مجلة بحوث جامعة تعز Taiz University Research Journal العدد (49) Issue (49) TURJ ISSN: 2958-7484 Vol :49.pp.279-300 سبتمبر 2026م September 2026

مواءمة أنظمة الأمان الرقمي مع السلامة البيئية: تحليل أداء متعدد الأبعاد للأطر التشفيرية المستدامة ومقياس النقاط الخضراء

Harmonizing Digital Fortresses with Ecological Integrity: A Multi-Dimensional Performance Analysis of Sustainable Cryptographic Frameworks and the Green Score Metric

Raad Abdo M. Al Selwi1,2, Niyazi A. S. Al-Areqi1,3, Maram Abdualkhaleq Saif Assag 4, Amatalrahman Ahmed Mohammed 4

1 Department of Information Technology, Al-Saeed Faculty of Engineering & IT, Taiz University, Taiz, Yemen. **raad.alselwi@taiz.edu.ye**

2Department of Cyber Security, Faculty of Engineering & IT, Al-Saeed University, Taiz, Yemen. **niyazi75.alareqi@taiz.edu.ye**

3Al-Saeed Center for Scientific Research, Al-Saeed University, Taiz, Yemen. **eng.maramassag@gmail.com**

4Department of Software Engineering, Al-Saeed Faculty of Engineering & IT, Taiz University, Taiz, Yemen. **amatamed90@gmail.com**

تاريخ استلام البحث: ٢٠٢٦/٦/١٨ تاريخ قبول النشر: ٢٠٢٦/٧/١٩

الملخص:

مع تزايد الاعتماد العالمي على تكنولوجيا المعلومات والاتصالات، برز استهلاك الطاقة المرتبط ببروتوكولات التشفير كعائق كبير أمام تحقيق أهداف التنمية المستدامة، لا سيما فيما يتعلق بتوفير الطاقة بأسعار معقولة (الهدف ٧) وبناء بنية تحتية مرنة (الهدف ٩). تقدم هذه الورقة البحثية تقييمًا متعدد الجوانب لأداء خوارزميات التشفير ودوال التجزئة القياسية، بما في ذلك خوارزميات AES و RSA و ECC و SHA، مع دراسة استدامتها من خلال مجموعة أدوات قياس أداء مصممة خصيصًا. تضمنت المنهجية قياس ثلاثة مؤشرات رئيسية: وقت التنفيذ، واستخدام الذاكرة، وحمل وحدة المعالجة المركزية، والذي يُعد مؤشرًا مباشرًا لاستهلاك الطاقة. تكشف النتائج الرئيسية عن تباين واضح في كفاءة استخدام الموارد: فقد برزت خوارزميات التشفير المتناظرة، ولا سيما AES المُسرّعة بواسطة الأجهزة، كأكثرها كفاءة، حيث حافظت على حمل لوحدة المعالجة المركزية بنسبة ٢٦.٤٪، مقارنةً بنسبة ١٠٠٪ التي تتطلبها العمليات غير المتناظرة. بالإضافة إلى ذلك، وُجد أن تشفير المنحنى الإهليلجي (ECC) يُحقق انخفاضًا في استهلاك الطاقة لكل بت بمقدار ١٤.٦ ضعفًا مقارنةً بخوارزمية RSA، مما يجعله الخيار الأمثل والمستدام لأمن المفتاح العام. في اختبارات سلامة البيانات، أظهرت خوارزمية SHA-256 أداءً متفوقًا على خوارزمية SHA3-256 ضمن بيئات التشغيل القياسية. تم وضع "مؤشر أخضر" جديد وموزون لدمج هذه المقاييس متعددة الأبعاد في مؤشر واحد قابل للتنفيذ، مما يسمح بتقييم الأداء وفقًا لملفات تعريف تشغيلية مختلفة (مثل: متوازن، حرج لوحدة المعالجة المركزية، مقيد بالذاكرة). تؤكد هذه النتائج أن اختيار الخوارزمية المناسبة يُمكن أن يُسهّل تحقيق أمن رقمي قوي مع الحفاظ على سلامة النظام البيئي.

الكلمات المفتاحية: خوارزميات التشفير، خوارزميات التجزئة، التشفير الأخضر، تشفير المنحنى الإهليلجي ECC، وحدة المعالجة المركزية.

