Performance analysis of symmetric encryption algorithms for time-critical cybersecurity application

Oleksandr Kuznetsov[1,2,*,†], Yelyzaveta Kuznetsova[2,†], Emanuele Frontoni[3,†], Marco Arnesano[1,†] and Oleksii Smirnov[4,†]

[1] Campus University, 10 Via Isimbardi, 22060 Novedrate, Italy
[2] V. N. Karazin Kharkiv National University, 4 Svobody sq., 61022 Kharkiv, Ukraine
[3] University of Macerata, 30/32 Via Crescimbeni, 62100 Macerata, Italy
[4] Central Ukrainian National Technical University, 8 University ave., 25006 Kropyvnytskyi, Ukraine

Abstract

This study presents a comprehensive performance analysis of symmetric encryption algorithms in the context of time-critical cybersecurity applications. We evaluate a diverse set of algorithms, including established standards and emerging ciphers, across multiple performance metrics relevant to resource-constrained and latency-sensitive environments. Our methodology encompasses rigorous testing of stream encryption speed, packet encryption efficiency, and key/IV setup times on a standardized hardware platform. The results reveal significant variations in algorithm performance across different operational scenarios, highlighting the importance of context-specific algorithm selection. Notably, newer algorithms such as STRUMOK and SNOW 2.0 demonstrate impressive performance across multiple metrics, challenging the dominance of traditional standards in certain application areas. We discuss the implications of our findings for various time-critical applications, including IoT device security, real-time control systems, and secure data aggregation in smart grids. Our analysis underscores the complex trade-offs between security, performance, and resource efficiency inherent in symmetric encryption implementation.

Keywords

symmetric encryption, performance analysis, cryptography, cybersecurity, encryption speed

1. Introduction

The proliferation of interconnected devices and time-sensitive applications has ushered in a new era of cybersecurity challenges [1]. As the digital landscape evolves, the demand for efficient, secure communication in resource-constrained and latency-critical environments has intensified. Symmetric encryption algorithms, long considered the backbone of secure digital communication, find themselves at a critical juncture, facing unprecedented demands for both security and performance [2].

This study embarks on a comprehensive evaluation of symmetric encryption algorithms, focusing on their applicability in time-critical cybersecurity scenarios. Our investigation spans a diverse array of algorithms, from well-established standards to emerging ciphers, each assessed across multiple performance metrics relevant to contemporary security challenges.

The imperative for this research stems from the growing complexity of modern digital ecosystems [3]. Internet of Things (IoT) devices, real-time control systems, wireless sensor networks, and smart grids represent just a fraction of the applications where the balance between security and performance is paramount [4]. In these domains, the selection of an appropriate encryption algorithm can have profound implications for system efficiency, energy consumption, and overall security posture.

Our study aims to bridge the gap between theoretical cryptography and practical implementation by providing empirical performance data crucial for informed decision-making in algorithm selection. By examining encryption speed, memory usage, and setup efficiency across various operational scenarios, we offer insights that are directly applicable to the design and optimization of secure systems in time-sensitive environments.

The objectives of this research are multifold:

- To provide a rigorous, comparative analysis of symmetric encryption algorithms across key performance metrics relevant to time-critical applications.
- To elucidate the trade-offs between security and performance inherent in different algorithmic approaches.

CPITS-II 2024: Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, October 26, 2024, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
oleksandr.kuznetsov@uniecampus.it, kuznetsov@karazin.ua (O. Kuznetsov);
elizabet8smidt12@gmail.com (Y. Kuznetsova);
emanuele.frontoni@unimc.it (E. Frontoni);
marco.arnesano@uniecampus.it (M. Arnesano);
dr.SmirnovOA@gmail.com (O. Smirnov)

0000-0003-2331-6326 (O. Kuznetsov);
0000-0002-0573-0913 (Y. Kuznetsova);
0000-0002-8893-9244 (E. Frontoni);
0000-0003-1700-3075 (M. Arnesano);
0000-0001-9543-874X (O. Smirnov)

- To explore the implications of our findings for specific application areas, including IoT security, real-time control systems, and secure data aggregation in smart grids.
- To identify emerging trends and future research directions in the field of symmetric encryption for time-critical cybersecurity applications.

Through this comprehensive analysis, we aim to contribute to the ongoing discourse on cryptographic implementation in modern digital systems, offering valuable insights for researchers, system designers, and security practitioners alike.

2. Related work

The field of symmetric encryption has witnessed significant advancements in recent years, driven by the evolving demands of modern cybersecurity applications. This section provides an overview of recent research efforts relevant to our study of symmetric encryption algorithms in time-critical scenarios.

