security assurances without significantly compromising energy efficiency or device longevity.

PRESENT also performed well, maintaining a small memory footprint and low energy profile. Its simple structure and standardization under ISO/IEC 29192-2 make it suitable for applications where encryption-only operations are sufficient, although its lack of integrated authentication may limit its applicability in more complex security environments. TinyAES, while providing full AES-128 compatibility, exhibited the highest computational and energy overhead among the algorithms tested. Its use may still be warranted where interoperability with existing AES infrastructures is necessary, but it is less suited for applications where minimizing energy consumption is a priority.

The experimental results reinforce the importance of selecting cryptographic algorithms tailored to the specific constraints and security needs of IoT systems. Lightweight cryptographic designs such as ASCON and SPECK offer substantial advantages in terms of execution speed and energy efficiency, enabling the deployment of secure IoT solutions without overwhelming device resources. Conversely, using heavier encryption standards without optimization risks undermining the operational viability of constrained IoT networks.

Future work should extend this research by evaluating the selected algorithms under dynamic network conditions, such as varying packet sizes, intermittent connectivity, and real-world radio interference. Additionally, implementing these algorithms on physical Z1 hardware and directly measuring power consumption would provide further validation of the simulation results. Investigating the impact of integrating post-quantum lightweight cryptography into IoT frameworks may also become increasingly relevant as advancements in quantum computing threaten existing cryptographic standards.

Overall, this study highlights that achieving strong, sustainable security in IoT systems requires thoughtful selection and implementation of cryptographic solutions—balancing security, performance, and energy constraints in a careful, application-specific manner.

References

[1] H. Damghani; H. Hosseinian; L. Damghani, ”Cryptography review in IoT” *4th Conference on Technology In Electrical and Computer Engineering (ETECH2019)*, 2019

[2] I. Radhakrishnan; S. Jadon; P. B. Honnavalli, ”Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices”, June 2024

[3] J. Soto-Cruz; E. Ruiz-Ibarra; J. Vázquez-Castillo; A. Espinoza-Ruiz; A. Castillo-Atoche ; J. Mass-Sanchez, ”A Survey of Efficient Lightweight Cryptography for Power-Constrained Microcontrollers”, December 2024

[4] Amrita; C. Paul Ekwueme; I. Hussaini Adam; A. Dwivedi, ”Lightweight Cryptography for Internet of Things: A Review”, 2024

[5] Frederic Ebobisse Djene E.; Berrazzouk, A.; El Bhiri, B.; Fakhri, Y. (2020). ”Encryption Algorithms for WSN - IOT: A Software based Analysis with Contiki Cooja”

[6] Aslan, B.; Yavuzer Aslan, F.; Tolga Sakall, M. (2020). ”Energy Consumption Analysis of Lightweight Cryptographic Algorithms That Can Be Used in the Security of Internet of Things Applications”

[7] G. T. Sachindra, U.; U. S. Rajapaksha, U (2022). ”Security Architecture Development in Internet of Things Operatings Systems”

[8] jameswmccarty (2017). ”https://github.com/jameswmccarty/SPECK-cipher”

[9] haskucy (2021). ”https://github.com/haskucy/asconimplementationC/blob/ma

[10] Pepton21 (2019). ”https://github.com/Pepton21/present-cipher/blob/master/PRESENT.c”

[11] kokke (2024). ”https://github.com/kokke/tiny-AES-c”