Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3196~3206
e-ISSN: 2548-6861

Security and Performance Analysis of PRESENT, SPECK, and ASCON Lightweight Cryptographic Algorithms in MQTT-Based IoT Environments

I Wayan Rangga Pinastawa [1]*, Susanto [2]*, Khoironi [3]**
* Department of Informatics, Universitas Pembangunan Nasional "Veteran" Jakarta, Jakarta, Indonesia
** Department of Informatics, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia.
rangga@upnvj.ac.id [1], susanto@upnvj.ac.id [2], khoironi@pens.ac.id [3]

Article Info

Article history:

Received 2026-06-03
Revised 2026-07-02
Accepted 2026-08-07

Keyword:

ASCON,
Lightweight Cryptography,
MQTT,
PRESENT,
SPECK

ABSTRACT

The rapid growth of the Internet of Things (IoT) has increased the demand for lightweight cryptographic algorithms capable of providing adequate security while maintaining low computational overhead on resource-constrained devices. This study aims to analyze and compare the ASCON, PRESENT, and SPECK algorithms in MQTT-based IoT environments using the MQTT-IoT-IDS2020 dataset as a source of communication payloads. The evaluation was conducted using a Python-based framework with security and randomness metrics, including Avalanche Effect, Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), Shannon Entropy, Approximate Entropy, Hamming Distance, Monobit Test, and Runs Test. Performance was evaluated using execution time, throughput, memory usage, and CPU usage. The validity of the results was strengthened through statistical analysis using 95% confidence intervals, One-Way ANOVA, and Tukey HSD. The results indicate that all three algorithms exhibit strong diffusion and randomness characteristics, with Avalanche Effect values close to the ideal value of 50% and ciphertext randomness metrics that satisfy the applied statistical tests. SPECK achieved the best performance with an execution time of 0.000238 seconds and a throughput of 2,199,891 bits/s, while PRESENT demonstrated the lowest memory consumption at 2.85 KB. Based on the calculated Security Score and Efficiency Score, PRESENT achieved the highest Security Score of 0.692, while SPECK achieved the highest Efficiency Score of 0.993, respectively. Statistical analysis revealed that significant differences among the algorithms primarily occurred in diffusion-related and computational performance metrics. Therefore, within the scope of the evaluated diffusion, randomness, and performance metrics, SPECK provides the most favorable balance between security-related characteristics and computational efficiency among the evaluated algorithms for MQTT-based IoT environments.

I. INTRODUCTION

The Internet of Things (IoT) has experienced rapid growth and has been widely implemented across various sectors, including smart homes, smart healthcare, smart agriculture, industrial monitoring, and wearable devices. This development has led to a significant increase in data exchange among interconnected devices, which are generally constrained by limited resources such as memory capacity, computational power, energy consumption, and communication bandwidth. In resource-constrained environments, the implementation of conventional cryptographic algorithms such as AES and RSA often introduces relatively high computational overhead, which may negatively affect overall system efficiency. Therefore, lightweight cryptography has emerged as a promising

solution for providing robust security mechanisms while requiring significantly fewer computational and hardware resources[1,2].

Several lightweight cryptographic algorithms have been widely adopted and extensively studied, including ASCON, PRESENT, and SPECK. These algorithms represent distinct design approaches within the field of lightweight cryptography. ASCON is a sponge permutation-based authenticated encryption algorithm that was selected as the winner of the National Institute of Standards and Technology (NIST) Lightweight Cryptography Standardization Process[3]. PRESENT is a lightweight block cipher based on a Substitution-Permutation Network (SPN) architecture, designed for hardware implementations with minimal resource requirements[4]. Meanwhile, SPECK is a lightweight block cipher based on the Addition-Rotation-XOR (ARX) design paradigm, developed to achieve high efficiency in software and embedded system implementations[5]. These differences in internal structure result in distinct security and performance characteristics when deployed in IoT environments.

The selection of these algorithms was motivated by their widespread use in lightweight cryptography research and their suitability as representative candidates for comparative evaluation. In addition, ASCON, PRESENT, and SPECK have been extensively investigated in previous studies, enabling the results obtained in this work to be more easily compared with existing literature. Evaluating these algorithms under the same MQTT-based IoT environment provides insights into the trade-offs between security-related characteristics and computational efficiency across different lightweight cryptographic approaches.

One of the most widely adopted communication protocols in IoT environments is Message Queuing Telemetry Transport (MQTT). MQTT was specifically designed to support lightweight communication over networks with limited bandwidth and devices with constrained computational capabilities[6]. However, the widespread deployment of MQTT over public networks has increased the need for effective security mechanisms to protect sensitive data during transmission. Therefore, selecting an appropriate lightweight cryptographic algorithm has become a critical factor in ensuring data confidentiality while maintaining communication efficiency in MQTT-based IoT environments[7].

In cryptography, the security of an algorithm is not solely determined by its encryption and decryption capabilities but also by the diffusion and randomness characteristics of the generated ciphertext. Diffusion refers to the ability of an algorithm to propagate a small change in the plaintext into substantial changes in the ciphertext, thereby increasing resistance against cryptanalytic attacks. This property is commonly evaluated using Avalanche Effect, Strict Avalanche Criterion (SAC), and Bit Independence Criterion (BIC)[8–10]. In addition, the randomness of ciphertext serves as an important indicator of an algorithm's resilience against statistical attacks. Randomness analysis can be performed using Shannon Entropy, Approximate Entropy Test, Monobit Test, and Runs Test to assess the distribution and unpredictability of encrypted bit sequences[11,12].

Radhakrishnan et al. evaluated the security and efficiency of AES-128, SPECK, and ASCON on resource-constrained IoT devices using Arduino Nano and Arduino Micro platforms. The evaluation considered several performance and security metrics, including memory utilization, execution time, throughput, latency, key scheduling performance, and security analysis. Their results demonstrated that SPECK achieved the highest throughput and the lowest latency, whereas ASCON provided lower memory consumption while maintaining a high level of security[13].