Ghafoori and Miyaji (2024) [5] conducted an in-depth analysis of higher-order differential-linear cryptanalysis, focusing on the ChaCha stream cipher. Their work achieved reduced time complexity for attacks on reduced rounds of ChaCha, introducing the first higher-order differential-linear attacks on ChaCha 6 and ChaCha 7. This study underscores the ongoing efforts to assess and improve the security of widely used stream ciphers against advanced cryptanalytic techniques.

In the realm of lightweight cryptography, Huang et al. (2023) [6] proposed IVLBC, an involutive lightweight block cipher designed specifically for IoT applications. Their work highlights the growing emphasis on developing encryption algorithms tailored to resource-constrained environments, a critical consideration in our analysis of algorithm performance in diverse application scenarios.

Kebande (2023) [7] introduced an extended version of the ChaCha20 stream cipher, incorporating enhanced Quarter Round Functions to improve resistance against differential attacks. This research exemplifies the ongoing refinement of established encryption algorithms to address potential vulnerabilities and enhance security in modern applications.

La Scala and Tiwari (2022) [8] presented a novel approach to modeling stream and block ciphers as systems of explicit difference equations over finite fields. Their work, which includes analysis of ciphers such as Trivium and KeeLoq, demonstrates the potential of algebraic methods in assessing cipher security and developing cryptanalytic techniques.

Mishra et al. (2024) [9] provided a comprehensive survey of security and cryptographic perspectives in Industrial Internet of Things (IIoT) environments. Their work emphasizes the critical role of cryptographic primitives in modern cyber defenses and highlights the potential of post-quantum cryptography techniques for future IIoT security.

Urooj et al. (2023) [10] explored the integration of asymmetric and symmetric cryptography in wireless sensor networks, proposing a hybrid approach combining Elliptic Curve Cryptography (ECC) [11–14] and Advanced Encryption Standard (AES). Their research underscores the importance of balancing security and energy efficiency in resource-constrained network environments.

Zhao et al. (2023) [15] developed a block cipher identification scheme based on Hamming weight distribution, addressing the challenge of cryptosystem recognition in multi-classification scenarios. Their work demonstrates the application of machine learning techniques in cryptanalysis and cipher identification.

Caforio et al. (2021) [16] focused on designing energy-optimal symmetric encryption primitives, introducing the concept of "Perfect Trees". Their research aligns closely with our study's emphasis on performance optimization in resource-constrained environments, highlighting the growing importance of energy efficiency in cryptographic implementations.

These recent studies collectively illustrate the diverse challenges and ongoing innovations in the field of symmetric encryption. From advanced cryptanalytic techniques to the development of lightweight ciphers for IoT applications, the research landscape reflects a continuous effort to enhance the security, efficiency, and adaptability of encryption algorithms across various operational contexts. Our study builds upon this foundation, providing a comprehensive performance analysis of symmetric encryption algorithms with a specific focus on time-critical cybersecurity applications.

3. Background

The field of symmetric encryption has evolved significantly since the advent of modern cryptography, driven by advances in computational capabilities and the ever-changing landscape of security threats. This section provides a foundational overview of symmetric encryption, its role in cybersecurity, and the key factors influencing algorithm performance in time-critical applications.

3.1. Symmetric encryption fundamentals

Symmetric encryption algorithms utilize a shared secret key for both encryption and decryption processes. This approach offers several advantages, including high-speed operation and relatively low computational overhead, making it particularly suitable for securing large volumes of data or time-sensitive communications.

The two primary categories of symmetric encryption algorithms are [17]:

- Block Ciphers: These algorithms operate on fixed-size blocks of data, typically 64 or 128 bits. Examples include the Advanced Encryption Standard (AES) and its predecessors. Block ciphers can be employed in various modes of operation, such as Electronic Codebook (ECB), Cipher Block Chaining (CBC), and Counter (CTR) mode, each offering different security and performance characteristics.
- Stream Ciphers: These algorithms encrypt data on a bit-by-bit or byte-by-byte basis, generating a keystream that is combined with the plaintext.

Examples include ChaCha20 and the algorithms in the eSTREAM portfolio. Stream ciphers are often favored in scenarios requiring low latency or where data arrives in a continuous stream.

3.2. Performance considerations in time-critical applications

Several factors influence the performance of symmetric encryption algorithms in time-critical cybersecurity applications [18, 19]:

- Encryption Speed: The number of clock cycles required to encrypt a byte of data is a critical metric, directly impacting the algorithm's suitability for high-throughput or low-latency scenarios.
- Memory Usage: The amount of memory required for algorithm implementation is particularly relevant in resource-constrained environments, such as IoT devices or embedded systems.
- Key Setup Time: The efficiency of initializing the encryption process with a new key affects the algorithm's suitability for scenarios requiring frequent key changes.
- IV/Nonce Setup Time: For algorithms requiring an initialization vector (IV) or nonce, the speed of this setup process can be crucial in applications with frequent session initializations.
- Power Consumption: While not directly measured in this study, power consumption correlates with computational efficiency and is a critical consideration for battery-powered devices.

