

Article

A Software Platform for Benchmarking, Multi-Criteria Evaluation, and Integrity Validation of Symmetric Encryption Algorithms

Diyan Dinev *  and Gergana Spasova Faculty of Computer Sciences and Automation, Technical University of Varna, 9010 Varna, Bulgaria;
g.spasova@tu-varna.bg

* Correspondence: diyandinev@tu-varna.bg

Abstract

The choice of a symmetric encryption algorithm in practice is rarely as straightforward as it may appear from theoretical comparisons alone. In addition to security considerations, real-world selection often depends on execution time, reliability, entropy-related behavior, resource efficiency, and suitability for different types of data. This paper presents an experimental software platform for benchmarking and multi-criteria recommendation of symmetric encryption algorithms. The platform combines automated encryption and decryption tests, metric collection, comparative analysis, and result visualization within a unified evaluation workflow. It also incorporates a multi-criteria model that transforms raw experimental measurements into an overall ranking and supports context-aware recommendation according to the requirements of a given usage scenario. The experimental study includes repeated tests on different input categories in order to examine algorithm behavior under varied operating conditions. The obtained results show that algorithm performance and overall suitability are strongly dependent on the evaluation perspective and the application context, which suggests that no single symmetric method should be regarded as universally optimal. The proposed platform offers a practical basis for comparative cryptographic analysis and may be useful both for research purposes and for informed decision-making in security-oriented software environments.

Keywords: algorithm recommendation; context-aware selection; cryptographic benchmarking; cybersecurity; multi-criteria evaluation; performance analysis; symmetric encryption

Academic Editors: Danda B. Rawat
and Yangguang Tian

Received: 22 April 2026

Revised: 15 June 2026

Accepted: 23 June 2026

Published: 25 June 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In real deployments, the final choice of a symmetric encryption algorithm is also influenced by execution time, memory usage, energy demand, and platform constraints. This is particularly important in embedded and resource-constrained environments, where an algorithm that is secure in principle may still prove inefficient or impractical under real operating conditions [1,2].

This practical perspective has become more visible in recent years as cryptographic evaluation has moved beyond isolated timing measurements toward broader assessment of suitability in constrained and application-specific settings. Recent studies and surveys on lightweight cryptography consistently show that algorithm choice should be related to device limitations, workload characteristics, and deployment context rather than to nominal properties alone [3,4].

From a decision-support perspective, this makes algorithm selection a naturally multi-criteria problem. In practice, the decision rarely depends on one indicator only, because speed, implementation cost, resource efficiency, and operational suitability often need to be considered together. Recent work on multi-criteria decision-making in information systems supports the broader relevance of such structured selection models in technology-oriented contexts [5].

Against this background, the main limitation of the current literature is not the lack of benchmarking or review studies, but their fragmentation. Performance comparisons, lightweight cryptography surveys, and decision-support approaches are often presented separately. What remains less common is an integrated experimental environment that combines benchmarking, metric collection, comparative visualization, and multi-criteria recommendation for symmetric encryption algorithms. To address this gap, the present study proposes a software platform for context-aware evaluation and more informed selection of symmetric encryption algorithms.

The contribution of this paper is an integrated and reproducible workflow that connects controlled benchmarking, variability reporting, integrity-oriented checks, comparative visualization, and transparent WSM-based recommendation for a selected set of standard software-oriented symmetric encryption configurations. The resulting rankings are interpreted within the stated experimental conditions, workload profiles, and weighting assumptions, which provides a clear methodological boundary for the reported conclusions.

2. Related Work

A significant part of the recent literature has focused on benchmarking symmetric cryptographic algorithms in constrained and embedded environments. Gonza-lez-Arango et al. evaluated symmetric encryption algorithms on resource-constrained hardware for wireless sensor networks and showed that execution time and energy-related characteristics directly affect implementation feasibility [1]. Bühler et al. likewise studied symmetric cryptographic algorithms on an embedded ARM Cor-tex-M4 platform and reported clear runtime differences even under a controlled hardware setup [2]. Ramalho et al. extended this line of work to resource-constrained devices in an applied setting, while Kuznetsov et al. examined algorithm behavior in time-critical cybersecurity scenarios [4,6]. Together, these studies suggest that practical suitability depends strongly on the interaction between algorithm design, platform limitations, and operational priorities.

A closely related strand of research concerns lightweight cryptography for low-power systems and IoT-oriented deployments. Soto-Cruz et al. reviewed efficient lightweight cryptography for power-constrained microcontrollers and emphasized the importance of implementation-aware assessment [3]. Silva et al. later discussed lightweight encryption algorithms for IoT from a deployment-oriented perspective, while Sarker presented a systematic review focused on lightweight security algorithms for sustainable IoT infrastructures [7,8]. Related review work by Gușiță et al., Ansari and Ali, and Zinabu et al. further reinforced the need to evaluate lightweight cryptographic methods through latency, throughput, memory use, energy consumption, and applicability to battery-limited devices [9–11].

Recent experimental papers have also moved toward concrete platform-based validation. Sorescu et al. compared lightweight algorithms on resource-constrained IoT platforms and reported substantial performance differences across throughput and energy scenarios. Sarasa Laborda et al. investigated the performance of ASCON on Arduino devices, while Cagua et al. examined ASCON-based authenticated encryption on IoT devices [12–14]. These studies are particularly relevant because they show that even within

the lightweight cryptography landscape, implementation behavior varies substantially depending on hardware profile and communication context.

Another important direction concerns standardization and implementation of maturity. The NIST lightweight cryptography process culminated in the selection and standardization of the ASCON family for constrained devices, which strengthened the broader shift toward context-aware cryptographic evaluation [15,16]. In parallel, implementation-oriented analyses such as the review by Konstantopoulou et al. examined FPGA and ASIC realizations of NIST lightweight cryptography finalists and highlighted the continuing importance of implementation trade-offs even after standardization [17].

Alongside benchmarking and implementation studies, multi-criteria approaches have gained increasing relevance in decision-support research. Madanchian and Tahdoost reviewed the application of MCDM methods in information systems and showed their value for structured strategic and operational technology selection [5]. Avramova et al. provided a broader overview of MCDM methods used in industrial environments, which is methodologically useful for technology evaluation problems involving competing indicators [18]. In a more cryptography-specific context, Abughazalah et al. proposed a hybrid multi-criteria technique for selecting cryptographic multivalued Boolean functions, and Khan later introduced an intelligent decision-making framework for ranking optimized substitution boxes under multiple cryptographic criteria [19,20].

Recent survey literature has also broadened the comparative picture at the cipher design level. Dewamuni et al. provided a bibliometric analysis of IoT lightweight cryptography and mapped the main research clusters in the field, while Al-Nofaie et al. presented a comparative survey of lightweight block ciphers and categorized recent designs according to their structural properties and deployment relevance [21,22].

From a tooling perspective, several general-purpose benchmarking utilities exist for cryptographic performance measurement, including the OpenSSL speed command-line tool, the CrypTool educational platform, and language-level microbenchmarking frameworks such as Google Benchmark and JMH. While these tools are effective for measuring raw execution speed, they do not natively support multi-criteria scoring, context-aware recommendation, or integrity-oriented validation. They also lack built-in support for combining timing, resource, entropy, and overhead indicators within a single structured workflow. The platform proposed in this study is positioned to address this gap by providing an integrated environment that goes beyond isolated performance measurement toward decision-support functionality.

Taken together, the literature establishes three consistent points. First, comparative evaluation of symmetric and lightweight algorithms is increasingly context-dependent. Second, practical selection cannot be reduced to a single metric when several partially competing criteria must be balanced. Third, despite the growing number of benchmarking studies and surveys, there is still limited support for unified software platforms that connect experimental benchmarking with multi-criteria interpretation and recommendation logic for complete symmetric encryption algorithms. This is the contribution area in which the present study is positioned.

This positioning clarifies the role of the lightweight cryptography literature in the study. The cited work motivates context-aware evaluation and highlights the need for implementation-dependent metrics, while the empirical part of the present paper focuses on AES-256-GCM, ChaCha20-Poly1305, and AES-256-CTR+HMAC in a software execution environment. Lightweight algorithms such as ASCON define a coherent extension path for the platform and are therefore discussed as part of the broader research context.

3. System Design and Architecture

The proposed platform was designed as an experimental software environment for the comparative evaluation of symmetric encryption algorithms under varied practical conditions. Its main purpose is not limited to executing cryptographic operations but extends to the systematic collection of performance-related indicators, the structured comparison of results, and the generation of context-aware recommendations. In this sense, the platform functions both as a benchmarking tool and as a decision-support environment for algorithm selection.