Iqbal et al. evaluated several lightweight cryptographic algorithms, including PRESENT, SPECK, SIMON, HIGHT, and KATAN, in IoT environments with a focus on security rating, execution time, and energy consumption. The results indicated that SPECK outperformed PRESENT in terms of computational efficiency, while the proposed algorithm achieved the best balance among security, processing speed, and energy efficiency[1].

Kurian and Chen conducted a comprehensive evaluation of the ASCON hash function family by comparing it with SHA3-256, BLAKE2s, MurmurHash3, and xxHash in IoT and blockchain environments using a Python-based SMHasher framework. Their findings demonstrated that Ascon-Hash256 exhibited strong diffusion properties, low statistical bias, and competitive performance across various application scenarios. However, the study did not incorporate randomness testing based on the NIST SP 800-22 standard, nor did it evaluate Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), or statistical validation techniques to quantitatively verify the significance of differences among the evaluated algorithms[14].

Based on the reviewed literature, the evaluation of lightweight cryptographic algorithms has been predominantly focused on performance-related metrics such as execution time, throughput, memory utilization, and energy consumption. In contrast, security evaluations incorporating Avalanche Effect, Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), entropy analysis, NIST SP 800-22-based randomness testing, and statistical validation remain relatively limited. Furthermore, the use of real-world MQTT communication datasets as plaintext sources has received little attention in existing studies. Therefore, a comprehensive evaluation framework that integrates security, randomness, performance, and statistical validity within a unified testing environment is needed.

To address this gap, this study introduces an integrated evaluation framework that combines diffusion analysis, randomness assessment, computational performance evaluation, and statistical validation within an MQTT-based IoT environment. Unlike many previous studies that primarily focus on computational performance or a limited set of security metrics, the proposed framework evaluates

lightweight cryptographic algorithms using multiple complementary perspectives. In addition, payloads derived from the MQTT-IoT-IDS2020 dataset are employed to provide a more representative evaluation scenario based on real IoT communication traffic. Security Score and Efficiency Score indicators are also incorporated to facilitate trade-off analysis between security-related characteristics and computational efficiency.

This study aims to comparatively evaluate ASCON, PRESENT, and SPECK in terms of security, randomness, and computational performance within MQTT-based IoT environments. The evaluation employs Avalanche Effect, Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), Shannon Entropy, Approximate Entropy Test, Monobit Test, Runs Test, Hamming Distance, execution time, throughput, memory usage, and CPU usage. To further enhance the validity of the findings, statistical analyses are conducted using confidence intervals, One-Way ANOVA, Tukey HSD, and Eta Squared effect size analysis.

The findings of this study are expected to contribute both academically and practically to the field of lightweight cryptography. From an academic perspective, this research provides a comprehensive evaluation framework for assessing lightweight cryptographic algorithms in MQTT-based IoT environments. From a practical perspective, the results may serve as a reference for researchers, IoT system developers, and information security practitioners in selecting lightweight cryptographic algorithms that are suitable for devices with constraints in computational power, memory capacity, and energy resources.

II. METHODOLOGY

This study adopts an experimental quantitative approach to conduct a comparative analysis of three lightweight cryptographic algorithms, namely ASCON, PRESENT, and SPECK, in an MQTT-based Internet of Things (IoT) environment. All experiments were implemented using the Python programming language and executed within the same computational environment to ensure consistency throughout the evaluation process. The study focuses on analyzing the security, randomness, performance, and statistical validity of each algorithm through an integrated evaluation framework.

A. Dataset Preparation

The dataset used in this study is MQTT-IoT-IDS2020, obtained from the Kaggle platform. This dataset contains MQTT communication traffic that represents real-world IoT device activities, including both normal communications and various attack scenarios. The use of a real-world dataset was intended to provide a more representative evaluation compared with synthetic plaintext or randomly generated data, which are commonly employed in cryptographic research.

The use of MQTT-IoT-IDS2020 in this study was not intended to influence the cryptographic strength of the evaluated algorithms, since the security properties of a cryptographic algorithm are independent of the dataset used as input. Instead, the dataset was employed to generate realistic MQTT payloads that better represent practical IoT communication patterns. Compared with synthetic or randomly generated plaintext, real MQTT communication data provide a more representative evaluation scenario for assessing computational performance, diffusion behavior, and ciphertext randomness in MQTT-based IoT environments.

During the preprocessing stage, the dataset was loaded using the Pandas library, and random sampling was performed with a seed value of 42 to ensure the reproducibility of the experiments. From the entire dataset, a maximum of 3,000 records were selected as the research subset. Subsequently, 1,000 samples were utilized for the evaluation process to maintain a balance between data representativeness and computational complexity.

Each record was subsequently converted into a communication payload by extracting six key attributes from the dataset. These attributes were concatenated using the "|" delimiter and represented as a byte sequence. To ensure a consistent input length across all evaluated algorithms, each payload was either truncated or padded to a fixed size of 64 bytes.

B. Cryptographic Algorithms

This study compares three lightweight cryptographic algorithms that represent different design paradigms.

ASCON is a sponge permutation-based authenticated encryption algorithm that was selected as the winner of the National Institute of Standards and Technology (NIST) Lightweight Cryptography Standardization Process. In this study, ASCON was configured with a 128-bit key and a 128-bit nonce, both of which were randomly generated for each evaluation iteration.

PRESENT is a lightweight block cipher based on the Substitution-Permutation Network (SPN) architecture, featuring a 64-bit block size and an 80-bit key length. The PRESENT-80 configuration was selected because it is the most widely adopted standard configuration in lightweight cryptography research and resource-constrained devices. Furthermore, this configuration represents the original specification of the PRESENT algorithm, which was specifically designed for highly constrained environments such as RFID systems and embedded devices.

SPECK is a lightweight block cipher based on the Addition-Rotation-XOR (ARX) design paradigm, developed to achieve high efficiency in software implementations. In this study, the SPECK-128 configuration with a 128-bit key length was employed.

To ensure a fair comparison, all three algorithms were implemented in Python and evaluated using the same communication payloads. This approach ensured consistency in the evaluation process and enabled an objective comparison of their security, randomness, and performance characteristics.