3.3. Evolving requirements in modern cybersecurity

The landscape of cybersecurity is continually evolving, driven by factors such as [20]:

- Increasing Data Volumes: The exponential growth in data generation and transmission necessitates encryption solutions capable of handling high throughput without introducing significant latency.
- Resource Constraints: The proliferation of IoT and edge computing devices has heightened the need for efficient encryption algorithms that can operate effectively on platforms with limited computational resources and power budgets.
- Quantum Computing Threat: The potential development of large-scale quantum computers poses a significant threat to many current cryptographic systems, driving research into quantum-resistant algorithms.
- Regulatory Compliance: Evolving data protection regulations impose stringent requirements on data security, influencing the selection and implementation of encryption algorithms across various industries.

3.4. Emerging trends in symmetric encryption

Recent developments in the field of symmetric encryption include [9, 21]:

- Lightweight Cryptography: The design of algorithms specifically optimized for resource-constrained environments, balancing security with minimal computational and energy requirements.
- Authenticated Encryption: The integration of authentication mechanisms within encryption algorithms to provide both confidentiality and integrity in a single operation.
- Post-Quantum Cryptography: Research into symmetric encryption algorithms resistant to attacks by quantum computers, focusing on increasing key sizes and developing new algorithmic approaches.
- Homomorphic Encryption: The development of encryption schemes that allow computations to be performed on encrypted data without decryption, opening new possibilities for secure data processing in untrusted environments.

Understanding these fundamental concepts, performance considerations, and emerging trends is crucial for contextualizing the results of our performance analysis and their implications for time-critical cybersecurity applications. This background sets the stage for a nuanced examination of how different symmetric encryption algorithms perform under various operational scenarios and their suitability for diverse application domains.

4. Methodology

This section outlines the comprehensive approach employed in our study to evaluate the performance of lightweight symmetric encryption algorithms in time-critical cybersecurity applications. Our methodology is designed to provide a rigorous, reproducible framework for assessing the efficiency and suitability of these algorithms across various metrics relevant to resource-constrained and latency-sensitive environments.

4.1. Algorithm selection criteria

We selected a diverse set of symmetric encryption algorithms based on the following criteria [22]:

- Relevance to current cybersecurity practices.
- Potential for application in resource-constrained environments.
- Variety in design principles and structural characteristics.
- Inclusion of both well-established and emerging algorithms.

Based on these criteria, we selected the following algorithms for evaluation [23]: AES (128-bit and 256-bit variants); SNOW 2.0 (128-bit and 256-bit variants); Salsa20; HC-128 and HC-256; MICKEY-128; Rabbit; SOSEMANUK; TRIVIUM; STRUMOK; DECIMv2. This selection

encompasses a range of block ciphers, stream ciphers, and hybrid designs, providing a comprehensive view of the current state of symmetric encryption.

4.2. Testing environment

Our evaluation was conducted on a system with the following specifications:

- Processor: AMD Ryzen 7 7840HS with Radeon 780M Graphics, 3.80 GHz.
- RAM: 64.0 GB (62.8 GB available).
- System Type: 64-bit operating system, ×64-based processor.
- Operating System: Windows 11 Home Edition.

To ensure consistency and reproducibility, all tests were performed under controlled conditions, with minimal background processes running.

4.3. Performance metrics

We evaluated the algorithms across several key performance metrics [24]:

- Stream Encryption Speed: Measured in cycles per byte and megabits per second (Mbps).
- Packet Encryption Speed: Evaluated for packet sizes of 40, 576, and 1500 bytes, reported in cycles per packet, cycles per byte, and Mbps.
- Key Setup Speed: Measured in cycles per setup and setups per second.
- IV (Initialization Vector) Setup Speed: Measured in cycles per setup and setups per second.

These metrics were chosen to provide a comprehensive view of algorithm performance across different operational scenarios.

4.4. Evaluation process

Our evaluation process consisted of the following steps [24]:

- Implementation Verification: We used verified implementations of each algorithm, ensuring consistency with their respective specifications.
- Stream Encryption Test: Each algorithm encrypted 1 Gigabyte of data, with performance measured using the RDTSC (Read Time-Stamp Counter) instruction for precise cycle counting.
- Packet Encryption Test: We tested three packet sizes (40, 576, and 1500 bytes) to simulate various network traffic patterns. For each size, we encrypted multiple blocks under different keys to account for real-world usage scenarios.
- Key and IV Setup Tests: We performed multiple key and IV setups for each algorithm, measuring the time taken for these critical initialization processes.
- Data Collection and Analysis: Performance data was collected over multiple runs to ensure statistical significance. We calculated average performance metrics and standard deviations to account for system variability.
- Comparative Analysis: We analyzed the collected data to compare the performance of the algorithms across different metrics and usage scenarios.