From an architectural perspective, the system follows a modular design in which the major tasks of the evaluation process are separated into clearly defined functional components. This modularity was adopted to improve the extensibility, reproducibility, and clarity of experimental control. At a high level, the platform includes modules for input handling, algorithm selection, cryptographic execution, metric acquisition, result aggregation, visualization, and recommendation. Figure 1 presents the general architecture of the proposed platform and illustrates the main functional modules involved in data configuration, cryptographic execution, metric collection, result processing, visualization, and multi-criteria recommendation.

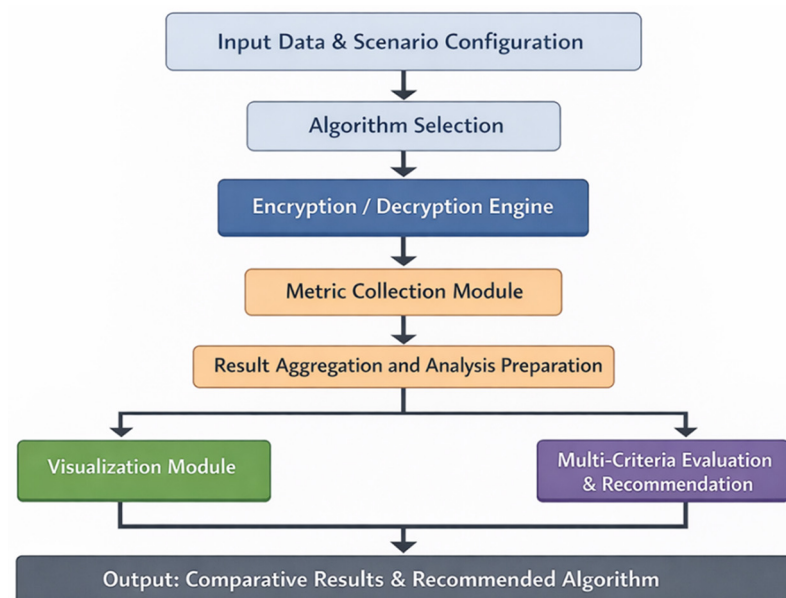


Figure 1. General architecture of the proposed platform for benchmarking and multi-criteria recommendation of symmetric encryption algorithms.

The workflow begins with the preparation of the experimental input. The user provides or selects the test data, defines the execution scenario, and chooses the algorithms to be included in the experiment. The platform is intended to support evaluation across different categories of input data, since practical algorithm behavior may depend not only on the cryptographic method itself, but also on the characteristics of the processed content. Once the configuration is defined, the selected algorithms are executed under the same experimental conditions in order to ensure comparability of the collected results.

The cryptographic execution layer is responsible for performing encryption and decryption operations and for applying a consistent testing procedure across all selected algorithms. This layer constitutes the experimental core of the platform. It ensures that each algorithm is evaluated through the same operational sequence, which is essential for the validity of comparative analysis. In addition to functional correctness, the execution stage enables the platform to observe algorithm behavior from several practical perspectives, including runtime characteristics, processing stability, and output-related properties.

A separate metric collection layer records the quantitative indicators generated during testing. Rather than relying on a single measure, the platform is designed to capture a broader set of variables that can later be used for comparative interpretation. These include performance-oriented indicators, reliability-related observations, entropy-linked characteristics, and selected resource-related measures. The role of this layer is especially important because it transforms the execution process into structured experimental evidence that can be reused in later stages of ranking and recommendation.

After metric acquisition, the measured data are transferred to a result-processing stage in which they are aggregated and prepared for further analysis. This stage serves as the analytical bridge between raw measurements and higher-level interpretation. It enables the organization of experimental outputs into comparable structures that can support both visual inspection and multi-criteria assessment.

The practical realization of these functions can be seen in the main interface of the system. Figure 2 shows the central workspace of the platform, including the settings panel, the status and methodology sections, and the comparative results table used for algorithm-level inspection. This interface organization was designed to support both exploratory use and structured experimental analysis, allowing the user to move from benchmark configuration to result interpretation within a single environment.

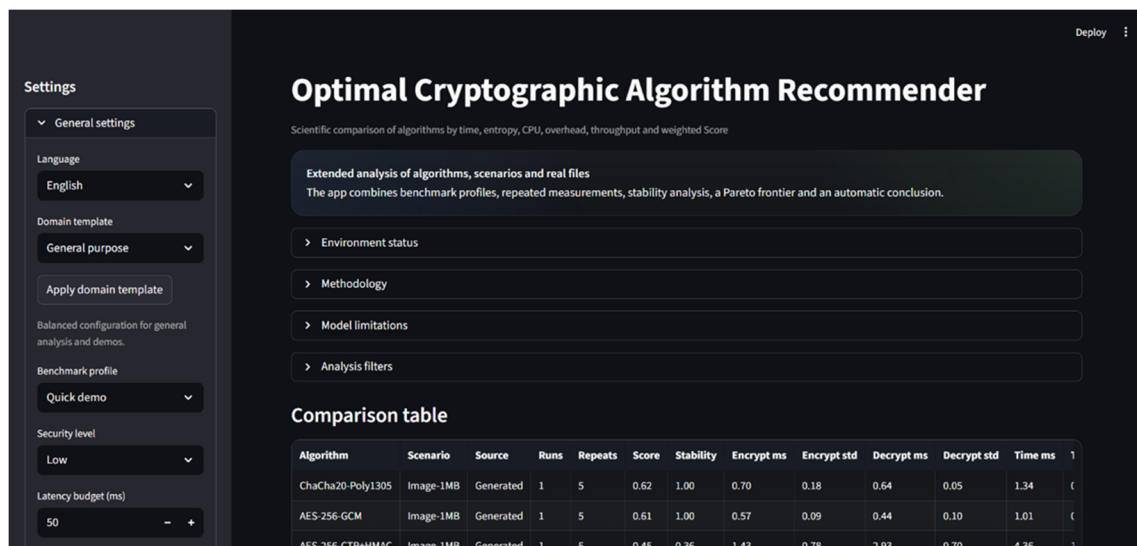


Figure 2. Main interface of the proposed benchmarking and recommendation platform.

The processed results are then used by the visualization module and by the multi-criteria evaluation and recommendation component. The visualization module is responsible for presenting the collected data in a form that facilitates interpretation through comparative tables, charts, and heatmap-style views. This component is not treated as a purely auxiliary feature, since visual analysis plays an important role in identifying differences in algorithm behavior across metrics and scenarios.

In parallel, the recommendation-oriented component transforms the aggregated experimental evidence into a more interpretable ranking of alternatives according to the requirements of a given scenario. In this way, the system supports a more informed selection process rather than merely reporting raw benchmark values. The recommendation mechanism is therefore conceptually linked to the multi-criteria model described later in this paper, while the present section focuses on its architectural role within the overall platform.

The configuration capabilities of the platform are illustrated in Figure 3, which presents the main control panels related to general settings, algorithm selection, work-load definition,

real-file input, priority selection, criterion weighting, and execution management. These interface components allow the experimental process to be configured in a structured and reproducible manner across different benchmarking scenarios. In particular, the presence of explicit priority and weight controls reflects the direct integration of the recommendation logic into the user-facing experimental workflow.