C. Evaluation Framework

All experiments were conducted using Google Colaboratory (Google Colab), a cloud-based computational platform. The evaluation framework was implemented in Python 3.12.13 and executed on a Linux-based environment (Linux 6.6.122+, x86_64 architecture) with approximately 12.67 GB of available memory. The implementation utilized NumPy 2.0.2 for numerical computation, Pandas 2.2.2 for data processing, SciPy 1.16.3 for scientific computing, and Statsmodels 0.14.6 for statistical analysis. To ensure reproducibility, a fixed random seed value of 42 was applied during the dataset sampling process.

The same experimental environment was maintained throughout the evaluation process to ensure consistency across all tested algorithms. In each iteration, the payloads generated from the dataset were encrypted using ASCON, PRESENT, and SPECK. Encryption keys were generated using a cryptographically secure random number generator, where PRESENT employed an 80-bit key according to its original specification, while SPECK and ASCON used 128-bit keys. In addition, ASCON utilized a 128-bit nonce that was randomly generated for each encryption process.

To evaluate the diffusion capability of each algorithm, a single-bit modification was introduced into the plaintext. The ciphertext generated from the original plaintext and the ciphertext generated from the modified plaintext were then compared using various security and randomness metrics. This approach enabled the assessment of how effectively each algorithm propagated minor changes in the input into significant changes in the resulting ciphertext.

Security evaluation was conducted to assess the diffusion characteristics, sensitivity to plaintext modifications, and randomness properties of the generated ciphertext. Diffusion performance was measured using Avalanche Effect, Strict Avalanche Criterion (SAC), and Bit Independence Criterion (BIC) to determine the extent to which small changes in the plaintext influence the resulting ciphertext[8–10]. The randomness of the ciphertext was analyzed using Shannon Entropy and the Approximate Entropy Test to evaluate the degree of randomness and pattern complexity in the encrypted data[11]. Furthermore, Hamming Distance was employed to quantify the number of bit differences between ciphertexts generated from the original plaintext and the modified plaintext[15]. To further strengthen the randomness analysis, this study applied the Monobit Test and Runs Test, which are statistical tests used to evaluate bit distribution and transition patterns between zeros and ones within the generated ciphertext[12].

In addition to security evaluation, this study also assessed algorithm performance to determine their computational efficiency in IoT environments. Performance evaluation was conducted using execution time to measure the time required for the encryption process, throughput to quantify the amount of data processed per unit time, memory usage to assess memory consumption during encryption, and CPU usage to measure processor utilization throughout the encryption process[13,16]. The combination of security and performance metrics was employed to provide a more comprehensive understanding of the trade-off between security strength and computational efficiency among the evaluated lightweight cryptographic algorithms in MQTT-based IoT environments.

Execution time was measured using Python's time.perf_counter() function and recorded for each encryption operation. Throughput was calculated as the ratio between the processed data size (in bits) and the corresponding encryption execution time. For throughput measurement, each MQTT payload was normalized to a fixed size of 64 bytes (512 bits) to ensure a fair comparison among the evaluated algorithms. The reported throughput values represent encryption throughput only and therefore do not include key generation or decryption operations. Memory consumption was measured using the tracemalloc library, where the peak memory allocation observed during the encryption process was recorded and reported in kilobytes (KB). CPU utilization was measured using the psutil.cpu_percent() function with a sampling interval of 0.05 seconds. To improve measurement reliability, all experiments were performed using 1,000 MQTT payload samples under the same Google Colaboratory environment, and the reported performance metrics represent the average values obtained across all experimental iterations under identical hardware and software conditions.

D. Statistical Analysis

To enhance the validity of the research findings, statistical analyses were performed on all security and performance metrics obtained from the evaluation process. The analysis began with the calculation of 95% confidence intervals to estimate the range within which the true mean value of each metric was expected to lie for ASCON, PRESENT, and SPECK. Subsequently, a One-Way Analysis of Variance (ANOVA) was conducted to determine whether statistically significant differences existed among the evaluated algorithms for each security and performance parameter.

In this analysis, the null hypothesis (H_0) stated that no statistically significant differences existed among the evaluated algorithms, whereas the alternative hypothesis (H_1) stated that a statistically significant difference existed in at least one pair of algorithms.

In addition to statistical significance testing, effect size analysis was considered to evaluate the practical magnitude of the observed differences among the evaluated algorithms. Effect size provides complementary information to p-values by quantifying the proportion of variance explained by the grouping factor. In this study, the effect size associated with the ANOVA analysis was assessed using Eta Squared η^2 which represents the ratio between the between-group sum of squares and the total sum of squares. According to commonly adopted guidelines, η^2 values of approximately 0.01, 0.06, and 0.14 correspond to small, medium, and large effect sizes, respectively.

If the ANOVA results indicated statistical significance according to the predefined significance level, a post-hoc

analysis was subsequently performed using Tukey's Honestly Significant Difference (HSD) test to identify the specific pairs of algorithms that differed significantly[17].

The combination of confidence intervals, One-Way ANOVA, and Tukey HSD was employed to ensure that the observed differences were not merely the result of sample variability but were supported by sufficient statistical evidence[18].

E. Security and Efficiency Scoring

In addition to the individual analysis of each evaluation metric, this study calculated Security Score and Efficiency Score to provide a more comprehensive assessment of the balance between security strength and computational efficiency for each evaluated algorithm.

The calculation was performed using the Min-Max Normalization approach, which transformed all evaluation metrics into a common scale ranging from 0 to 1. The normalization process was carried out using Equation (1).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Where:
X denotes the value of the evaluated metric;
X_{min} denotes the minimum value of the metric;
X_{max} denotes the maximum value of the metric; and
X_{norm} denotes the normalized value.

For metrics exhibiting a *lower-is-better* characteristic, including Execution Time, Memory Usage, and CPU Usage, inverse normalization was applied after the normalization process using Equation (2).

$$X_{inv} = 1 - X_{norm} \quad (2)$$

Where:
X_{inv} denotes the inversely normalized value; and
X_{norm} denotes the normalized value.