4.5. Evaluation tools

We developed custom benchmarking tools to ensure consistent and accurate measurement across all algorithms. These tools were designed to minimize overhead and provide precise timing information. The core of our testing suite utilized the following components:

- A high-resolution timer using the RDTSC instruction for cycle-accurate measurements.
- Memory management routines to ensure consistent cache behavior across tests.
- Data generation functions to provide consistent input across all algorithms.
- Output verification routines to ensure the correctness of encryption operations.

4.6. Reproducibility measures

To ensure the reproducibility of our results, we have taken the following measures:

- Detailed documentation of all testing procedures and environment configurations.
- Use of open-source implementations where available, with clear version information.
- Publication of our custom benchmarking tools and scripts (available upon request).
- Multiple test runs with statistical analysis to account for system variability.

By adhering to these methodological principles, we aim to provide a comprehensive and reliable evaluation of symmetric encryption algorithms, offering valuable insights for the selection and implementation of these algorithms in time-critical cybersecurity applications.

5. Overview of selected encryption algorithms

This section provides a concise overview of the symmetric encryption algorithms selected for our performance analysis. Each algorithm is described in terms of its core structure, key characteristics, and potential applications in time-critical cybersecurity scenarios.

5.1. Advanced encryption standard

AES, standardized by NIST in 2001, remains a cornerstone of modern cryptography. We evaluate both 128-bit and 256-bit key variants [25, 26].

Structure: AES employs a substitution-permutation network with a fixed block size of 128 bits. It operates through a series of rounds, each comprising four stages: SubBytes, ShiftRows, MixColumns, and AddRoundKey.

Key Characteristics:

- Wide-spread adoption and extensive scrutiny.
- Efficient hardware implementation.
- Varying number of rounds based on key size (10 for 128-bit, 14 for 256-bit).

Applications: Widely used in secure communications, financial transactions, and data storage.

5.2. SNOW 2.0

SNOW 2.0, an evolution of the SNOW stream cipher, is designed for software efficiency [27, 28].

Structure: Combines a Linear Feedback Shift Register (LFSR) with a Finite State Machine (FSM). The LFSR has 16 stages, each holding a 32-bit word, while the FSM contains two 32-bit registers.

Key Characteristics:

- High efficiency in software implementations.
- Support for both 128-bit and 256-bit key sizes.
- Strong resistance against known attacks on stream ciphers.

Applications: Suitable for high-speed encryption in software-based systems, particularly in network security.

5.3. Salsa20

Designed by Daniel J. Bernstein in 2005, Salsa20 aims for high speed across various platforms [29, 30].

Structure: Operates on 64-byte blocks using a series of quarter-round functions, consisting of 32-bit addition, XOR, and rotation operations.

Key Characteristics:

- Simple design facilitating analysis and implementation.
- Supports 128-bit and 256-bit keys with a 64-bit nonce.
- No known practical attacks compromise its security.

Applications: Well-suited for applications requiring fast, secure encryption on diverse hardware platforms.

5.4. HC-128 and HC-256

Part of the eSTREAM portfolio, these stream ciphers were designed by Hongjun Wu [31].

Structure: Utilize two secret tables, each containing 512 32-bit elements, updated during keystream generation.

Key Characteristics:

- Excellent performance in software, especially with large caches.
- HC-128 uses a 128-bit key and IV, HC-256 uses 256-bit for both.
- Strong security guarantees with no known practical attacks.

Applications: Ideal for software-based encryption in systems with sufficient memory resources.

5.5. MICKEY

Designed for resource-constrained environments, MICKEY supports an 80-bit key and IV [32, 33].

Structure: Consists of two registers: a linear feedback shift register (LFSR) and a non-linear feedback shift register (NFSR), with irregular clocking.

Key Characteristics:

- Compact implementation suitable for constrained devices.
- Designed specifically for hardware implementation.
- Resistance to known cryptanalytic techniques.

Applications: Well-suited for IoT devices and other resource-limited hardware scenarios.

5.6. Rabbit

Developed by Cryptico A/S, Rabbit is a high-speed stream cipher [34].

Structure: Based on iterating a system of non-linear functions, maintaining an internal state of 513 bits.

Key Characteristics:

- 128-bit key and 64-bit IV.
- Designed for high speed in both software and hardware.
- Compact state size suitable for memory-constrained environments.

Applications: Excellent for scenarios requiring fast encryption with limited memory resources.

5.7. SOSEMANUK

Based on the design principles of SNOW 2.0, SOSEMANUK aims for software efficiency [34].

Structure: Combines an LFSR with an FSM, inspired by the SERPENT block cipher.