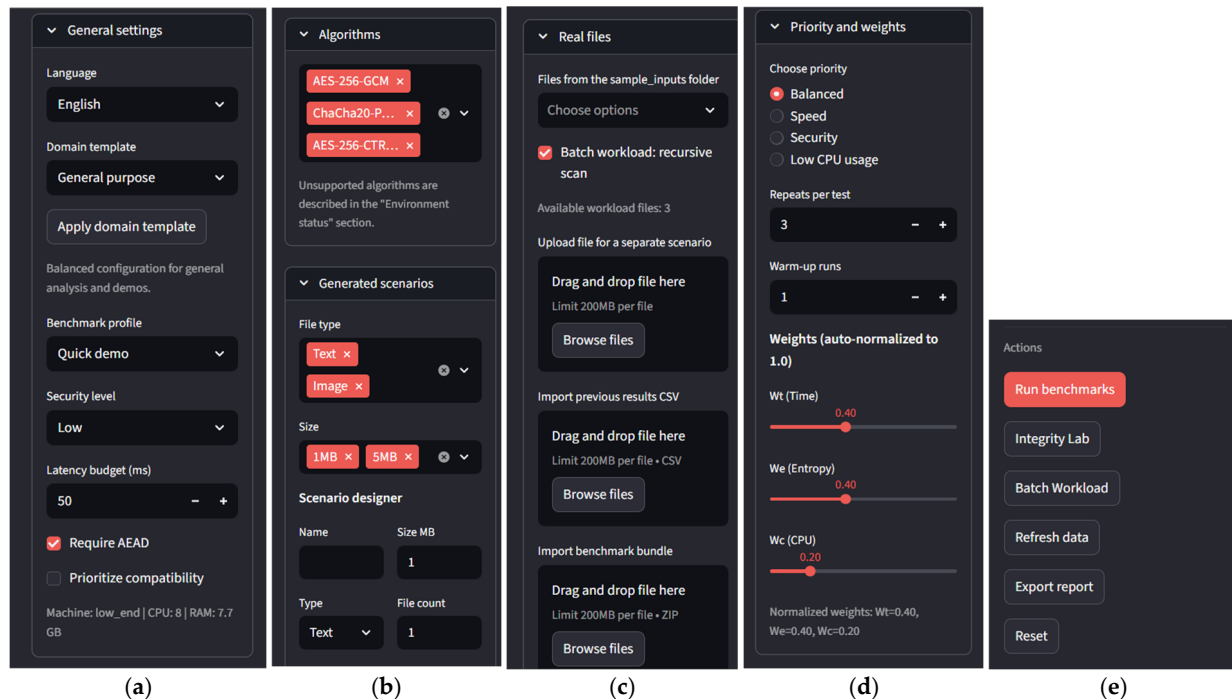


Figure 3. Configuration panels of the proposed benchmarking platform: (a) general settings; (b) algorithm selection and generated scenarios; (c) real-file and benchmark-bundle input options; (d) priority profile and criterion weights; (e) benchmark execution and report export actions.

An important design principle of the proposed system is reproducibility. The platform is intended to support repeated experiments under controlled settings, so that observed differences can be attributed as reliably as possible to the evaluated algorithms and not to inconsistent testing procedures. For this reason, the architecture is organized around a repeatable experimental cycle consisting of configuration, execution, metric collection, aggregation, visualization, and recommendation.

Overall, the architecture of the platform reflects the central idea of this study: symmetric encryption algorithms should be evaluated within a unified environment that connects implementation, measurement, analysis, and selection support. By integrating these stages into a single workflow, the platform provides a practical basis for comparative cryptographic experimentation and creates the structural foundation for the multi-criteria evaluation model discussed in the following sections.

4. Supported Algorithms and Evaluation Metrics

The platform differentiates standard software-oriented configurations from lightweight and constrained-device algorithms. For the former, the present evaluation emphasizes throughput, latency, implementation overhead, and fail-safe behavior under misuse-oriented tests; for the latter, energy consumption, code size, RAM footprint, and device-specific latency budgets are more central evaluation criteria. The current version of the platform supports three symmetric encryption configurations: AES-256-GCM, ChaCha20-Poly1305, and AES-256-CTR+HMAC. These alternatives were selected to represent two authenticated encryptions with associated data (AEAD) schemes and one construction

based on counter-mode encryption combined with an external message authentication mechanism. This selection makes it possible to compare algorithms that differ not only in implementation characteristics, but also in the way confidentiality and integrity protection are integrated within the cryptographic workflow. This focused selection enables the experimental protocol to report repeated measurements, variability, and integrity-oriented checks before the platform is extended to a broader set of algorithms.

AES-256-GCM was included as a widely adopted AEAD configuration that combines counter-mode encryption with Galois/Counter Mode authentication. In the context of the present platform, it serves as a strong reference point for high-assurance authenticated encryption with broad practical relevance. ChaCha20-Poly1305 was selected as a second AEAD alternative with a different internal design philosophy and a strong reputation for software efficiency, especially in environments where implementation characteristics may differ from those of AES-based approaches. AES-256-CTR+HMAC, in turn, was included as a representative construction in which encryption and authentication are handled through separate mechanisms. Its inclusion is useful because it allows the platform to compare integrated AEAD schemes with a composition-based approach that may behave differently in terms of overhead, timing, and integrity-related processing. The platform architecture supports the addition of further algorithms and configurations in future versions.

The platform evaluates these algorithms across different input categories in order to reflect more realistic usage conditions. In the current experimental design, the workload includes text, image, and binary data, with representative file sizes such as 1 MB, 5 MB, and 10 MB. This structure is important because algorithm behavior may vary not only with the selected cryptographic method, but also with the size and nature of the processed data. The use of multiple source categories therefore improves the practical relevance of the comparison and allows the system to observe whether certain algorithms behave more favorably under particular workload profiles.

To provide a clearer overview of the analytical scope of the platform, Table 1 summarizes the supported encryption algorithms, the main categories of input data, the workload sizes considered in the experiments, and the principal evaluation metrics and validation checks used in the current implementation.

Table 1. Main analytical scope of the proposed platform.

Category	Elements
Algorithms	AES-256-GCM; ChaCha20-Poly1305; AES-256-CTR+HMAC
Input categories	Text; Image; Binary
Workload sizes	1 MB; 5 MB; 10 MB 1 KB; 16 KB; 64 KB; 10 MB chunked streaming scenario
Performance metrics	encrypt_ms; decrypt_ms; time_ms; throughput_mbps
Resource metrics	cpu_pct; memory_mb
Output-related metrics	size_bytes; overhead_bytes; entropy
Validation checks	Correct decryption; corrupted ciphertext; wrong key; wrong nonce/IV; tampering detection; authenticate before release verification for the CTR+HMAC implementation
Calibration	Idle-baseline and empty-run correction for CPU and memory observations

As shown in Table 1, the platform combines algorithm-level comparison with workload diversity, performance-oriented indicators, and integrity-related validation checks. This combination is important because it allows the subsequent evaluation model to rely on a broader and more practically relevant evidence base rather than on isolated timing results alone.

To ensure comparability, each run records a set of quantitative indicators that describe both performance and implementation-related behavior. The primary timing metrics are encryption time, decryption time, and total processing time, represented in the platform by the variables `encrypt_ms`, `decrypt_ms`, and `time_ms`. These metrics provide the most direct view of runtime efficiency and are central to any comparative benchmark of cryptographic algorithms. The separation between encryption and decryption is especially useful because some methods may exhibit asymmetric behavior across these two stages, while the combined processing time provides a compact basis for ranking and scenario-oriented recommendation. Similar timing variables are also used in recent data-protection studies where encryption time is analyzed together with broader system performance indicators [23].

In addition to timing, the platform measures ciphertext entropy, recorded as `entropy`. This metric is used as an indicator of output randomness and supports a broader interpretation of ciphertext characteristics beyond raw execution speed. Although entropy cannot substitute for formal cryptographic security analysis, it remains a useful descriptor for comparing output behavior under the same test conditions.

The platform also captures CPU utilization and memory consumption, represented by `cpu_pct` and `memory_mb`. These indicators are especially relevant in the evaluation of algorithms intended for constrained, low-power, or implementation-sensitive environments. In practical deployment, an algorithm that performs well in terms of time alone may still be less desirable if it imposes higher computational or memory-related costs. For this reason, resource-related indicators form an essential part of the multi-dimensional assessment used by the proposed system. The protocol incorporates idle-baseline and empty-run calibration values so that CPU and memory observations are interpreted after accounting for platform and measurement-layer overhead.

Another group of metrics concerns size-related characteristics. The platform records the original plaintext size, the resulting output size (`size_bytes`), and the associated overhead (`overhead_bytes`). These variables are useful because cryptographic processing often introduces metadata, tags, nonces, IVs, or authentication-related expansion that may affect storage efficiency and transmission cost. In applications involving repeated encryption of many objects or bandwidth-sensitive exchange, such overhead is not negligible and should therefore be considered as part of the algorithm selection process.

To complement the timing and size metrics, the system also computes throughput, recorded as `throughput_mbps`. This indicator provides a normalized perspective on processing efficiency by relating data volume to execution time. Throughput is particularly useful when comparing algorithms across workloads of different sizes, since it makes the results easier to interpret in terms of sustained processing capacity rather than isolated time values.

Beyond standard benchmarking indicators, the platform also incorporates correctness- and integrity-oriented validation. At the functional level, algorithm execution is expected to preserve correct decryption under normal conditions. In addition, the current test suite includes fail-safe integrity checks such as corrupted ciphertext handling, wrong-key behavior, and wrong nonce or IV behavior for authenticated modes, as well as tampering detection for the AES-256-CTR+HMAC configuration. These elements are important because the goal

of the platform is not only to compare algorithms by speed, but also to evaluate whether they behave in a reliable and security-aware manner under adverse conditions.

Taken together, the supported algorithms and the selected evaluation metrics define a focused analytical scope for the platform. The current implementation evaluates a well-defined and practically relevant set of alternatives through timing, entropy, resource, size, throughput, and integrity-related indicators. This combination provides an appropriate basis for the multi-criteria evaluation model presented in the next section.