ABSTRACT:

As the global dependence on Information and Communication Technology (ICT) intensifies, the energy consumption associated with cryptographic protocols has emerged as a substantial impediment to realizing Sustainable Development Goals (SDGs), particularly in relation to affordable energy (SDG 7) and resilient infrastructure (SDG 9). This paper offers a multi-faceted performance assessment of standard cryptographic and hashing algorithms—including AES, RSA, ECC, and SHA variants—scrutinizing their sustainability through a bespoke benchmarking toolkit. The methodology involved quantifying three critical metrics: execution time, memory usage, and CPU load, the last of which served as a direct proxy for energy consumption. Pivotal findings reveal a pronounced disparity in resource efficiency: symmetric ciphers, notably hardware-accelerated AES, have emerged as the most efficient, maintaining a CPU load of 26.4% in contrast to the 100% utilization mandated by asymmetric operations. Additionally, Elliptic Curve Cryptography (ECC) has been found to yield a 14.6-fold reduction in energy consumption per bit relative to RSA, establishing it as the optimal sustainable option for public-key security. In tests of data integrity, SHA-256 exhibited superior performance compared to SHA3-256 within standard operational environments. A novel, weighted "Green Score" was formulated to synthesize these multi-dimensional metrics into a single, actionable score, allowing for performance evaluation against different operational profiles (e.g., Balanced, CPU-Critical, Memory-Constrained). These insights affirm that judicious algorithm selection can facilitate robust digital security while preserving ecological integrity.

Keywords: Cryptographic algorithms, Hashing algorithms, Green Cryptography, ECC, CPU.

1. INTRODUCTION

In the contemporary epoch characterized by accelerated digital transformation, the domains of cybersecurity and environmental sustainability have amalgamated into a unified engineering quandary.[1] Conventional assessments of cryptographic systems predominantly emphasize mathematical robustness, frequently neglecting the persistent energy overhead engendered by encryption and authentication protocols, which substantially exacerbates the carbon footprint of global data centers.[2] This predicament is particularly pronounced for resource-constrained Internet of Things (IoT) devices, wherein computationally intensive algorithms can swiftly deplete battery reserves.

The extant literature pertaining to "Green Cryptography" delineates it as an essential strategy for achieving sustainable security through the incorporation of environmentally mindful practices into cryptographic methodologies. Research conducted by Adi Bima Setiawan (2024) underscores the necessity of cryptography in formulating sustainable information security systems that alleviate threats while conforming to principles of sustainability.[3] A pivotal element of this framework is the confluence of artificial intelligence (AI) and green computing, which augments cybersecurity through enhanced threat detection, concurrently minimizing energy consumption.[2] Kumar et al. (2025) advocate for a holistic strategy that integrates sustainable technologies with robust security protocols to safeguard digital assets while concurrently mitigating environmental degradation.[2]

Moreover, sustainable design frequently encompasses the "recycling" of cryptographic components. Troutman and Rijmen (2009) propose an austere philosophy that repurposes primitives to devise simpler, more secure implementations, such as employing AES for both encryption and authentication.[4] In order to assess these initiatives quantitatively, energy profiling has emerged as a fundamental element of contemporary research. Muruganandham (2025) introduced the "Environmental Cost Index" (ECI) to appraise algorithm sustainability, elucidating that increased RSA key sizes precipitate an exponential escalation in power consumption and carbon emissions.[5] Beyond the realm of computational intensity, Buchmann
(2025) advocates for a sustainability perspective rooted in temporal considerations, asserting that cryptographic resilience must remain secure over extended durations despite potential fluctuations in mathematical complexity assumptions.[6]

Within the sphere of ubiquitous computing, the imperative for lightweight alternatives is critical. Sarker (2025) [7] and Zinabu et al. (2025) [8] underscore that traditional schemes may prove impractical for battery-restricted devices, thereby catalyzing the evolution of lightweight block ciphers such as SIMON, SPECK, and PRESENT.[9] As we advance toward a post-quantum paradigm, Gadiyar et al. (2025) [10] and Halak et al. (2024) [11] accentuate the necessity for the deployment of lightweight post-quantum signatures and key encapsulation mechanisms (KEMs) that are expressly optimized for "Green IoE" to avert a surge in ICT-related emissions.[8] This investigation seeks to address these challenges by quantifying the resource expenditures associated with standard primitives, thereby providing a data-driven trajectory toward sustainable security frameworks.