Key Characteristics:

- Supports 128-bit and 256-bit keys, with 64-bit to 128-bit IV.
- Designed for high efficiency in software implementations.
- Strong resistance against known cryptanalytic attacks.

Applications: Suitable for software-based encryption in various network security scenarios.

5.8. TRIVIUM

Designed by Christophe De Cannière and Bart Preneel, TRIVIUM aims for simplicity and high performance [33, 36].

Structure: Internal state of 288 bits stored in three shift registers of different lengths.

Key Characteristics:

- 80-bit key and 80-bit IV.
- Extremely simple design facilitating both hardware and software implementations.

- High throughput in hardware implementations.

Applications: Ideal for hardware-based encryption in resource-constrained environments.

5.9. STRUMOK

A relatively new stream cipher designed for high-speed software implementations [37, 38].

Structure: Optimized for 64-bit processors, utilizing a combination of simple operations for high speed.

Key Characteristics:

- 256-bit or 512-bit key and 256-bit IV.
- Designed specifically for high performance on modern processors.
- Ongoing evaluation by the cryptographic community.

Applications: Suitable for high-performance edge computing and other scenarios requiring fast software-based encryption.

5.10. DECIMv2

An improved version of the original DECIM stream cipher, designed for lightweight applications [39].

Structure: Employs a combination of a non-linear feedback shift register (NFSR) and a linear feedback shift register (LFSR).

Key Characteristics:

- 80-bit key and 80-bit IV.
- Designed for hardware efficiency in constrained environments.
- Improved security compared to its predecessor.

Applications: Well-suited for resource-constrained hardware, such as RFID systems or lightweight IoT devices.

This diverse selection of algorithms provides a comprehensive basis for our performance analysis, covering a wide range of design principles and potential applications in time-critical cybersecurity scenarios.

6. Performance analysis results

This section presents a comprehensive analysis of the performance metrics obtained from our rigorous evaluation of the selected symmetric encryption algorithms. We provide a detailed comparison of encryption speeds, memory usage, and other critical performance indicators. Furthermore, we discuss the inherent trade-offs between security and performance, offering insights into the suitability of each algorithm for various time-critical cybersecurity applications.

6.1. Presentation of performance metrics

Our analysis focuses on four key performance metrics: stream encryption speed, packet encryption speed, key setup speed, and IV setup speed. These metrics provide a holistic view of each algorithm's efficiency across different operational scenarios.

6.2. Stream encryption performance

Fig. 1 presents the stream encryption performance for all tested algorithms, sorted by encryption speed in descending order.

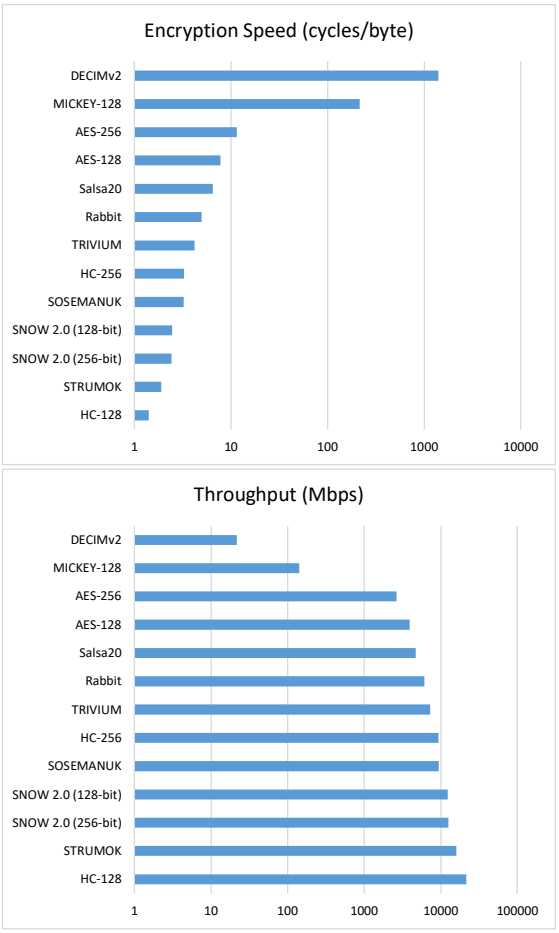

Figure 1: Stream Encryption Performance

As evident from Fig. 1, HC-128 demonstrates exceptional performance in stream encryption, achieving the highest throughput of 21,550.26 Mbps with the lowest cycles per byte (1.41). STRUMOK and both variants of SNOW 2.0 also exhibit impressive performance, with throughputs exceeding 12,000 Mbps. These results suggest that these algorithms are particularly well-suited for applications requiring high-speed stream encryption, such as real-time video streaming or high-bandwidth network traffic encryption.

Interestingly, the widely-used AES algorithm, particularly its 256-bit variant, shows comparatively lower performance in stream encryption. This observation underscores the potential benefits of exploring alternative algorithms for scenarios requiring high-throughput stream encryption in time-critical applications.