After the normalization process was completed, a composite Security Score was calculated to provide an aggregated representation of the security-related characteristics of each evaluated algorithm. The Security Score was derived from diffusion and randomness metrics that are commonly employed in cryptographic evaluation, namely Avalanche Effect, SAC, BIC, Hamming Distance, Shannon Entropy, Monobit Test, and Runs Test. Since no established consensus exists regarding the relative importance of these metrics for lightweight cryptographic evaluation, all metrics were assigned equal weights following the principle of equal contribution. Consequently, the Security Score reflects the overall diffusion and randomness characteristics observed in this study rather than an absolute measure of cryptographic security. The calculation was performed using Equation (3).

$$\text{SecurityScore} = \frac{1}{n}\sum_{i=1}^{n} M_i \quad (3)$$

Where:
M_i denotes the normalized value of the (i)-th security or randomness metric; and
n denotes the total number of security and randomness metrics used in the evaluation.

Similarly, the Efficiency Score was calculated to provide a unified measure of computational efficiency. The score was derived from normalized Execution Time, Throughput, Memory Usage, and CPU Usage metrics. Equal weighting was adopted to avoid introducing subjective preferences toward a specific performance metric and to ensure that all evaluated aspects contributed equally to the final efficiency assessment. The calculation was performed using Equation (4).

$$\text{EfficiencyScore} = \frac{1}{m}\sum_{j=1}^{m} P_j \quad (4)$$

Where:
P_j denotes the normalized value of the (j)-th performance metric; and
m denotes the total number of performance metrics used in the evaluation.

This approach was adopted to facilitate the identification of the algorithm that provides the most favorable trade-off between security strength and computational efficiency. A higher Security Score indicates stronger security and randomness characteristics, whereas a higher Efficiency Score reflects better computational efficiency. Consequently, these scores can serve as comprehensive evaluation indicators for determining the most suitable lightweight cryptographic algorithm for MQTT-based IoT environments.

It should be noted that the Security Score and Efficiency Score were introduced solely as comparative indicators to facilitate trade-off analysis among the evaluated algorithms. These composite scores do not replace formal cryptographic security proofs and should not be interpreted as definitive measures of overall cryptographic strength.

F. Research Workflow

The research process began with the acquisition and preprocessing of the MQTT-IoT-IDS2020 dataset. Subsequently, the data were converted into MQTT communication payloads that served as plaintext inputs for the encryption process. The payloads were then encrypted using the ASCON, PRESENT, and SPECK algorithms. The resulting ciphertexts were utilized to compute various security and performance metrics. The measurement results were further analyzed using confidence intervals, One-Way ANOVA, and Tukey HSD to identify significant differences in security and performance among the evaluated algorithms. The final stage involved interpreting the results to determine the algorithm that achieved the most favorable balance between security and computational efficiency in MQTT-based IoT environments.

To facilitate a clearer understanding of the research procedure, Figure 1 illustrates the overall workflow of the study. The workflow encompasses dataset acquisition and

preprocessing, MQTT payload generation, encryption using ASCON, PRESENT, and SPECK, security and performance evaluation, statistical analysis, and interpretation of the experimental results.

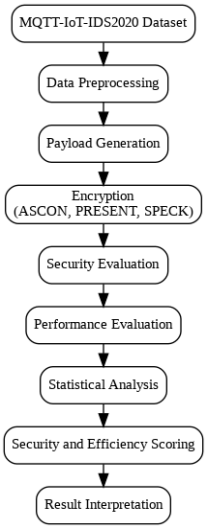

Figure 1. Research Workflow

III. Results And Discussion

This section presents and discusses the results of the security, randomness, performance, and statistical evaluations of the PRESENT, SPECK, and ASCON algorithms in an MQTT-based IoT environment. The experiments were conducted using the MQTT-IoT-IDS2020 dataset and the Python-based evaluation framework described in the methodology section. The findings are analyzed to compare the security characteristics, computational efficiency, and overall trade-offs among the evaluated lightweight cryptographic algorithms.

A. Security and Randomness Analysis

The experimental results were analyzed using various security and randomness metrics to identify the characteristics of each evaluated algorithm. The analysis focused on assessing diffusion capability, sensitivity to plaintext modifications, and the randomness properties of the generated ciphertext. These evaluations provide insights into the effectiveness of each algorithm in protecting data confidentiality and resisting cryptanalytic and statistical attacks within MQTT-based IoT environments.

TABLE I
SECURITY AND RANDOMNESS EVALUATION RESULTS

Metric	PRESENT	SPECK	ASCON
Avalanche Effect (%)	50.10	50.05	49.44
SAC	0.2813	0.3125	0.2932
BIC	0.1413	0.1334	0.1428
Approximate Entropy	0.6892	0.6893	0.6892
Shannon Entropy	5.7689	5.7662	5.7650
Correlation	-0.0139	-0.0142	-0.0190
Hamming Distance	256.54	256.27	253.15
Monobit P-Value	0.4909	0.4982	0.4994
Runs P-Value	0.6072	0.6070	0.6066

Table I presents the results of the security and randomness evaluation of ASCON, PRESENT, and SPECK using various diffusion and statistical metrics. The results indicate that all three algorithms exhibit strong diffusion characteristics, with Avalanche Effect values close to the ideal value of 50%. PRESENT achieved the highest Avalanche Effect value at 50.10%, followed closely by SPECK at 50.05%, while ASCON achieved 49.44%. These results demonstrate that a one-bit modification in the plaintext can produce substantial changes in the ciphertext, thereby enhancing resistance against differential cryptanalysis.

Regarding randomness evaluation, all three algorithms produced nearly identical Shannon Entropy values of approximately 5.77. Similarly, the Approximate Entropy values were highly comparable, averaging around 0.689. These findings indicate that all evaluated algorithms generate ciphertext with a high degree of randomness and a relatively uniform bit distribution. This observation is further supported by the results of the Monobit Test and Runs Test, where all algorithms produced p-values greater than 0.05, suggesting the absence of statistically significant deviations in ciphertext bit distributions.