2. METHODOLOGY AND ANALYSIS

This inquiry employs a quantitative experimental design actualized via an interactive benchmarking toolkit developed in Python and leveraging the Streamlit framework. The toolkit meticulously isolates computational behavior across three essential dimensions to furnish a comprehensive perspective on cryptographic sustainability.

Table 1 Basic algorithms used in encryption

Algorithm	Category	Type	Key/Digest Size	Strength	Rationale & Use Case Example
AES	Encryption	Symmetric	128-bit	High	The global standard for bulk data encryption (e.g., SSDs, TLS).
DES	Encryption	Symmetric	56-bit	Obsolete	Included as a legacy baseline for historical performance comparison.
RSA	Encryption	Asymmetric	2048-bit	High	Widely used for key exchange and digital signatures, known for high cost.
ECC	Digital Signature	Asymmetric	256-bit	Very High	A modern, efficient alternative to RSA, ideal for mobile/IoT devices.
SHA-256	Hashing	Hash	256-bit	High	The most widely used hash function on the internet (e.g., TLS, Bitcoin).
SHA3- 256	Hashing	Hash	256-bit	High	The modern NIST standard, designed with a different internal structure.

2.1 Algorithm Selection and Data Preparation

The investigation scrutinizes a meticulously selected array of cryptographic primitives: AES and DES (symmetric), RSA and ECC (asymmetric), alongside SHA-256 in comparison to SHA3-256 (hashing). As delineated in **Table 1**, these selections embody a synthesis of contemporary standards and historical benchmarks.

The selected algorithms represent the most widely used cryptographic techniques across the three principal categories of modern cryptography: symmetric encryption (AES and DES), asymmetric encryption (RSA and ECC), and cryptographic hash functions (SHA-2 and SHA-3). [12] This selection ensures broad coverage of contemporary cryptographic methods while maintaining a manageable number of experiments for comprehensive analysis. The excluded algorithms were omitted because they either represent alternative key lengths or output sizes within the same cryptographic family, are no longer recommended for secure applications, or would substantially increase the experimental workload without providing significant additional methodological value. DES was included exclusively for comparative research purposes as a representative of legacy symmetric encryption algorithms. Its inclusion is intended to facilitate the analysis of how advances in cryptographic algorithm design have influenced computational resource consumption and environmental efficiency, rather than to recommend its use in modern secure applications.

All benchmarks were conducted using the interactive Streamlit application on a standardized consumer- grade system to ensure real-world applicability. For the analysis presented in this chapter, the Medium (approx. 500 KB) dataset was used as the primary input. This size was chosen as it is substantial enough to produce meaningful and stable measurements while representing a common workload, such as processing application logs, documents, or medium-sized data transfers. The zlib compression library was employed to reduce the size of the plaintext before encryption. Compressing the input data decreases the amount of information processed by the cryptographic algorithms, which can reduce execution time, CPU

utilization, and memory usage. [13] From a green software engineering perspective, processing fewer data may also reduce computational workload and indirectly improve energy efficiency

. Furthermore, zlib was selected because it is an open-source, mature, and widely adopted compression library based on the Deflate algorithm, offering an effective balanced between compression ratio and processing speed.

2.2 MEASUREMENT INSTRUMENTATION

Performance metrics are acquired utilizing a multi-tiered monitoring framework as expounded in **Table 2**:

Table 2 Tools and Libraries used in encryption

Tool/Library	Purpose
Streamlit	To build the interactive web-based user interface and dashboard.
Python 3.x	The core programming language for implementation.
pycryptodome	A comprehensive library providing AES, DES, RSA, SHA-256, and SHA3-256.
Ecdsa	A dedicated library for the Elliptic Curve Digital Signature Algorithm.
Pandas	For data manipulation, structuring results, and analysis.
Matplotlib	For generating the static performance charts and plots.
Psutil	For cross-platform monitoring of CPU usage.
Tracemalloc	A standard Python library for accurately tracking memory allocation.
Zlib	A standard Python library for data compression and decompression.