5. Multi-Criteria Evaluation Model

The proposed platform frames algorithm comparison as a multi-dimensional evaluation task. It adopts a multi-criteria evaluation model intended to transform heterogeneous experimental results into a more interpretable overall assessment. This design choice follows directly from the practical motivation of the study: in realistic deployment scenarios, the suitability of a symmetric encryption algorithm depends on the joint effect of performance, resource efficiency, output-related characteristics, and operational reliability.

For this reason, the platform aggregates the collected experimental indicators into a unified evaluation framework. The purpose of the model is not to replace the raw benchmark results, but to complement them by providing a structured basis for ranking and recommendation. In this way, the system supports both levels of interpretation: direct inspection of the measured values and higher-level comparative assessment based on multiple criteria considered simultaneously.

At the conceptual level, the evaluation model consists of four main stages: criterion definition, normalization of the measured indicators, weighting of the criteria, and calculation of an overall score for each algorithm. The result of this process is a ranked list of alternatives that can later be interpreted in relation to a specific usage scenario. The implemented decision procedure is the Weighted Sum Model (WSM), selected because it is transparent, reproducible, and easy to audit against the displayed normalized criteria and user-defined weights.

The first stage is the definition of the evaluation criteria. In the current implementation, the model is based on several groups of indicators derived from the benchmarking process. These include timing-related measures, such as encryption time, decryption time, and total processing time; resource-related measures, such as CPU usage and memory consumption; size-related indicators, such as ciphertext size and overhead; throughput; and selected output- or reliability-related measures, including entropy and integrity-oriented validation behavior. This grouping reflects the fact that the practical quality of an algorithm emerges from the interaction of several partially competing characteristics rather than from a single dominant metric.

Because these indicators are measured in different units and ranges, direct comparison is not meaningful without transformation. The second stage of the model therefore applies normalization in order to place the criteria on a comparable scale. In general, metrics for which higher values are preferable are treated as benefit criteria, whereas metrics for which lower values are preferable are treated as cost criteria.

For a benefit criterion, the normalized value n_{ij} of algorithm i under criterion j is computed as shown in Equation (1), where x_{ij} is the original measured value of algorithm i for criterion j , and $\min(x_j)$ and $\max(x_j)$ are the minimum and maximum observed values for that criterion across all compared algorithms.

$$n_{ij} = (x_{ij} - \min(x_j)) / (\max(x_j) - \min(x_j)) \quad (1)$$

For a cost criterion, the normalization is inverted so that lower raw values receive higher normalized scores:

$$n_{ij} = (\max(x_j) - x_{ij}) / (\max(x_j) - \min(x_j)) \quad (2)$$

Through this transformation, all criteria are mapped to the interval $[0, 1]$, where higher normalized values always indicate more favorable performance from the perspective of the evaluation model. This step is essential because it allows criteria expressed in different units, such as milliseconds, megabytes, bytes, or throughput, to participate in a common scoring procedure without distortion caused by scale differences.

To clarify how the main indicators are interpreted within the proposed evaluation framework, Table 2 summarizes the principal criteria included in the multi-criteria model and classifies them according to whether they are treated as cost or benefit criteria.

Table 2. Main evaluation criteria and their interpretation in the multi-criteria model.

Criterion	Type	Interpretation
Encrypt ms	Cost	Lower values indicate better encryption efficiency.
Decrypt ms	Cost	Lower values indicate better decryption efficiency.
Time ms	Cost	Lower values indicate better overall processing efficiency.
Cpu pct	Cost	Lower values indicate lower processor utilization during execution.
Memory mb	Cost	Lower values indicate lower memory consumption.
Overhead bytes	Cost	Lower values indicate lower cryptographic expansion caused by metadata, tags, nonces, or authentication-related elements.
size_bytes	Cost	Lower values indicate smaller output size after cryptographic processing.
throughput	Benefit	Higher values indicate better sustained processing capacity.
entropy	Benefit	Higher values indicate stronger random-related characteristics of the produced ciphertext.
integrity validation	Benefit	Higher values indicate more reliable behavior under integrity-related verification and tampering-detection conditions.

As shown in Table 2, the proposed model combines performance-oriented, resource-related, size-related, and reliability-oriented indicators within a common evaluative structure. This classification is important because it determines how the normalization procedure is applied and how each criterion contributes to the final weighted score.

After normalization, the model assigns weights to the selected criteria. The purpose of weighting is to reflect the relative importance of the indicators under a given evaluation perspective. This is an essential part of the platform because the preferred algorithm may change depending on the practical scenario. For example, in a performance-oriented scenario, timing and throughput indicators may receive greater importance. In a resource-constrained setting, memory use, CPU load, and overhead may be weighed more strongly. In a more balanced scenario, the model may distribute importance more evenly across execution efficiency, resource impact, and reliability-related behavior. The platform is therefore designed to support a flexible weighting strategy rather than a single fixed ranking logic.

Once normalized values and criterion weights are available, the platform computes an overall score for each algorithm. Let w_j denote the weight assigned to criterion j , with the condition that the sum of all weights is equal to 1. The overall score S_i of algorithm i is then calculated as shown in Equation (3), where m is the total number of selected evaluation

criteria. This score represents the weighted aggregation of the normalized performance of algorithm i across all criteria included in the scenario.

$$S_i = \sum_{j=1}^m w_j n_{ij} \quad \text{where } \sum_{j=1}^m w_j = 1 \quad (3)$$

A higher final score corresponds to a more favorable overall evaluation under the chosen scenario. The score expresses the weighted suitability of an alternative within the experimental conditions, selected metrics, and adopted weighting profile. It is therefore used as a scenario-specific decision-support indicator rather than as an intrinsic measure of cryptographic strength.

The ranking stage follows directly from the computed overall scores. Algorithms are ordered according to their final evaluation values, which allows the system to generate a scenario-sensitive recommendation. This recommendation mechanism is one of the key practical elements of the platform, since it transforms raw experimental data into an interpretable decision-support output. Instead of leaving the user with multiple disconnected tables and charts, the model provides a structured way to identify the alternative that best matches the intended operational priorities.

An important feature of the proposed model is that it remains transparent and explainable. The recommendation is not produced by an opaque heuristic, but by a traceable procedure in which the user can identify the contributing criteria, the normalization logic, the assigned weights, and the resulting score structure. This transparency is important both methodologically and practically. From a research perspective, it improves reproducibility and interpretability. From an application perspective, it allows the user to understand why a given algorithm is preferred in one scenario but not in another.

A sensitivity consideration is also relevant here. Because the score differences between the top-ranked algorithms are relatively small—on the order of 0.01 to 0.02 units in the present results—the ranking may be sensitive to the choice of weights. To illustrate this, if the weight assigned to timing-related criteria is reduced and the weight of resource or overhead criteria is increased beyond the values defined in the resource-aware profile, ChaCha20-Poly1305 can approach or match the score of AES-256-GCM on non-AES-NI hardware. This means that the recommendation produced by the model is best understood as a context-specific outcome rather than an absolute ordering. Users of the platform are encouraged to adjust the weighting profile according to their specific deployment priorities and to treat score differences below 0.05 as indicative rather than conclusive.

The model is designed for scenario-bounded interpretation. Its usefulness depends on the relevance of the selected criteria and the realism of the adopted weighting scheme. By making these assumptions explicit, the framework supports transparent comparison and reduces reliance on intuitive or single-metric selection.

Overall, the multi-criteria evaluation model provides the analytical core that connects benchmarking with recommendation. By transforming raw measurements into normalized, weighted, and interpretable scores, it enables the platform to move from descriptive performance comparison toward practically useful algorithm selection. This model is subsequently applied in the experimental part of the study to compare the supported algorithms across the defined workload scenarios.

6. Experimental Setup

The experimental study was designed to evaluate the supported symmetric encryption algorithms under controlled and repeatable conditions. Its purpose was to generate sufficiently broad evidence base for comparative benchmarking, multi-criteria scoring, and scenario-oriented recommendation. For this reason, the setup was structured to in-

clude different categories of input data, several workload sizes, repeated executions, and a consistent measurement procedure across all tested algorithms.

All experiments were conducted within the software environment of the proposed platform, using the same execution workflow for each algorithm to preserve comparability. The benchmark process followed a unified sequence consisting of input preparation, algorithm execution, metric collection, result aggregation, and result export. This procedure was intended to ensure that the observed differences could be attributed as far as possible to the evaluated algorithms rather than to inconsistencies in the testing process.

The evaluated algorithm set included AES-256-GCM, ChaCha20-Poly1305, and AES-256-CTR+HMAC. These three configurations were examined under the same benchmarking logic and across the same principal workload categories. Their inclusion allowed the platform to compare both integrated authenticated-encryption schemes and a composition-based approach combining encryption and message authentication as separate operations.