The sensitivity analysis based on Hamming Distance revealed that PRESENT and SPECK exhibited slightly stronger diffusion characteristics than ASCON. PRESENT achieved a Hamming Distance of 256.54 bits, followed by SPECK with 256.27 bits, whereas ASCON achieved 253.15 bits. These results indicate that PRESENT and SPECK propagate plaintext modifications more effectively throughout the ciphertext.

Overall, the security evaluation results suggest that PRESENT and SPECK provide slightly superior diffusion performance compared with ASCON, whereas all three algorithms demonstrate comparable randomness characteristics based on entropy analysis, the Monobit Test, and the Runs Test.

Although diffusion and randomness metrics provide valuable insights into the statistical characteristics of the generated ciphertext, these metrics alone do not constitute proof of cryptographic security. Metrics such as Avalanche Effect, SAC, BIC, Hamming Distance, Shannon Entropy, Monobit Test, and Runs Test are useful for evaluating diffusion behavior and statistical randomness; however, strong performance on these metrics does not necessarily imply resistance against differential cryptanalysis, linear cryptanalysis, side-channel attacks, or key-recovery attacks. Therefore, the results presented in this study should be interpreted as indicators of diffusion and randomness characteristics rather than definitive evidence of overall cryptographic strength, and the reported Security Score

should be regarded as a comparative indicator derived from the evaluated metrics.

B. Performance Analysis

In addition to security and randomness assessments, performance evaluation was conducted to determine the computational efficiency of each algorithm when deployed in an MQTT-based IoT environment.

TABLE II
PERFORMANCE EVALUATION RESULTS

Metric	PRESENT	SPECK	ASCON
Execution Time (s)	0.00522206	0.00023830	0.00027694
Throughput (bit/s)	99,284.54	2,199,891.60	1,885,963.22
Memory Usage (KB)	2.85	3.12	12.45
CPU Usage (%)	9.74	8.30	8.39

Table II presents the performance evaluation results of the evaluated algorithms. The results reveal considerable differences in computational efficiency among ASCON, PRESENT, and SPECK. In terms of execution time, SPECK was the fastest algorithm, achieving an average execution time of 0.000238 seconds, followed by ASCON at 0.000277 seconds, whereas PRESENT required a substantially longer execution time of 0.005222 seconds. These findings suggest that the Addition-Rotation-XOR (ARX) design employed by SPECK provides greater computational efficiency than the Substitution-Permutation Network (SPN) architecture utilized by PRESENT.

A similar trend can be observed in the throughput results. SPECK achieved the highest throughput of 2,199,891 bits/s, followed by ASCON with 1,885,963 bits/s, while PRESENT reached only 99,285 bits/s. Consequently, SPECK processed data approximately 22 times faster than PRESENT and about 1.17 times faster than ASCON.

From a resource utilization perspective, PRESENT demonstrated the lowest memory consumption at 2.85 KB, followed by SPECK at 3.12 KB. In contrast, ASCON required substantially more memory, reaching 12.45 KB. However, CPU utilization was relatively similar across all three algorithms, ranging from approximately 8% to 10%, indicating no substantial differences in processor usage.

Overall, the results indicate that SPECK is the most computationally efficient algorithm among the evaluated candidates, while PRESENT offers the lowest memory footprint. ASCON provides competitive performance but requires significantly more memory resources than both PRESENT and SPECK.

C. Security and Efficiency Trade-off Analysis

Although the security and performance evaluations provide valuable insights into the characteristics of each algorithm, a combined analysis is necessary to assess the balance between security strength and computational efficiency. Such an analysis is particularly important in MQTT-based IoT environments, where cryptographic algorithms must provide adequate security while operating under resource constraints. Therefore, Security Score and Efficiency Score were utilized to provide a comprehensive comparison of the trade-offs among ASCON, PRESENT, and SPECK.

Security Score was derived from normalized Avalanche Effect, SAC, BIC, Hamming Distance, Shannon Entropy, Monobit Test, and Runs Test results, whereas Efficiency Score was derived from normalized Execution Time, Throughput, Memory Usage, and CPU Usage metrics.

TABLE III
SECURITY SCORE AND EFFICIENCY SCORE

Algorithm	Security Score	Efficiency Score
PRESENT	0.6920	0.2500
SPECK	0.6565	0.9929
ASCON	0.3403	0.6955

Table III presents the Security Score and Efficiency Score obtained from the normalized security, randomness, and performance metrics. These scores provide a comprehensive assessment of the trade-off between security strength and computational efficiency for each evaluated algorithm. A higher Security Score indicates stronger security and randomness characteristics, whereas a higher Efficiency Score reflects better computational efficiency.

The results indicate that PRESENT achieved the highest Security Score of 0.692, whereas SPECK achieved the highest Efficiency Score of 0.993. Although PRESENT demonstrated the strongest security-related characteristics among the evaluated algorithms, it recorded the lowest Efficiency Score of 0.250, primarily due to its substantially higher execution time compared with SPECK and ASCON. ASCON achieved a moderate Efficiency Score of 0.695 but produced the lowest Security Score of 0.340. Meanwhile, SPECK achieved a Security Score of 0.657, which was slightly lower than PRESENT but substantially higher than ASCON.

The lower Security Score achieved by ASCON was primarily influenced by its lower Avalanche Effect and Hamming Distance values, despite exhibiting randomness characteristics comparable to those of PRESENT and SPECK. This finding suggests that the differences in Security Score were mainly driven by diffusion-related properties rather than by the randomness quality of the generated ciphertext.

Figure 2 illustrates the relationship between the Security Score and Efficiency Score of the evaluated algorithms. As shown in the figure, PRESENT achieved the highest Security Score (0.692), indicating slightly stronger diffusion and statistical security characteristics than the other evaluated algorithms. In contrast, SPECK achieved the highest Efficiency Score (0.993), substantially outperforming PRESENT and ASCON in terms of computational efficiency. ASCON exhibited moderate computational efficiency but produced the lowest Security Score among the evaluated algorithms.