- **Temporal Performance**: Assessed through high-resolution timers (time.time()) to accurately capture the execution duration for encryption, signing, and hashing processes.
- **Memory Pressure**: Monitored via the tracemalloc library, enabling the identification of peak resident set size (RSS) throughout operational activities.
- **Energy Proxy**: Average CPU utilization is observed through the psutil library, functioning as a credible proxy for physical power consumption and thermal emissions. .[14]

2.3 THE GREEN SCORE FORMULATION

To amalgamate these multi-faceted metrics, we propose the "Green Score" (GS), a dimensionless metric $(0-1)$ that signifies an algorithm's relative environmental efficiency.[15] The calculation of the score is articulated as follows:

$$GS = 1 - \sum_{i \in \{t,m,c\}} (\omega_i . N_i)$$

where i Index representing each evaluation metric, N_i Normalized value of metric i, scaled between 0 and 1, N_t, N_m, N_c represent normalized metrics for time, memory, and CPU utilization, and ω_i denotes user-defined weighting factors. For instance, a "CPU-Critical Profile" allocates $\omega_c = 0.60$ to emphasize energy conservation within data center contexts.

Assume that three performance indicators are considered: Execution time (T), Memory usage (M), CPU utilization (C), since these metrics have different units (seconds, megabytes, and percentage of CPU utilization), they cannot be combined directly. Each metric is transformed into a dimensionless value ranging from 0 to 1 using a normalization technique such as Min-Max normalization:

$$N_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

This process ensure that all metrics are comparable on the same scale.

The proposed GS is used to evaluate the overall efficiency of an algorithm by combining multiple performance metrics into a single normalized score, Specifically, the formula integrates three key computational metrics: execution time, memory consumption, and CPU utilization. Since these metrics are measured in different units, each metric is first normalized to a common scale ranging from 0 to 1. This normalization ensures that no single metric dominates the evaluation due to differences in measurement units.

The normalized metrics are then multiplied by their corresponding weights (ω_i), which represent the relative importance of each metric according to the evaluation objectives. The weighted values are summed to produce an overall performance cost:

$$D = \sum_{i \in \{t,m,c\}} (\omega_i . N_i)$$

Finally, the GS is calculated as:

$$GS = 1 - D = 1 - \sum_{i \in \{t,m,c\}} (\omega_i . N_i)$$

Subtracting the weighted cost from one ensures that higher GS values correspond to better overall performance. Consequently, the GS ranges from 0 to 1, where values closer to 1 indicate higher computational efficiency characterized by lower execution time, lower memory consumption, and lower CPU utilization.

This formulation is particularly suitable for efficiency evaluation because it provides a balanced assessment rather than relying on a single performance metric. In practice, an algorithm with the shortest execution time may consume excessive memory or require high

CPU utilization, whereas another algorithm may offer a better trade-off among all three resources. By integrating these metrics into a weighted composite index, the GS captures the overall computational efficiency and facilitates objective comparisons between different algorithms.

Moreover, the weighting mechanism provides flexibility, allowing researchers to prioritize specific metrics depending on the application. For example, real-time systems may assign a higher weight to execution time, while resource-constrained environments may emphasize memory consumption or CPU utilization. Therefore, the proposed GS serves as a comprehensive, scalable, and adaptable performance evaluation metric for comparing computational algorithms.

Since direct energy measurement were not available, CPU utilization was adopted as a proxy metric for energy consumption. This choice is justified because higher CPU utilization is generally associated with increased processor power usage, particularly when all experiments are performed on the same hardware platform under identical operating conditions. Therefore, CPU utilization provides a practical and consistent basis for Comparing the relative environmental efficiency of the evaluated cryptographic algorithms.

In the absence of a universally accepted weighting scheme for evaluating the environmental efficiency of cryptographic algorithms, equal weights were assigned to execution time, memory consumption, and CPU utilization. This approach ensures that each metric contributes equally to the GS, thereby avoiding subjective bias and providing a balanced assessment of computational efficiency.

3. RESULTS AND DISCUSSION

Empirical evaluations performed on a standardized consumer-grade system (Intel Core i7-10750H, 16 GB RAM) uncover noteworthy disparities in the resource efficiency of secure systems.

3.1 PERFORMANCE HIERARCHY IN ENCRYPTION AND SIGNING

The performance metrics of confidentiality and authenticity algorithms (summarized in Table 3 and illustrated in Figure 1) highlight a significant dichotomy between symmetric and asymmetric frameworks.