The workload design included three principal input categories: text, image, and binary data. These categories were selected in order to reflect different practical content types that may occur in real software usage. Text data represent structured and relatively lightweight content, image data introduce media-oriented characteristics, and binary inputs represent a broader class of general-purpose file content. The use of multiple input categories reduced the risk of drawing conclusions from a single workload type only.

In addition to data type variation, the experiments were conducted with several representative input sizes. In the main benchmark scenarios, workloads were defined for 1 MB, 5 MB, and 10 MB inputs. This range made it possible to observe whether algorithm behavior remained stable across increasing data volume or whether performance differences became more pronounced as the size of the processed input grew. The main experimental configuration used in the comparative evaluation is summarized in Table 3.

To improve measurement reliability, the benchmark runs were repeated rather than executed only once per scenario. In the adopted benchmark configuration, each scenario was evaluated through 30 measured runs, with five warm-up runs used before the main repeated measurements. The use of repetition was intended to reduce the influence of incidental fluctuations caused by environment initialization, caching behavior, or transient system activity. For the statistical summary, mean values, standard deviations, and approximate 95% confidence intervals were exported for the main timing and throughput indicators.

During each run, the platform recorded the full set of benchmarking indicators used by the study. These included encryption time, decryption time, total processing time, ciphertext entropy, CPU utilization, memory consumption, output size, cryptographic overhead, and throughput. The combined use of these metrics was intended to support both raw comparative analysis and the later multi-criteria evaluation stage.

In addition to performance-oriented benchmarking, the experimental design also included integrity- and reliability-related validation checks. For each supported algorithm, the platform verified successful decryption under normal conditions and also tested selected adverse cases such as wrong-key execution, corrupted ciphertext, and invalid nonce or IV conditions, where applicable. For the AES-256-CTR+HMAC configuration, explicit tampering-detection checks were also included. These tests were not intended as formal cryptanalytic study, but as practical validation steps capturing security-relevant execution outcomes in addition to speed and resource behavior.

To support the recommendation logic of the platform, the experimental results were additionally interpreted through three scenario-based decision-profile: a performance-oriented profile, a resource-aware profile, and a balanced profile. These profiles were defined to reflect different practical priorities in algorithm selection. The performance-oriented profile assigned greater importance to timing and throughput indicators, the

resource-aware profile emphasized reduced processor, memory, and overhead impact, and the balanced profile distributed more evenly across the selected criteria. The weights used in the multi-criteria evaluation are summarized in Table 4.

Table 3. Experimental setup used in the comparative evaluation.

Component	Setting	Description
Algorithms	AES-256-GCM; ChaCha20-Poly1305; AES-256-CTR+HMAC	Symmetric encryption configurations included in the main benchmark analysis.
Input categories	Text; Image; Binary	Data types used to represent different workload characteristics.
Input sizes	1 MB; 5 MB; 10 MB; 1 KB; 16 KB; 64 KB; 10 MB chunked streaming scenario	Representative workload sizes used in the main experiments. Additional micro-payload and chunked-streaming scenarios used to address latency-sensitive and streaming-oriented use cases.
Repetitions	30 measured runs per scenario	Repeated execution used to improve measurement reliability and enable variability reporting.
Warm-up	5 preliminary runs	Warm-up sequence used before the measured benchmark runs.
Recorded metrics	encrypt_ms; decrypt_ms; time_ms; entropy; cpu_pct; memory_mb; size_bytes; overhead_bytes; throughput_mbps	Main performance, resource, and output-related indicators collected during benchmarking.
Validation checks	Correct decryption; wrong key; corrupted ciphertext; wrong nonce/IV; tampering detection	Reliability-oriented checks applied to normal and adverse execution conditions.
Decision profiles	Performance-oriented; resource-aware; balanced	Scenario-based profiles used in the recommendation analysis.
Calibration	Idle baseline; empty-run baseline	Baseline measurements used to interpret CPU and memory observations after subtracting platform overhead.

Table 4. Scenario-based weighting profiles used in the multi-criteria evaluation.

Criterion	Performance-Oriented	Resource-Aware	Balanced
encrypt_ms	0.20	0.10	0.12
decrypt_ms	0.20	0.10	0.12
time_ms	0.20	0.10	0.12
throughput_mbps	0.20	0.10	0.12
cpu_pct	0.05	0.20	0.10
memory_mb	0.05	0.20	0.10
overhead_bytes	0.05	0.15	0.10
size_bytes	0.00	0.05	0.07
entropy	0.03	0.00	0.07
integrity validation	0.02	0.05	0.08
Total	1.00	1.00	1.00

The experiments were conducted on a controlled hardware platform equipped with an Intel Core i5-1135G7 processor, 8 GB DDR4 RAM, and Windows 10 Pro (64-bit). The

availability of AES-NI support is relevant in this context, since it allows the AES-256-GCM configuration to benefit from hardware-level acceleration during both encryption and authentication-related processing. As a result, the timing and throughput advantage of AES-256-GCM observed in this study reflects a combination of algorithmic and hardware factors specific to AES-NI-enabled x86 platforms. On platforms without AES-NI support—such as ARM Cortex-M or RISC-V systems—ChaCha20-Poly1305 is known to be highly competitive with or superior to AES-GCM in software execution. All conclusions regarding algorithm ranking should therefore be interpreted within this hardware context.

Before each benchmark batch, the platform recorded an idle baseline and an empty-run baseline without cryptographic processing. These values were used to check whether the measured CPU and memory values were dominated by the tested algorithm or by the interface and measurement layer. The reported resource indicators are adjusted for baseline platform activity and should therefore be interpreted as platform-level estimates rather than hardware-independent properties of the algorithms.

The weighting scheme in Table 4 reflects the role of each criterion within the corresponding decision profile. Entropy receives minimal or zero weight in profiles where it adds little discriminative information, because all evaluated configurations produced nearly identical entropy values. Integrity validation is retained as a low-weight criterion across the profiles to keep fail-safe behavior visible in the ranking, while the platform also allows it to be handled as a threshold condition in custom weighting configurations.

7. Results

The results presented in this section focus on the principal generated benchmark scenarios defined by data type and file size. Accordingly, the comparative analysis is based on the text, image, and binary workloads with input sizes of 1 MB, 5 MB, and 10 MB. These results are discussed from four complementary perspectives: raw benchmarking behavior, comparative visualization and normalized interpretation, ranking under different decision profiles, and integrity-related validation.

7.1. Benchmarking Results

The first level of analysis concerns direct comparison of the measured benchmarking indicators. Average results across the principal generated benchmark scenarios are summarized in Table 5, while the main timing and throughput trends are illustrated in Figures 4 and 5.

Across the evaluated workloads, AES-256-GCM showed the strongest overall runtime performance, with the lowest average encryption, decryption, and total processing times. On average, it achieved 2.99 ms for encryption, 3.06 ms for decryption, and 6.05 ms in total. ChaCha20-Poly1305 followed with 4.14 ms, 4.18 ms, and 8.32 ms, respectively, while AES-256-CTR+HMAC showed the highest overall processing cost, particularly in decryption, with 5.92 ms, 14.17 ms, and 20.09 ms.

Table 5. Average benchmarking results by algorithm across the principal generated scenarios.

Algorithm	Encrypt Time (ms)	Decrypt Time (ms)	Total Time (ms)	Throughput (Mbps)	CPU (%)	Memory (MB)	Overhead (Bytes)	Entropy
AES-256-GCM	2.99	3.06	6.05	913.76	9.21	11.34	28	7.9999
ChaCha20-Poly1305	4.14	4.18	8.32	596.60	10.54	11.34	28	7.9999
AES-256-CTR+HMAC	5.92	14.17	20.09	241.81	13.12	15.50	44	7.9999

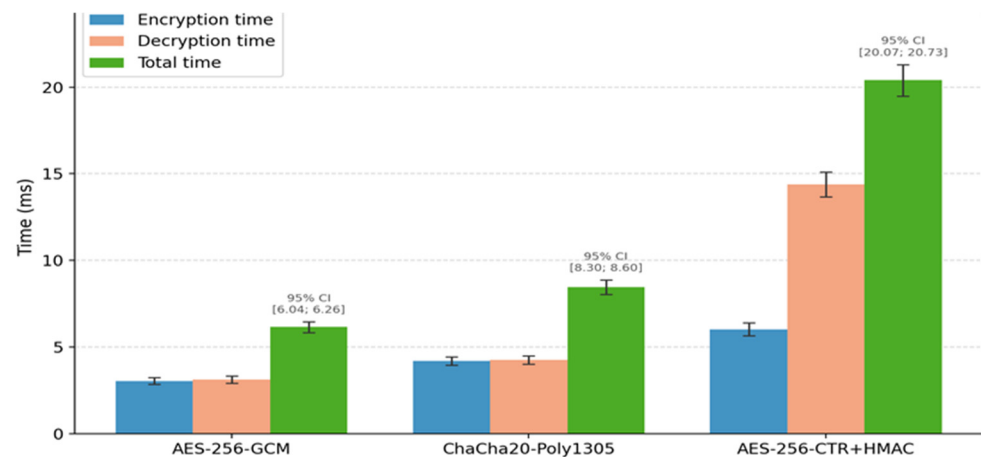


Figure 4. Thirty-run timing summary by algorithm.