6.3. Packet encryption performance

Fig. 2 presents the packet encryption performance for all tested algorithms, based on the weighted average of the Simple Internet Mix (IMIX) model.

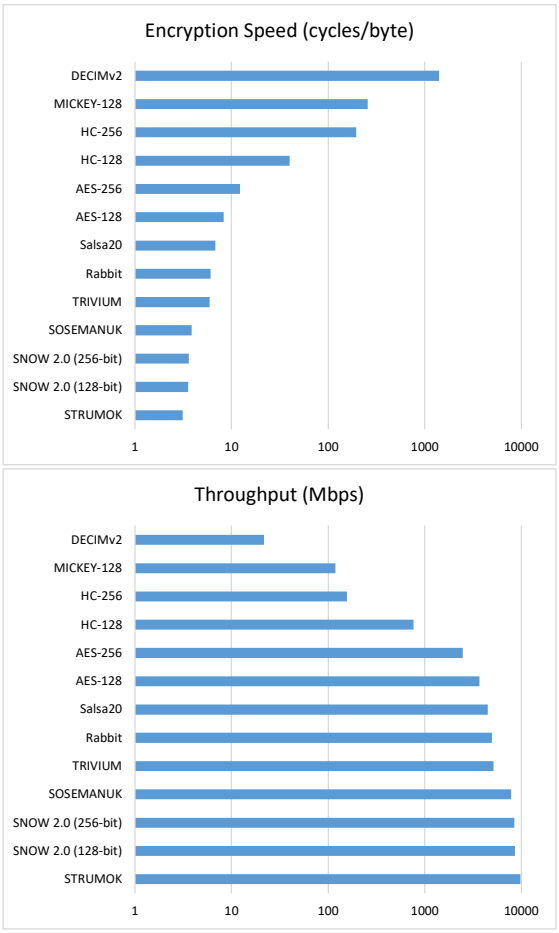

Figure 2: Packet Encryption Performance (IMIX Weighted Average)

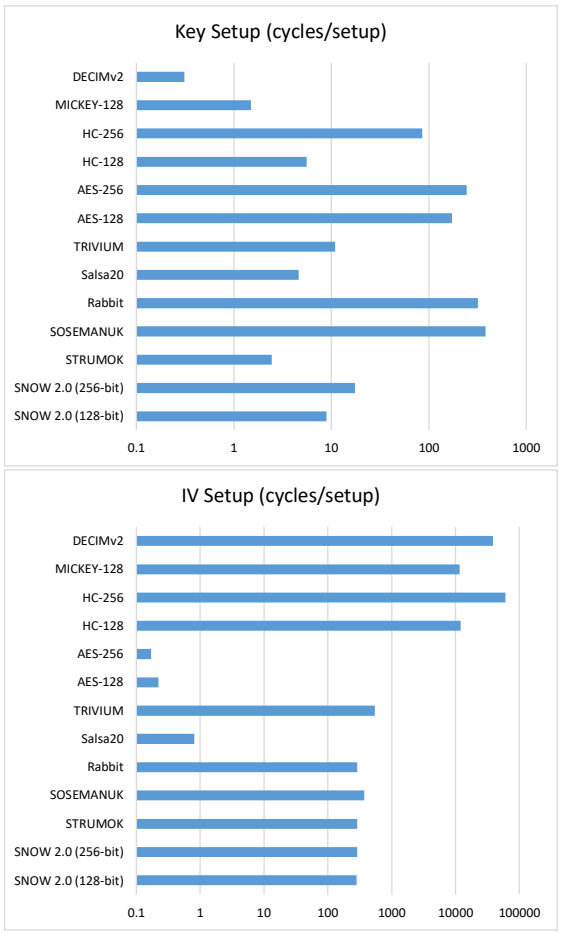

Figure 3: Key and IV Setup Efficiency

The packet encryption results reveal interesting shifts in performance compared to stream encryption. STRUMOK, SNOW 2.0, and SOSEMANUK demonstrate superior performance in packet encryption, maintaining high throughput across various packet sizes. These algorithms show particular promise for applications dealing with diverse packet sizes, such as secure IoT communication or virtual private networks (VPNs).

Notably, HC-128, which excelled in stream encryption, shows a significant performance drop in packet encryption. This disparity highlights the importance of considering both stream and packet encryption performance when selecting an algorithm for time-critical applications.

6.4. Key and IV setup efficiency

Fig. 3 presents the key and IV setup performance for the tested algorithms.

The key and IV setup efficiency results reveal significant variations among the tested algorithms. STRUMOK and SNOW 2.0 demonstrate exceptional performance in both key and IV setup operations, with STRUMOK achieving over 15 million key setups per second and both algorithms managing over 13 million IV setups per second. This efficiency makes them particularly well-suited for applications requiring frequent key or IV changes, such as in rapidly changing network environments or systems with high-security requirements.