Although PRESENT achieved the highest Security Score, its computational efficiency was considerably lower than that of SPECK. Conversely, SPECK maintained a comparable Security Score while providing a substantial advantage in execution time, throughput, and overall computational efficiency. These findings highlight the trade-off between security-related characteristics and computational performance among the evaluated lightweight cryptographic algorithms.

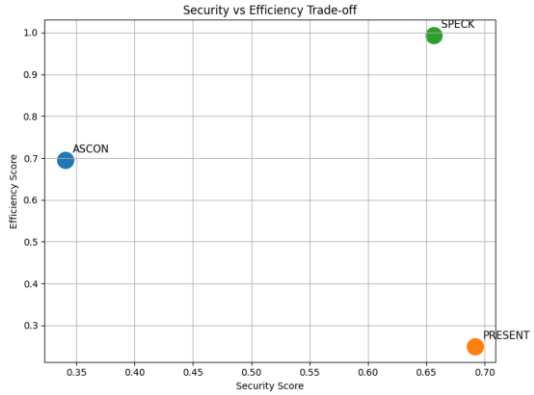

Figure 2. Security versus Efficiency Trade-off among PRESENT, SPECK, and ASCON

The trade-off analysis indicates that no single algorithm achieved the highest score in both security and efficiency. PRESENT obtained the highest Security Score, whereas SPECK achieved the highest Efficiency Score. However, the difference in Security Score between PRESENT and SPECK was relatively small compared with the substantial efficiency advantage demonstrated by SPECK. Therefore, SPECK provides the most favorable balance between security-related characteristics and computational efficiency for MQTT-based IoT environments, where resource constraints make computational efficiency a critical consideration. Nevertheless, this conclusion is limited to the evaluated diffusion, randomness, and performance metrics and should not be interpreted as evidence of superior resistance against all forms of cryptanalytic attacks.

To verify the stability of the experimental results and identify statistically significant differences among the evaluated algorithms, additional analyses were conducted using 95% confidence intervals, One-Way ANOVA, and Tukey's Honestly Significant Difference (HSD) test.

D. Statistical Analysis

The results of the 95% confidence interval analysis indicate that all evaluated metrics exhibited relatively narrow estimation ranges, suggesting a high degree of stability and consistency across the experimental samples. For example, the Avalanche Effect values for PRESENT, SPECK, and ASCON were within the ranges of 49.97–50.24, 49.91–50.19, and 49.31–49.58, respectively. In terms of performance, the Execution Time of PRESENT ranged from 0.0052 to 0.0053 seconds, while SPECK and ASCON exhibited ranges of 0.0002–0.0002 seconds and 0.0003–0.0003 seconds, respectively. Similarly, Throughput values demonstrated relatively narrow confidence intervals, ranging from 98,613 to 99,955 bits/s for PRESENT, 2,182,318 to 2,217,466 bits/s for SPECK, and 1,872,012 to 1,899,915 bits/s for ASCON. The relatively narrow confidence intervals observed across both security and performance metrics indicate that the experimental results are reliable and are not substantially affected by sample variability.

TABLE IV
ONE-WAY ANOVA AND EFFECT SIZE RESULTS FOR SECURITY, RANDOMNESS, AND PERFORMANCE METRICS

Metric	p-value	Significant	η^2	Effect Size
Avalanche Effect	8.24×10^{-13}	Yes	0.018	Small
SAC	< 0.001	Yes	0.941	Large
BIC	< 0.001	Yes	0.672	Large
Approximate Entropy	0.8021	No	0.000	Negligible
Shannon Entropy	0.5230	No	0.000	Negligible
Correlation	0.5890	No	0.000	Negligible
Execution Time	< 0.001	Yes	0.978	Large
Throughput	< 0.001	Yes	0.951	Large
Memory Usage	< 0.001	Yes	0.999	Large
CPU Usage	0.1670	No	0.001	Negligible
Hamming Distance	8.24×10^{-13}	Yes	0.018	Small
Monobit Test	0.7785	No	0.000	Negligible
Runs Test	0.9984	No	0.000	Negligible

Table IV presents the results of the One-Way ANOVA conducted on all security, randomness, and performance metrics. The results indicate that Avalanche Effect, SAC, BIC, Hamming Distance, Execution Time, Throughput, and Memory Usage produced p-values below the significance threshold of 0.05, indicating statistically significant differences among the evaluated algorithms. In contrast, Approximate Entropy, Shannon Entropy, Correlation, CPU Usage, the Monobit Test, and the Runs Test yielded p-values greater than 0.05, suggesting no statistically significant differences among the algorithms for these metrics. These findings demonstrate that the main distinctions among ASCON, PRESENT, and SPECK arise from their diffusion capabilities and computational performance rather than from the randomness quality of the generated ciphertext, which remained statistically comparable across all evaluated algorithms.

To complement the ANOVA results, effect size analysis was performed using Eta Squared (η^2). While p-values indicate whether statistically significant differences exist among the evaluated algorithms, Eta Squared quantifies the

practical magnitude of those differences. The results show that SAC, BIC, Execution Time, Throughput, and Memory Usage exhibited large effect sizes, indicating that algorithm selection substantially influenced these metrics. In contrast, Approximate Entropy, Shannon Entropy, Correlation, CPU Usage, Monobit Test, and Runs Test exhibited negligible effect sizes, suggesting that the observed differences were practically insignificant. Avalanche Effect and Hamming Distance produced small effect sizes, indicating only modest practical differences among the evaluated algorithms.

TABLE V
SUMMARY OF TUKEY HSD POST-HOC ANALYSIS RESULTS.