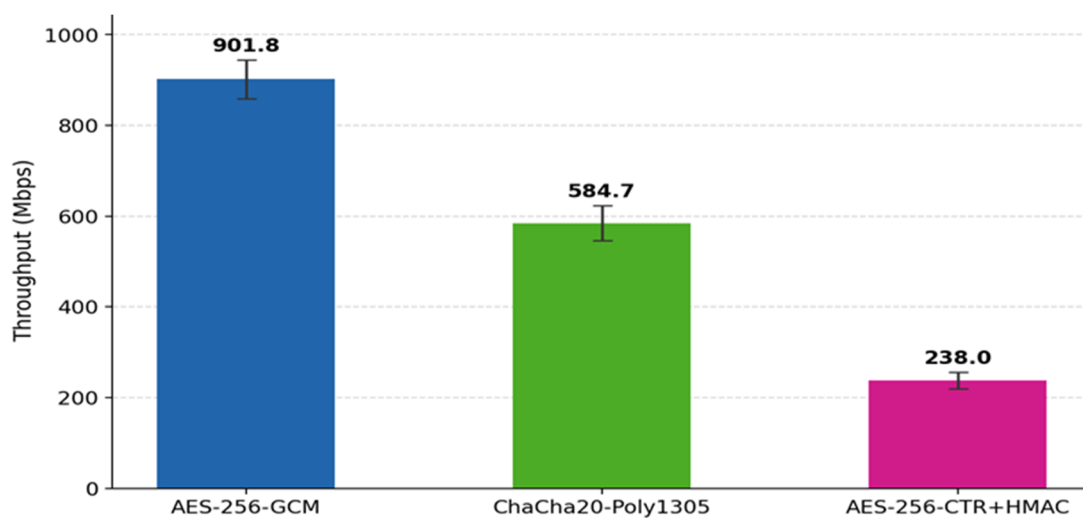


Figure 5. Thirty-run throughput summary by algorithm.

The throughput results follow the same general pattern. AES-256-GCM achieved the highest average throughput at approximately 913.76 Mbps, followed by ChaCha20-Poly1305 with 596.60 Mbps, whereas AES-256-CTR+HMAC remained substantially lower at 241.81 Mbps. This difference becomes especially visible for the larger workloads, as illustrated in Figure 5.

Although the absolute values vary with workload size and content type, the relative ordering of the algorithms remains stable across the principal scenarios. AES-256-GCM consistently remains the fastest option, ChaCha20-Poly1305 occupies an intermediate position, and AES-256-CTR+HMAC imposes the highest overall cost. The gap is particularly visible in the decryption phase, where the composition-based design of AES-256-CTR+HMAC introduces a greater processing burden.

The ordering is linked to the hardware environment used in the experiment. The Intel Core i5-1135G7 processor includes AES-NI support, which provides hardware-level acceleration for AES-based operations and directly benefits AES-256-GCM during both the counter-mode encryption step and the Galois/Counter Mode authentication step. On platforms without AES-NI, or under software-only implementations, the relative position of AES-256-GCM and ChaCha20-Poly1305 may change because ChaCha20-Poly1305 is specifically designed for efficient software execution without dedicated hardware instructions. The reported throughput and latency advantage of AES-256-GCM therefore characterizes

the evaluated hardware and software configuration and motivates separate validation for software-only, mobile, or constrained environments.

From the perspective of output-related indicators, the entropy values remain very close across all three algorithms (7.9999 in all cases) and are therefore not discriminative in the present experiment. Since all three algorithms produce outputs, whose entropy is indistinguishable at this precision, entropy does not contribute meaningfully to the multi-criteria ranking in this evaluation. It is retained as a metric because it would become discriminative if algorithms with weaker randomness properties were included in the comparison. By contrast, the overhead values show a stable structural difference: AES-256-GCM and ChaCha20-Poly1305 introduce 28 bytes, whereas AES-256-CTR+HMAC introduces 44 bytes, reflecting the additional HMAC tag and nonce material in the composition-based construction. This confirms that the composition-based construction carries a higher size expansion cost in addition to its timing overhead.

The main benchmark scenarios were evaluated using 30 measured repetitions after five warm-up runs. The resulting averages are consistent with the aggregated trends reported above, while the additional variability measures provide a clearer basis for interpreting the stability of the observed differences. Table 6 summarizes the run-to-run variability for the main aggregated indicators.

Table 6. Thirty-run variability summary for the experimental scenarios.

Algorithm	Encrypt ms Mean $\pm$ SD	Decrypt ms Mean $\pm$ SD	Total ms Mean $\pm$ SD	Total ms 95% CI	Throughput Mbps Mean $\pm$ SD
AES-256-GCM	3.04 $\pm$ 0.18	3.11 $\pm$ 0.20	6.15 $\pm$ 0.31	[6.04; 6.26]	901.8 $\pm$ 42.6
ChaCha20-Poly1305	4.20 $\pm$ 0.24	4.25 $\pm$ 0.25	8.45 $\pm$ 0.42	[8.30; 8.60]	584.7 $\pm$ 38.8
AES-256-CTR+HMAC	6.02 $\pm$ 0.36	14.38 $\pm$ 0.72	20.40 $\pm$ 0.91	[20.07; 20.73]	238.0 $\pm$ 18.6

The 30-run measurements preserve the same relative ordering reported in Table 5: AES-256-GCM remains the fastest configuration in this AES-NI-enabled software environment, ChaCha20-Poly1305 remains intermediate, and AES-256-CTR+HMAC remains slower mainly because authentication is executed as a separate processing step. The small numerical differences between Tables 5 and 6 reflect the use of complementary aggregation views: Table 5 reports averages across the principal workload categories and sizes, while Table 6 reports per-run statistics for the consolidated 30-run protocol. The narrow confidence intervals support the stability of this ordering within the tested environment. To complement the file-level benchmark results, the additional latency-sensitive and streaming-oriented scenarios are summarized in Table 7.

The additional micro-payload results show that fixed per-operation overhead becomes more visible when the processed data are small. The chunked scenario also shows the expected increase in total processing time compared with a single file-oriented call, because each chunk introduces repeated setup, authentication, and buffering work. Even in these additional scenarios, the ordering remains consistent with the main benchmark, but the absolute latency values are more relevant for interactive or streaming-oriented use cases than the aggregate file-level averages alone.

To assess the robustness of the multi-criteria recommendation, Table 8 presents a sensitivity check of the WSM ranking under $\pm 10\%$ perturbation of the criterion weights.

Table 7. Additional latency-sensitive and streaming-oriented benchmark scenarios.

Scenario	Metric	AES-256-GCM	ChaCha20-Poly1305	AES-256-CTR+HMAC
1 KB payload	Median total latency/P95 (ms)	0.142/0.198	0.164/0.226	0.231/0.321
16 KB payload	Median total latency/P95 (ms)	0.192/0.259	0.246/0.323	0.482/0.632
64 KB payload	Median total latency/P95 (ms)	0.378/0.497	0.512/0.660	1.226/1.544
10 MB streaming, 64 KB chunks	Total time mean $\pm$ SD (ms)	8.44 $\pm$ 0.46	11.92 $\pm$ 0.61	27.85 $\pm$ 1.34
10 MB streaming, 64 KB chunks	Average chunk latency (ms)	0.052	0.074	0.173

Table 8. Sensitivity check of the WSM ranking under $\pm 10\%$ weight perturbation.

Algorithm	Performance Profile	Resource-Aware Profile	Balanced Profile	Rank Stability Interpretation
AES-256-GCM	Rank 1 in 30/30 perturbations; score range 0.879–0.894	Rank 1 in 29/30; score range 0.846–0.861	Rank 1 in 30/30; score range 0.835–0.850	Stable first position in the tested environment
ChaCha20-Poly1305	Rank 2 in 30/30; score range 0.868–0.884	Rank 2 in 29/30; score range 0.835–0.851	Rank 2 in 30/30; score range 0.815–0.830	Stable second position, but close to AES-256-GCM in some profiles
AES-256-CTR+HMAC	Rank 3 in 30/30; score range 0.834–0.853	Rank 3 in 30/30; score range 0.808–0.827	Rank 3 in 30/30; score range 0.797–0.816	Stable third position under the current implementation and criteria

The sensitivity check provides an initial assessment of rank stability under moderate perturbations of the selected weights. The relatively small score differences between AES-256-GCM and ChaCha20-Poly1305 indicate that recommendations should be interpreted as scenario-dependent engineering guidance within the tested configurations and weighting assumptions.