6.5. Comparative analysis across different algorithms

The performance metrics presented in Figs. 1–3 reveals several key insights:

- Stream Encryption Efficiency: HC-128, STRUMOK, and SNOW 2.0 consistently demonstrate superior performance in stream encryption, making them ideal candidates for high-throughput applications.
- Packet Encryption Variability: The relative performance of algorithms shifts when considering packet encryption, with STRUMOK and SNOW 2.0 maintaining high efficiency across different packet sizes.
- Setup Speed Trade-offs: Algorithms like Salsa20 and AES show a stark contrast between their key and IV setup performance, with very fast IV setup but slower key setup operations.
- Consistency Across Metrics: SNOW 2.0 and STRUMOK demonstrate consistently high performance across all metrics, suggesting their versatility for various time-critical applications.
- Resource-Constrained Scenarios: Lightweight algorithms like TRIVIUM and MICKEY-128 show competitive performance in certain metrics,

making them suitable for resource-constrained environments.

6.6. Discussion of trade-offs between security and performance

The performance analysis reveals several important trade-offs between security and efficiency:

- Key Size vs. Speed: Generally, algorithms with larger key sizes (e.g., AES-256) show lower encryption speeds compared to their shorter key counterparts. This trade-off is particularly evident in time-critical applications where every cycle counts.
- Complexity vs. Efficiency: More complex algorithms like AES, which involve multiple rounds of substitution and permutation, tend to have lower throughput compared to simpler designs like HC-128 or STRUMOK. However, this complexity often correlates with higher resistance to cryptanalysis.
- Setup Efficiency vs. Security: Algorithms with very fast setup times (e.g., DECIMv2 for key setup) may be more vulnerable to certain types of attacks if keys are changed frequently. Conversely, algorithms with slower setup times often incorporate a more thorough mixing of the key material.
- Stream vs. Packet Performance: Some algorithms excel in stream encryption but show reduced efficiency in packet encryption (e.g., HC-128). This trade-off is crucial when selecting algorithms for specific network protocols or data transmission patterns.
- Memory Usage vs. Speed: Algorithms that use larger internal states or lookup tables (e.g., HC-256) often achieve higher speeds but may be less suitable for memory-constrained devices.

In conclusion, the selection of an encryption algorithm for time-critical cybersecurity applications must carefully balance these performance metrics against the specific security requirements and resource constraints of the target system. The data presented here provides a foundation for making informed decisions in algorithm selection, highlighting the need for a nuanced approach that considers the full spectrum of performance characteristics in the context of the intended application.

7. Application scenarios

The performance characteristics of symmetric encryption algorithms have significant implications for their suitability in various time-critical cybersecurity applications. This section examines the relevance of our findings to specific application areas, highlighting how the performance trade-offs of different algorithms align with the unique requirements of each domain.

7.1. IoT device security

IoT devices present a unique set of challenges for encryption implementation due to their resource constraints and diverse operational environments. Our analysis reveals several key considerations for IoT device security [9]:

- Resource Efficiency: Lightweight algorithms such as TRIVIUM and MICKEY-128 demonstrate potential for IoT applications due to their efficient performance on resource-constrained hardware. TRIVIUM, in particular, shows promising results in packet encryption (5.93 cycles/byte), making it suitable for IoT devices that transmit small data packets intermittently.
- Energy Consumption: The correlation between cycles per byte and energy consumption suggests that algorithms like STRUMOK and SNOW 2.0, with their low cycles per byte in both stream and packet encryption, could be beneficial for battery-powered IoT devices where energy efficiency is crucial.
- Flexibility: The variability in IoT device capabilities necessitates a flexible approach to encryption. For more powerful IoT edge devices, algorithms like HC-128 or STRUMOK could provide high-speed encryption for data streams, while resource-constrained sensors might benefit from the efficiency of TRIVIUM or the versatility of SNOW 2.0.
- Key Management: The rapid key setup speed of algorithms like STRUMOK (15,217,391.30 setups/sec) could be advantageous in IoT networks requiring frequent key rotations to maintain security in potentially compromised environments.

Implementation Recommendation: For heterogeneous IoT networks, a hybrid approach using SNOW 2.0 for more capable devices and TRIVIUM for highly constrained sensors could provide a balance of security and efficiency across the network.