Metric	Algorithm Pairs with Significant Differences
Avalanche Effect	PRESENT-ASCON, SPECK-ASCON
SAC	PRESENT-SPECK, PRESENT-ASCON, SPECK-ASCON
BIC	PRESENT-SPECK, PRESENT-ASCON, SPECK-ASCON
Approximate Entropy	None
Shannon Entropy	None
Correlation	None
Execution Time	PRESENT-SPECK, PRESENT-ASCON, SPECK-ASCON
Throughput	PRESENT-SPECK, PRESENT-ASCON, SPECK-ASCON
Memory Usage	PRESENT-SPECK, PRESENT-ASCON, SPECK-ASCON
CPU Usage	None
Hamming Distance	PRESENT-ASCON, SPECK-ASCON
Monobit Test	None
Runs Test	None

The Tukey HSD results presented in Table V indicate that most statistically significant differences occurred in diffusion-related and computational performance metrics. For Avalanche Effect and Hamming Distance, no statistically significant differences were observed between PRESENT and SPECK, whereas both algorithms differed significantly from ASCON. These findings indicate that PRESENT and SPECK possess comparable diffusion characteristics and outperform ASCON in terms of ciphertext diffusion capability.

For SAC and BIC, statistically significant differences were identified across all algorithm pairs. This result indicates that although PRESENT and SPECK achieved similar average Avalanche Effect values, their bit-level diffusion characteristics differ and can be distinguished through SAC and BIC analyses.

Regarding computational performance, all algorithm pairs exhibited statistically significant differences in Execution Time, Throughput, and Memory Usage. These findings confirm that SPECK possesses computational characteristics that are significantly different from those of PRESENT and ASCON. Furthermore, PRESENT demonstrated the lowest memory consumption, whereas ASCON required substantially more memory resources than the other evaluated algorithms.

In contrast, randomness-related metrics, including Approximate Entropy, Shannon Entropy, Correlation, Monobit Test, and Runs Test, did not exhibit statistically significant differences among the evaluated algorithms. These findings are consistent with the ANOVA results and indicate that all three algorithms generate ciphertext with comparable randomness characteristics.

Based on the security, randomness, performance, and statistical evaluations conducted in this study, SPECK demonstrates the most favorable balance between security strength and computational efficiency. PRESENT provides competitive diffusion characteristics while maintaining a low memory footprint, whereas ASCON delivers competitive performance but requires substantially greater memory resources. These findings form the basis for the conclusions presented in the following section.

The findings obtained in this study are generally consistent with previous research on lightweight cryptography for IoT environments. Iqbal et al. reported that lightweight cryptographic algorithms exhibit different trade-offs between security, computational efficiency, and resource consumption, emphasizing that no single algorithm consistently outperforms others across all evaluation criteria[1]. The results of the present study support this observation, as PRESENT achieved the highest Security Score, whereas SPECK demonstrated the highest computational efficiency. Similarly, Radhakrishnan et al. highlighted the importance of jointly considering security and performance characteristics when selecting lightweight cryptographic algorithms for resource-constrained IoT devices[13]. Consistent with their findings, the present study shows that algorithm selection should depend on application requirements rather than a single evaluation metric.

However, some differences were observed relative to the broader literature on ASCON. Although ASCON has been standardized by NIST as the lightweight cryptography standard for constrained devices[3], the present evaluation showed lower diffusion-related characteristics and higher memory consumption compared with PRESENT and SPECK under the tested configuration. These findings do not contradict the NIST standardization process, as the selection of ASCON was based on a broader set of considerations including security, functionality, implementation flexibility, and suitability for constrained environments rather than solely on the diffusion and performance metrics evaluated in this study. Therefore, the results should be interpreted within the scope of the evaluation framework and implementation settings adopted in this work.

Overall, the results confirm the existence of trade-offs among security-related characteristics, computational efficiency, and resource consumption, while extending previous studies through the integration of diffusion analysis, randomness assessment, performance evaluation, and

statistical validation within an MQTT-based IoT communication environment.

It should be noted that the MQTT-IoT-IDS2020 dataset was used solely as a source of representative MQTT communication payloads and does not affect the inherent cryptographic properties of ASCON, PRESENT, or SPECK. The primary purpose of using this dataset was to improve the realism of the evaluation environment by employing plaintext derived from actual IoT communication traffic rather than synthetic or randomly generated data.

Several limitations of this study should be acknowledged. First, all experiments were conducted in a Python-based environment using Google Colaboratory rather than on physical IoT devices. Consequently, the measured execution time, memory usage, CPU utilization, and throughput may not fully reflect the behavior of the evaluated algorithms when deployed on resource-constrained platforms such as ESP32, Arduino, Raspberry Pi, or other embedded systems. Second, the evaluation focused on diffusion, randomness, and computational performance metrics and did not include practical cryptanalytic assessments such as differential cryptanalysis, linear cryptanalysis, side-channel analysis, or key-recovery attacks. Therefore, the reported Security Score should be interpreted as a comparative indicator derived from the evaluated metrics rather than as a definitive measure of overall cryptographic strength. Third, the experiments were performed at the encryption algorithm level and did not include end-to-end MQTT communication scenarios involving network latency, protocol overhead, energy consumption, or interoperability with security protocols such as TLS and DTLS. Future research should validate the findings on real IoT hardware platforms, evaluate end-to-end MQTT communication performance, and incorporate additional cryptanalytic assessments to provide a more comprehensive evaluation of lightweight cryptographic algorithms.

IV. CONCLUSION

This study conducted a comparative analysis of the lightweight cryptographic algorithms ASCON, PRESENT, and SPECK in an MQTT-based IoT environment using the MQTT-IoT-IDS2020 dataset. The security evaluation results demonstrated that all three algorithms were capable of producing secure ciphertext, as evidenced by Avalanche Effect values close to the ideal value of 50% and Monobit Test and Runs Test results that satisfied randomness criteria. However, PRESENT and SPECK exhibited slightly stronger diffusion characteristics than ASCON based on the Avalanche Effect and Hamming Distance metrics.

From a performance perspective, SPECK achieved the highest computational efficiency, with the lowest execution time of 0.000238 seconds and the highest throughput of 2,199,891 bits/s. PRESENT demonstrated the lowest memory consumption at 2.85 KB, whereas ASCON provided competitive performance but required substantially more memory resources than the other evaluated algorithms. Statistical analyses using 95% Confidence Intervals, One-Way ANOVA, and Tukey HSD revealed that significant differences among the algorithms were primarily associated with diffusion-related and computational performance metrics, while most randomness metrics showed no statistically significant differences.