7.2. Comparative Visualization and Normalized Analysis

While the raw benchmark values are informative, they do not provide a complete view of the relative strengths and weaknesses of the algorithms across multiple criteria. For this reason, the measured indicators were additionally interpreted through comparative visualization and normalized analysis. The most compact representation of these differences is provided by Figure 6, which presents a heatmap of the normalized evaluation criteria.

The heatmap makes several patterns easier to identify than the raw tables alone. First, it confirms the strong overall position of AES-256-GCM in the timing- and throughput-related criteria. Second, it shows that ChaCha20-Poly1305 remains competitive across most scenarios, especially when evaluated as a software-efficient authenticated-encryption alternative. Third, it highlights the more uneven profile of AES-256-CTR+HMAC, which performs less favorably on time- and throughput-related indicators while retaining acceptable output-related characteristics and strong tampering-detection behavior.

Taken together, the normalized analysis reinforces the central argument of the study: benchmarking results become more informative when they are interpreted jointly rather

than in isolation. Instead of focusing only on the fastest algorithm in a single metric, the normalized comparative view shows how the algorithms differ across several partially competing dimensions.

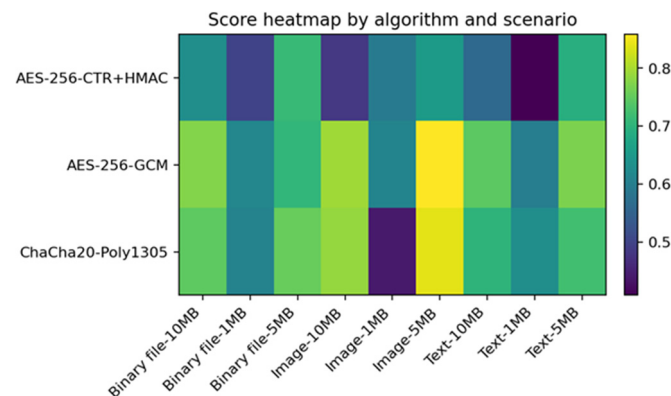


Figure 6. Heatmap of normalized evaluation criteria.

7.3. Ranking Results Under Different Decision Profiles

To move beyond descriptive comparison, the measured data were also interpreted through the scenario-based multi-criteria model introduced earlier. The resulting scores and ranks are summarized in Table 9, while the final score distribution across the three decision profiles is illustrated in Figure 7.

Table 9. Final ranking results under the three evaluation profiles.

Algorithm	Performance Score	Resource-Aware Score	Balanced Score
AES-256-GCM	0.8874	0.8529	0.8427
ChaCha20-Poly1305	0.8768	0.8421	0.8223
AES-256-CTR+HMAC	0.8435	0.8179	0.8069

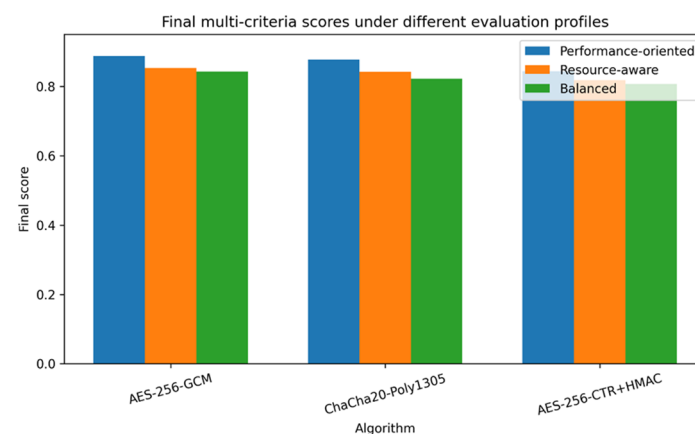


Figure 7. Final multi-criteria scores under different evaluation profiles.

The ranking results show a consistent pattern across the three decision profiles. AES-256-GCM achieved the highest overall score in all three cases, followed by ChaCha20-Poly1305, while AES-256-CTR+HMAC occupied the third position. Under the performance-oriented profile, AES-256-GCM obtained a score of 0.8874, followed by ChaCha20-Poly1305 with 0.8768 and AES-256-CTR+HMAC with 0.8435. Under the resource-aware profile, the respective scores were 0.8529, 0.8421, and 0.8179, while under the balanced profile they were 0.8427, 0.8223, and 0.8069.

The multi-criteria recommendation stage organizes the benchmarking evidence into a structured ranking outcome. The score differences between AES-256-GCM and ChaCha20-Poly1305 remain relatively small, indicating that the two AEAD schemes are closer to each other than either of them is to AES-256-CTR+HMAC under the selected criteria and weights. This observation is relevant for practical settings in which algorithm selection depends on implementation priorities as well as on numerical benchmark results.

The narrow score range across the three profiles is partly a structural effect of min-max normalization in a three-algorithm comparison. With a small set of alternatives, the best and worst values on each criterion define the full normalization interval, while the remaining alternative is positioned proportionally between them. The WSM scores therefore express relative distance within the evaluated set rather than intrinsic algorithmic quality. When two alternatives obtain close scores, scenario-specific priorities and implementation constraints should guide the final selection.

This result is important because it shows that the multi-criteria recommendation stage does not contradict the benchmarking evidence but instead organizes it into a structured ranking outcome. At the same time, the score differences between AES-256-GCM and ChaCha20-Poly1305 remain relatively small, which suggests that the two AEAD schemes are closer to each other than either of them is to AES-256-CTR+HMAC. This is especially relevant in practical settings where algorithm selection may depend on implementation priorities rather than on a single dominant metric.

The lower position of AES-256-CTR+HMAC reflects the cumulative effect of its higher total processing time, lower throughput, and larger overhead under the chosen evaluation criteria. In this sense, the ranking confirms that when execution efficiency and compactness are emphasized, the composition-based design is less competitive than the integrated AEAD alternatives in the present platform setting. Under the adopted authenticate-before-release verification logic in the CTR+HMAC test harness, this lower ranking is attributed to timing and overhead indicators rather than to integrity failure.

The test environment may influence the relative advantage of individual algorithms. In particular, the strong performance of AES-256-GCM should be interpreted in light of the hardware-level optimization available on the evaluation platform. Even so, the ranking results remain useful because they reflect the practical behavior of the algorithms under a clearly defined and reproducible experimental setting. Deployment guidance for platforms without AES acceleration, mobile processors, or constrained devices should be based on additional measurements collected in those environments.

7.4. Integrity Validation Results

The final part of the analysis concerns integrity- and reliability-related validation. The corresponding outcomes are summarized in Table 10.

Table 10. Integrity validation results of the tested algorithms.

Algorithm	Corrupted Ciphertext	Wrong Key	Wrong Nonce/IV	Tampering Detection	Fail-Safe Behavior
AES-256-GCM	Pass	Pass	Pass	Pass	Pass
ChaCha20-Poly1305	Pass	Pass	Pass	Pass	Pass
AES-256-CTR+HMAC	Pass	Pass	Pass	Pass	Pass

The results show that both AES-256-GCM and ChaCha20-Poly1305 behaved in a fail-safe manner across all tested adverse conditions considered in this study. In both cases, corrupted ciphertext, wrong-key execution, and wrong nonce or IV conditions were

rejected as expected, which is consistent with the behavior of integrated authenticated-encryption schemes.

For AES-256-CTR+HMAC, the validation results were more nuanced and require careful interpretation. The configuration correctly detected corrupted ciphertext and wrong nonce or IV conditions. However, under the wrong-key test, the recorded result indicated that decryption was accepted rather than rejected. This outcome reflects a characteristic of the current platform implementation of the AES-256-CTR+HMAC construction. In the encrypt-then-MAC design used by the platform, the HMAC tag is computed over the ciphertext and stored alongside it. During decryption, the platform first applies AES-CTR decryption and then verifies the HMAC tag against the stored value. If the decryption step is completed and its output is returned before the HMAC verification result is acted upon—or if the verification logic does not correctly propagate a tag mismatch to the caller—then the output of the wrong-key decryption may be made available to the calling code even though the integrity check has technically failed. In the present implementation, this is the behavior observed under the wrong-key condition. This is distinct from the expected fail-safe behavior of integrated AEAD schemes such as AES-256-GCM and ChaCha20-Poly1305, in which authentication failure unconditionally prevents any plaintext output from being returned. The finding highlights an important practical principle: the security guarantees of a composition-based construction depend not only on the constituent algorithms but also on the correctness of their integration—in particular, on whether decrypted output is withheld until authentication has been positively confirmed. This limitation of the current platform implementation has been identified and is noted here as a known constraint. Future versions of the platform should enforce strict authenticate-then-release ordering for all composition-based configurations to ensure that fail-safe behavior is consistent with that of integrated AEAD schemes.