Symmetric algorithms exhibited a superior efficiency profile. Hardware-accelerated AES accomplished encryption tasks in approximately 7.996 ms while maintaining a moderate CPU load of 26.4%. This level of efficiency is ascribed to its substitution-permutation network and the employment of the AES-NI instruction set, which reduces per-packet overhead by up to 60%. Although DES displayed reduced memory consumption, its obsolescence renders it an inadequate choice for contemporary security requirements despite its rapid processing capabilities.

Table 3: Empirical Performance Metrics for Encryption/Signing (500 KB Dataset)

Algorithm	Time (s)	Memory (KB)	CPU (%)	Green Score
AES	0.007996	156.4199	26.4	0.402191
DES	0.006894	6.1152	23.5	0.822649
RSA	0.004898	3.5586	100.0	0.662492
ECC (Signing)	0.009024	0.0000	100.0	0.330000

Note: Data derived from the medium data volume scenario.

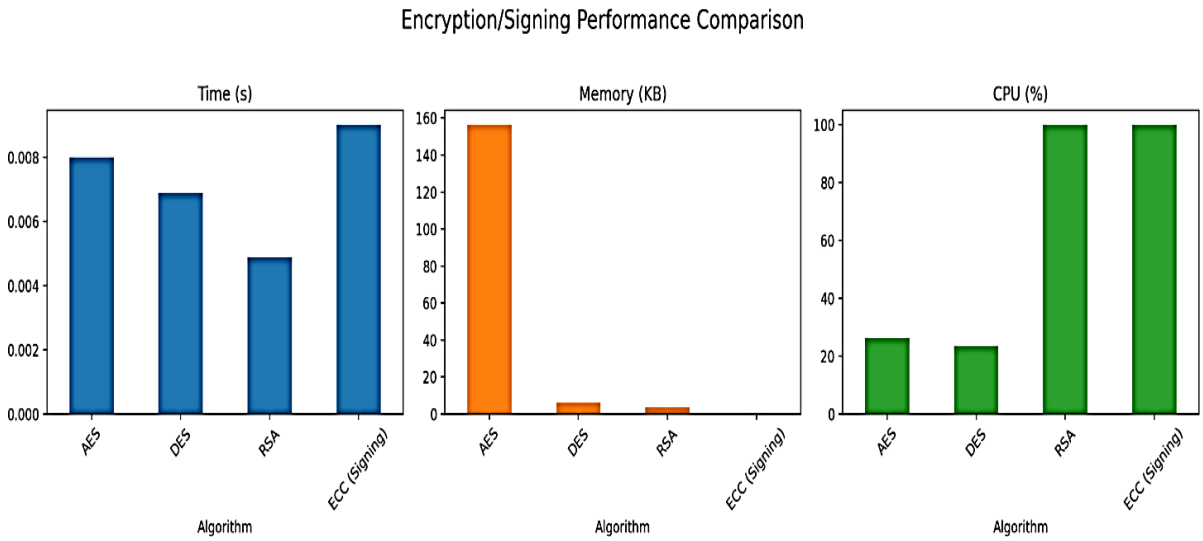

Figure 1 Comparing the performance metrics of encryption algorithms

3.2 THE ASYMMETRIC DIVIDE: ECC VS. RSA

The findings affirm that asymmetric operations are typified by maximal resource utilization, achieving a 100% CPU load as depicted in Figure 1. Nonetheless, ECC (signing) delivers comparable security to RSA with significantly smaller key sizes (256−bit vs 2048−bit). Recent research conducted by Woo et al. (2023) [16] substantiates our conclusions, indicating that ECC−based systems can realize a 14.6−fold decrease in energy consumption per bit in relation to RSA.[17] Furthermore, Muruganandham (2025) observeed that the key generation process in RSA is particularly detrimental to CO_2 emissions, with emissions escalating exponentially as key sizes increase to 4096 bits.[5]

3.3 PERFORMANCE OF HASHING ALGORITHMS

Performance of hashing algorithms (refer to Table 4 and Figure 2), SHA-256 has been identified as the most efficient cryptographic primitive among those assessed. The observed 100% CPU utilization indicates that the hashing process fully occupied the processing core during its short execution interval. It should not be interpreted as continuous utilization of the entire processor. No measurable increase in memory consumption was observed.