Based on the calculated Security Score and Efficiency Score, PRESENT achieved the highest Security Score of 0.692, indicating slightly stronger diffusion and statistical security characteristics, whereas SPECK achieved the highest Efficiency Score of 0.993. Although PRESENT obtained the highest Security Score, the difference between PRESENT and SPECK was relatively small compared with the substantial computational efficiency advantage achieved by SPECK. Therefore, SPECK provides the most favorable balance between security-related characteristics and computational efficiency for resource-constrained MQTT-based IoT environments.

In addition, the results should be interpreted with consideration of the algorithm configurations used in this study. PRESENT was implemented using its standard 80-bit key configuration, whereas ASCON and SPECK employed 128-bit keys. Consequently, the results represent a comparison of the standard implementations of each algorithm rather than a comparison under identical key lengths. Future work may extend this evaluation by utilizing real IoT hardware platforms, varying payload sizes, incorporating additional lightweight cryptographic algorithms, and evaluating resistance against practical cryptanalytic attacks.

REFERENCES

[1] Iqbal R, Ansari NM, Awan M ur R, Ismail M, Gul H. Design and Evaluation of Lightweight Cryptographic Algorithms for Internet of Things (IoT) Devices: Achieving Optimal Trade-Offs Between Security, Computational Speed, and Energy Efficiency in Resource-Constrained Environments. THE PROGRESS: A Journal of Multidisciplinary Studies 2025;6:85–99. https://doi.org/10.71016/tp/smfybz24.

[2] K. Aruna Kumari. Cryptographic Algorithms and Computational Complexity: A Mathematical Approach to Securing IT Networks. Journal of Information Systems Engineering and Management 2025;10:409–20. https://doi.org/10.52783/jisem.v10i25s.4037.

[3] Turan MS, McKay KA, Chang D, Kang J, Kelsey J. Ascon-based lightweight cryptography standards for constrained devices : 2025. https://doi.org/10.6028/NIST.SP.800-232.

[4] Loke SV. Evaluation of Lightweight Cryptography Algorithms of National Institute of Standards and Technology: Structural Design, Security Assessment and Implementation Performance. 2025 IEEE International Conference on Computation, Big-Data and Engineering (ICCBE), IEEE; 2025, p. 502–7. https://doi.org/10.1109/ICCBE65177.2025.11256020.

[5] Dwivedi AD, Morawiecki P, Srivastava G. Differential Cryptanalysis of Round-Reduced SPECK Suitable for Internet of Things Devices. IEEE Access 2019;7:16476–86. https://doi.org/10.1109/ACCESS.2019.2894337.

[6] Alharbi S, Awad W, Bell D. HECS4MQTT: A Multi-Layer Security Framework for Lightweight and Robust Encryption in Healthcare IoT Communications. Future Internet 2025;17:298. https://doi.org/10.3390/fi17070298.

[7] Voloshyn V, Khan MS, Srivastava G, M D. Analysis of NIST Lightweight Cryptographic Algorithms Performance in IoT Security Environments based on MQTT. 2024 IEEE Wireless Communications and Networking Conference (WCNC), IEEE; 2024, p. 1–6. https://doi.org/10.1109/WCNC57260.2024.10571199.

[8] Abughazalah N, Khan M, Iqbal M. Hybrid multi-criteria decision-making technique for the selection of best cryptographic multivalued Boolean function. Complex & Intelligent Systems 2024;10:455–68. https://doi.org/10.1007/s40747-023-01150-z.

[9] Chatterjee R, Chakraborty R. NFBC: an efficient FPGA based NFSR-oriented lightweight block cipher suitable for embedded system. Sci Rep 2026;16:3651. https://doi.org/10.1038/s41598-025-33692-2.

[10] Mohamed K, Pauzi MNM, Ali FHHM, Ariffin S. Analyse On Avalanche Effect In Cryptography Algorithm, 2022, p. 610–8. https://doi.org/10.15405/epms.2022.10.57.

[11] Yavuzyiğit AE, Yayla O. Random Data and Its Cryptographic Applications, 2026, p. 140–55. https://doi.org/10.1007/978-3-032-10759-6_9.

[12] Pallangan JTK, Wowor AD. Identifikasi Nilai Keacakan Berdasarkan Reposisi Fungsi XOR Pada Blok Kedua LFSR A5/1. Journal of Information System Research (JOSH) 2024;6:679–87. https://doi.org/10.47065/josh.v6i1.6112.

[13] Radhakrishnan I, Jadon S, Honnavalli PB. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. Sensors 2024;24:4008. https://doi.org/10.3390/s24124008.

[14] Gladis Kurian M, Chen Y. The Untapped Potential of Ascon Hash Functions: Benchmarking, Hardware Profiling, and Application Insights for Secure IoT and Blockchain Systems. Sensors 2025;25:5936. https://doi.org/10.3390/s25195936.

[15] Mema A, Thomann S, Genssler PR, Amrouch H. FDSOI-Based Analog Computing for Ultra-Efficient Hamming Distance Similarity Calculation. IEEE Transactions on Circuits and Systems I: Regular Papers 2023;70:2679–88. https://doi.org/10.1109/TCSI.2023.3267837.

[16] Banu Y, Rath BK, Gountia D. Analyzing cryptographic algorithm efficiency with in graph-based encryption models. Front Comput Sci 2025;7. https://doi.org/10.3389/fcomp.2025.1630222.

[17] Agbangba CE, Sacla Aide E, Honfo H, Glèlè Kakai R. On the use of post-hoc tests in environmental and biological sciences: A critical review. Heliyon 2024;10:e25131. https://doi.org/10.1016/j.heliyon.2024.e25131.

[18] Nanda A, Mohapatra DrBB, Mahapatra APK, Mahapatra APK, Mahapatra APK. Multiple comparison test by Tukey's honestly significant difference (HSD): Do the confident level control type I error. International Journal of Statistics and Applied Mathematics 2021;6:59–65. https://doi.org/10.22271/maths.2021.v6.i1a.636.