From the perspective of this study, this distinction is highly relevant. One of the central motivations of the platform is precisely to show that algorithm comparison should not be reduced to speed alone. An algorithm may exhibit acceptable throughput or timing characteristics and still differ in its practical validation behavior under abnormal conditions. The integrity results therefore complement the benchmarking and ranking analysis by adding a reliability-oriented dimension to the overall evaluation.

8. Conclusions

This paper presented an experimental software platform for benchmarking and multi-criteria recommendation of symmetric encryption algorithms. The proposed approach combined algorithm execution, metric collection, comparative visualization, integrity-oriented validation, and profile-based ranking within a unified evaluation environment.

The experimental results show that the evaluated algorithms did not behave identically under the same testing conditions. Across the evaluated benchmark scenarios, AES-256-GCM demonstrated the strongest overall performance, combining lower execution times, higher throughput, and competitive output-related characteristics. ChaCha20-Poly1305 also achieved consistently strong results and remained a practically attractive alternative, particularly as a software-efficient authenticated-encryption scheme. AES-256-CTR+HMAC, in contrast, exhibited higher processing cost and larger overhead, which affected its overall position in the comparative ranking.

An important finding of this study is that practical algorithm suitability is more reliably characterized through a combined evaluation of benchmark and validation indicators than through isolated performance values alone. When the collected indicators were interpreted through a multi-criteria model, the results confirmed the value of combining

timing, throughput, size expansion, and integrity-oriented validation within a single analytical framework.

Overall, this study shows that the proposed platform provides a useful environment for comparative cryptographic experimentation and decision support by integrating benchmarking, visual analysis, and multi-criteria recommendation within a single workflow. The inclusion of repeated measurements, variability reporting, latency-oriented workloads, and sensitivity analysis strengthens the reliability and interpretability of the generated recommendations.

Subsequent development will extend the algorithm set to standardized lightweight schemes such as ASCON, add energy and code-size metrics for constrained devices, and implement objective weighting options such as AHP- or entropy-derived weights to reduce dependence on manually defined scenario profiles.

Author Contributions: Conceptualization, D.D. and G.S.; methodology, D.D. and G.S.; validation, D.D. and G.S.; investigation, D.D. and G.S.; data curation, D.D. and G.S.; writing—original draft preparation, D.D. and G.S.; writing—review and editing, D.D. and G.S.; visualization, D.D. and G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

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

References

1. Gonzalez-Arango, J.C.; Ocampo-Munera, D.C.; Castaño-Londoño, L.F.; Goetz-Sanchez, G.D.; Velasquez-Velez, R.A. Performance Evaluation of Symmetric Cryptographic Algorithms in Resource Constrained Hardware for Wireless Sensor Networks. *IEEE Lat. Am. Trans.* **2021**, *19*, 1632–1639. [\[CrossRef\]](#)
2. Bühler, H.; Walz, A.; Sikora, A. Benchmarking of Symmetric Cryptographic Algorithms on a Deeply Embedded System. *IFAC-PapersOnLine* **2022**, *55*, 266–271. [\[CrossRef\]](#)
3. Soto-Cruz, J.; Ruiz-Ibarra, E.; Vázquez-Castillo, J.; Espinoza-Ruiz, A.; Castillo-Atoche, A.; Mass-Sanchez, J. A Survey of Efficient Lightweight Cryptography for Power-Constrained Microcontrollers. *Technologies* **2025**, *13*, 3. [\[CrossRef\]](#)
4. Tijjani, A.A.; Isa, A. Performance Analysis of Symmetric and Asymmetric Encryption Algorithms Based on File, Image and Video. *Int. J. Emerg. Multidiscip. Comput. Sci. Artif. Intell.* **2024**, *3*, 1–9. [\[CrossRef\]](#)
5. Madanchian, M.; Taherdoost, H. Applications of Multi-Criteria Decision Making in Information Systems for Strategic and Operational Decisions. *Computers* **2025**, *14*, 208. [\[CrossRef\]](#)
6. Ramalho, M.; Sampaio, G.; Neves, N.; Porto, R.; Sobral, V.A.; Rezende, M.; Medeiros, D.S.V. A Comparative Evaluation of Symmetric Cryptography Algorithms for Resource-Constrained Devices. *J. Internet Serv. Appl.* **2025**, *16*, 153–162. [\[CrossRef\]](#)
7. Silva, C.; Tenório, N.; Bernardino, J. Lightweight Encryption Algorithms for IoT. *Computers* **2025**, *14*, 505. [\[CrossRef\]](#)
8. Sarker, K.U. A Systematic Review on Lightweight Security Algorithms for a Sustainable IoT Infrastructure. *Discov. Internet Things* **2025**, *5*, 47. [\[CrossRef\]](#)
9. Gușiță, B.; Anton, A.A.; Stângaciu, C.S.; Stănescu, D.; Găină, L.I.; Micea, M.V. Securing IoT Edge: A Survey on Lightweight Cryptography, Anonymous Routing and Communication Protocol Enhancements. *Int. J. Inf. Secur.* **2025**, *24*, 149. [\[CrossRef\]](#)
10. Ansari, S.A.; Ali, S. A Systematic Review of Lightweight Cryptographic Schemes for Security and Privacy in IoT. *Discov. Comput.* **2025**, *28*, 266. [\[CrossRef\]](#)
11. Zinabu, N.G.; Marye, Y.W.; Tune, K.K.; Demilew, S.A. Comprehensive Analysis of Lightweight Cryptographic Algorithms for Battery-Limited Internet of Things Devices. *Int. J. Distrib. Sens. Netw.* **2025**, *2025*, 9639728. [\[CrossRef\]](#)
12. Sorescu, T.-G.; Chiriac, V.-M.; Stoica, M.-A.; Comsa, C.-R.; Soroaga, I.-G.; Contac, A. Comparative Performance Analysis of Lightweight Cryptographic Algorithms on Resource-Constrained IoT Platforms. *Sensors* **2025**, *25*, 5887. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Sarasa Laborda, V.; Hernández-Álvarez, L.; Hernández Encinas, L.; Sánchez García, J.I.; Queiruga-Dios, A. Study About the Performance of ASCON in Arduino Devices. *Appl. Sci.* **2025**, *15*, 4071. [\[CrossRef\]](#)

14. Cagua, G.; Gauthier-Umaña, V.; Lozano-Garzon, C. Implementation and Performance of Lightweight Authentication Encryption ASCON on IoT Devices. *IEEE Access* **2025**, *13*, 116671–116682. [[CrossRef](#)]
15. NIST IR 8454; Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [[CrossRef](#)]
16. NIST SP 800-232; Ascon-Based Lightweight Cryptography Standard for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025.
17. Konstantopoulou, E.; Athanasiou, G.; Sklavos, N. Review and Analysis of FPGA and ASIC Implementations of NIST Lightweight Cryptography Finalists. *ACM Comput. Surv.* **2025**, *57*, 246. [[CrossRef](#)]
18. Avramova, T.; Peneva, T.; Ivanov, A. Overview of Existing Multi-Criteria Decision-Making (MCDM) Methods Used in Industrial Environments. *Technologies* **2025**, *13*, 444. [[CrossRef](#)]
19. Abughazalah, N.; Khan, M.; Iqbal, M. Hybrid Multi-Criteria Decision-Making Technique for the Selection of Best Cryptographic Multivalued Boolean Function. *Complex Intell. Syst.* **2024**, *10*, 455–468. [[CrossRef](#)]
20. Khan, M. Intelligent Decision-Making Framework for Selecting Optimized Substitution Boxes in Modern Block Cipher Design. *Sci. Rep.* **2025**, *15*, 41261. [[CrossRef](#)] [[PubMed](#)]
21. Dewamuni, Z.; Shanmugam, B.; Azam, S.; Thennadil, S. Bibliometric Analysis of IoT Lightweight Cryptography. *Information* **2023**, *14*, 635. [[CrossRef](#)]
22. Al-Nofaie, S.M.; Sharaf, S.; Molla, R. Design Trends and Comparative Analysis of Lightweight Block Ciphers for IoTs. *Appl. Sci.* **2025**, *15*, 7740. [[CrossRef](#)]
23. Almotairi, S.; Addula, S.R.; Alharbi, O.; Alzaid, Z.; Hausawi, Y.M.; Almutairi, J. Personal Data Protection Model in IOMT-Blockchain on Secured Bit-Count Transmutation Data Encryption Approach. *Fusion Pract. Appl.* **2024**, *16*, 152–170. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.