The architecture of SHA-256 is meticulously optimized for general-purpose CPU environments, whereas SHA3-256 employs a Keccak-based sponge architecture. Although SHA-3 offers structural variety and enhanced resistance to length-extension vulnerabilities, it entails markedly greater computational expenses on conventional processors. For long-term integrity verifications, SHA-256 remains the judicious option unless there is a specific demand for post-quantum resilience or architectural diversity.

From a green software engineering perspective, these results indicate that the cryptographic hash functions, particularly SHA-256 and SHA3-256, exhibit high computational efficiency when processing the evaluated datasets. Their short execution times and the absence of any measurable increase in memory consumption suggest that they require relatively limited computational resources. Although both algorithms reached 100% CPU utilization during execution, this does not necessarily imply higher energy consumption. Rather, it indicates that the hashing process is CPU-intensive for a very short period. Consequently, CPU utilization

should be interpreted together with execution time. An algorithm may fully utilize the processor during execution, yet complete the task rapidly, resulting in lower overall computational workload and potentially lower energy consumption. Therefore, from a green computing perspective, the combination of short execution time and negligible memory overhead suggests that SHA−256 and SHA3−256 are environmentally efficient cryptographic hash functions for the tested workload. .[18]

Table 4: Hashing Performance Comparison

Algorithm	Time (s)	Memory (KB)	CPU (%)
SHA-256	0.004988	0.0000	100.0
SHA3-256	0.004282	0.0000	100.0

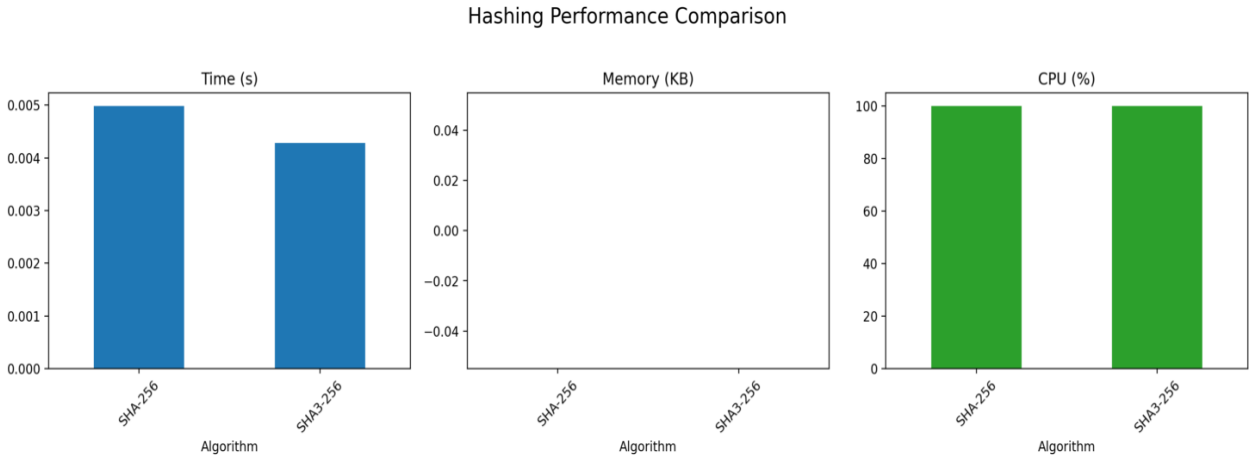

Figure 2 Performance of hashing algorithms

3.4 ANALYSIS OF THE GREEN SCORE METRIC

The "Green Score" metrics illustrated in Table 5 and Figure 3 furnish a comprehensive index of sustainability. Within a well-balanced weighting framework, AES stands as the unequivocal frontrunner for sustainable general-purpose encryption.

Table 5: Comparative Green Score Values (Balanced Profile)

Algorithm	Green Score	Sustainability Tier
DES	0.822649	Moderate (Due to Obsolescence)
RSA	0.662492	Low (High Energy Cost)
AES	0.402191	High (Highly Efficient)
ECC (Signing)	0.330000	Moderate (Greenest Asymmetric)

Note: Higher scores represent better resource alignment within experimental bounds.

The significance of this metric resides in its versatility. In a CPU-Critical Profile, prevalent in data centers, asymmetric algorithms suffer substantial drawbacks due to their complete load profile. This emphasizes the necessity for hardware offloading to specialized co-processors in order to alleviate thermal output and operational expenditures. Conversely, the Memory-Constrained Profile substantiates the critical role of symmetric ciphers in IoT devices characterized by limited SRAM availability [19].

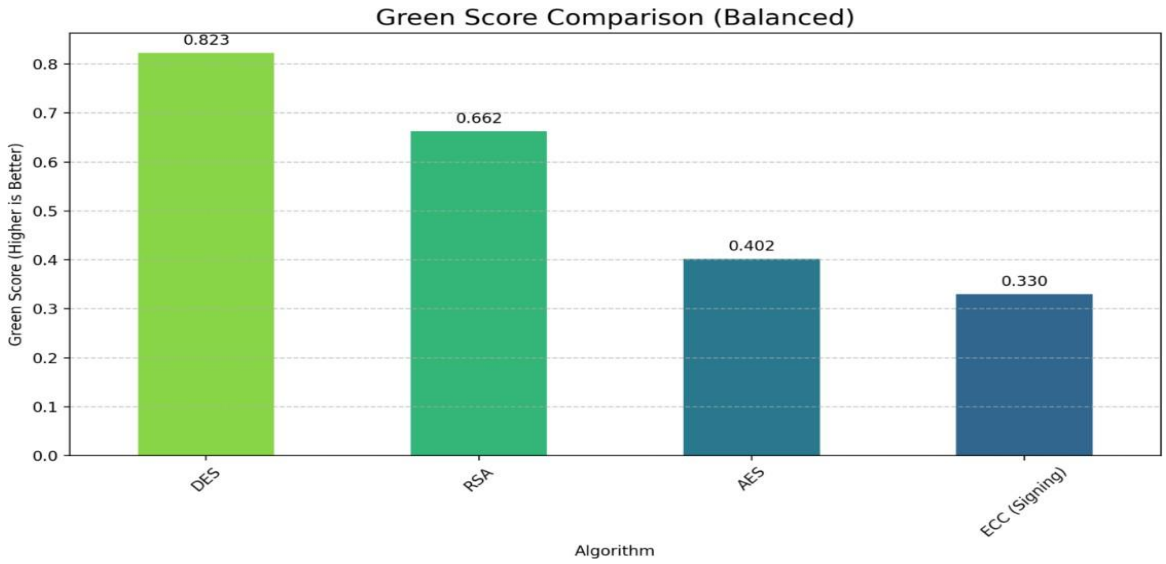

Figure 3 Comparative Green Score Values (Balanced Profile)

The findings of this study are generally consistent with recent research on green cryptography and lightweight cryptographic algorithms. Studies [7] and [8] demonstrated that lightweight encryption algorithms, including SIMON, SPECK, and PRESENT, achieve lower computational overhead and reduced resource consumption, making them well suited for Internet of Things (IoT) environments. These studies primarily evaluated performance in terms of execution time, memory usage, and energy efficiency on resource-constrained devices.

In contrast, the present study evaluates widely deployed cryptographic algorithms, namely AES-128, DES, RSA-2048, ECC-256, SHA-256, and SHA3-256, using a composite Green Score that integrates normalized execution time, memory consumption, and CPU utilization into a single environmental efficiency indicator. This approach extends previous work by providing a unified metric for comparing algorithms from different cryptographic categories rather than focusing exclusively on lightweight ciphers.

Although lightweight algorithms generally exhibit lower computational costs than conventional algorithms, they are designed for different application scenarios. AES, RSA, ECC, and the SHA family remain the dominant cryptographic standards in enterprise systems, cloud computing, financial applications, and secure Internet communications. Therefore, the proposed Green Score complements previous studies by enabling an objective evaluation of the environmental efficiency of widely adopted cryptographic algorithms under a common assessment framework.

4. CONCLUSION

This investigation elucidates that digital security and environmental stewardship are not inherently contradictory. By employing the "Green Score" metric, developers can transcend a solely security-centric design approach toward a paradigm that is cognizant of resource utilization. Notable recommendations encompass the obligatory adoption of ECC in lieu of RSA for public-key functionalities and the strategic implementation of pre-encryption compression to reduce computational demands. As we brace for a post-quantum landscape, the incorporation of energy-efficient primitives alongside hardware acceleration will be imperative for preserving the ecological integrity of our global digital infrastructure.[20]

REFERENCES:

[1] K. Achuthan, S. Sankaran, S. Roy, and R. Raman, "Integrating sustainability into cybersecurity: insights from machine learning based topic modeling," Discover Sustainability, vol. 6, no. 1, p. 11, Jan. 2025.

[2] R. Kumar et al., "AI-Driven Green Computing for Sustainable Information Security," Advances in Computational Intelligence and Robotics, pp. 99-126, Feb. 2025.

[3] Adi Bima Setiawan, "The Use of Cryptography in Digital Message Security," Journal of Computer Science and Information Technology, vol. 8, no. 2, pp. 67-74, Jul. 2024.

[4] J. Troutman and V. Rijmen, "Green Cryptography: Cleaner Engineering through Recycling," IEEE Security & Privacy, vol. 7, no. 4, pp. 71-73, Jul. 2009.

[5] S. K. Muruganandham, "Introducing Sustainability Metrics in Cryptography: Evaluating RSA Algorithms Through Energy Profiling Tools," Apr. 2025, doi: 10.21203/rs.3.rs-6343058/v1.

[6] J. Buchmann, "Sustainable Cryptography," 10.1007/978-981-15-5191-8_1, 2025.

[7] K. Sarker, "A systematic review on lightweight security algorithms for a sustainable IoT infrastructure," Discover Internet of Things, vol. 5, no. 1, p. 5, Jan. 2025.

[8] N. Zinabu, Y. Marye, K. Tune, and S. Demilew, "Comprehensive Analysis of Lightweight Cryptographic Algorithms for Battery-Limited Internet of Things Devices," International Journal of Distributed Sensor Networks, vol. 2025, 2025.

[9] S. Georgiou, S. Rizou, and D. Spinellis, "Software development lifecycle for energy efficiency: techniques and tools," ACM Computing Surveys (CSUR), vol. 52, no. 4, pp. 1-33, 2019.

[10] H. Gadiyar et al., "Implementing Lightweight Post-Quantum Signatures for Enhanced Security in Green IoE," in 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2025, pp. 1-6.

[11] B. Halak et al., "Evaluation of Performance, Energy, and Computation Costs of Quantum-Attack Resilient Encryption Algorithms for Embedded Devices," IEEE Access, vol. 12, pp. 8791-8805, 2024.

[12] A. Srinivasan et al., "Symmetric Encryption Leveraging Euclidean Geometry for Resource Constrained IoT Devices," in 2025 IEEE Guwahati Subsection Conference (GCON), 2025, doi: 10.1109/GCON65540.2025.11173351.

[13] S. Mubeena et al., "Lightweight Compression and Chaos-Based Encryption for Secure IoT Healthcare Data Storage on Blockchain," Journal of Engineering, Technology and Applied Science Research, Vol. 15 No. 6 (2025): December, 2025.

[14] S. A. Ansari, et al., "A systematic review of lightweight cryptographic schemes for security and privacy in IoT," Journal of Discover Computing, November 2025, doi: 10.1007/s10791-025-09755-3.

[15] M. Nalini, Yamini B, and Monica G K, "Sustainability Aspects and Future Prospects of Post-Quantum Cryptography in Smart IoT Ecosystems," Sep. 2025.

[16] J. Woo, V. A. Vasudevan, and B. Z. Kim, "CERMET: Coding for Energy Reduction with Multiple Encryption Techniques – It's easy being green," arXiv.org, Aug. 2023

[17] V. Naga et al., "Enhancing Data Security Solutions for Smart Energy Systems in IoT-Enabled Cloud Computing Environments through Lightweight Cryptographic Techniques," IOP Conference Series: Earth and Environmental Science, vol. 1375, 2024.

[18] N. F. Mufidah, et al., "Performance and Security Analysis of Lightweight Hash Functions in IoT," Journal of IT Development, December 2024, doi: 10.30591/jpit.v9i3.7633.

[19] A. Panigrahi et al., "Enhancing Trust and Lightweight Security in Green IoT: Models for Sustainable Computing," in 2025 2nd International Conference on Multidisciplinary Research and Innovations in Engineering (MRIE), 2025, pp. 280–284.

[20] M. U. Tariq, "AI-Powered Cybersecurity," Advances in Computational Intelligence and Robotics, Feb. 2025.