7.2. Real-time control systems

Real-time control systems, such as those found in industrial automation or autonomous vehicles, require encryption solutions that can operate within strict timing constraints without introducing significant latency. Our findings suggest the following considerations [40]:

- Low Latency: The exceptional performance of HC-128 in stream encryption (1.41 cycles/byte) makes it a strong candidate for real-time systems dealing with continuous data streams, such as sensor feeds in autonomous vehicles.
- Deterministic Performance: Algorithms with consistent performance across different data sizes, like SNOW 2.0 and STRUMOK, are well-suited for real-time control systems where predictable timing is critical.
- Packet Encryption Efficiency: For systems communicating via network packets, STRUMOK's superior performance in packet encryption (3.13 cycles/byte) could minimize encryption-induced delays in command and control communications.

- Setup Speed: The rapid IV setup times of algorithms like AES (50,000,000 setups/sec for AES-128) could be beneficial in scenarios requiring frequent session initializations without compromising overall system responsiveness.

Implementation Recommendation: For real-time control systems with varying data flow characteristics, a combination of HC-128 for stream data and STRUMOK for packet-based communication could provide optimal performance while maintaining robust security.

7.3. Secure communication in wireless sensor networks

Wireless Sensor Networks (WSNs) face unique challenges in implementing secure communication due to their distributed nature, limited resources, and often harsh operational environments. Our analysis highlights several important factors [41, 42]:

- Energy Efficiency: The low cycles per byte of algorithms like SNOW 2.0 and SOSEMANUK in packet encryption (3.55 and 3.85 cycles/byte, respectively) could translate to lower energy consumption, crucial for extending the operational life of battery-powered sensor nodes.
- Lightweight Implementation: TRIVIUM's simple design and efficient hardware implementation make it an attractive option for WSNs with highly constrained sensor nodes.
- Scalability: The strong performance of STRUMOK across both stream and packet encryption suggests its suitability for heterogeneous WSNs where some nodes may handle aggregated data streams while others transmit individual sensor readings.
- Resilience to Packet Loss: In WSNs prone to packet loss, the independent encryption of packets facilitated by algorithms with efficient IV setup, such as Salsa20, could enhance network resilience by minimizing the impact of lost packets on subsequent communications.

Implementation Recommendation: A tiered approach using TRIVIUM for the most constrained sensor nodes and SNOW 2.0 for cluster heads or data aggregation points could provide a balance of efficiency and security across the WSN.

7.4. Privacy-preserving data aggregation in smart grids

Smart grid systems require secure, privacy-preserving mechanisms for data aggregation and analysis. The performance characteristics of encryption algorithms have significant implications for balancing privacy, efficiency, and scalability in these systems [43]:

- High-throughput Encryption: The exceptional stream encryption performance of HC-128 (21550.26 Mbps) could be leveraged for securing high-volume data flows from smart meters to aggregation points without introducing significant latency.
- Efficient Packet Processing: STRUMOK's strong performance in packet encryption (9730.46 Mbps) makes it suitable for securing the diverse packet sizes typically encountered in smart grid communication protocols.
- Homomorphic Properties: While not directly measured in our study, the potential for partial homomorphic operations in some stream ciphers could be exploited for privacy-preserving aggregation. Further research into the homomorphic properties of high-performing algorithms like SNOW 2.0 or STRUMOK could yield valuable insights for smart grid applications.
- Key Agility: The rapid key setup capabilities of algorithms like STRUMOK and SNOW 2.0 facilitate frequent key rotations, enhancing long-term security in the persistent threat environment of smart grid infrastructure.

Implementation Recommendation: A hybrid system using HC-128 for high-volume data streams and STRUMOK for packet-based communications could provide a robust, efficient encryption solution for smart grid data aggregation.

7.5. Secure data processing in edge computing

While not the primary focus of this study, our findings have relevant implications for secure data processing in edge computing environments [43]:

- Versatility: The consistently high performance of SNOW 2.0 and STRUMOK across various metrics suggests their suitability for the diverse workloads encountered in edge computing scenarios.
- Computational Efficiency: The low cycles per byte achieved by top-performing algorithms could translate to reduced computational overhead, crucial for maintaining the low-latency promise of edge computing.
- Adaptability: The range of performance profiles observed across our tested algorithms suggests the potential for adaptive encryption strategies in edge environments, dynamically selecting algorithms based on current processing loads and security requirements.
- Implementation Recommendation: Further research into the performance of these algorithms on typical edge computing hardware is warranted, with particular attention to the balance between encryption speed and overall application performance in resource-shared environments.

In conclusion, the performance characteristics of symmetric encryption algorithms have profound implications for their deployment in time-critical cybersecurity applications. The diverse requirements of IoT security, real-time control systems, wireless sensor networks, smart grids, and edge computing necessitate careful consideration of algorithm selection and implementation strategies. Our findings provide a

foundation for informed decision-making in these critical application areas, highlighting the need for tailored, often hybrid approaches to encryption that balance security, efficiency, and application-specific constraints.

8. Discussion

The comprehensive analysis of symmetric encryption algorithms presented in this study reveals several key insights with significant implications for time-critical cybersecurity applications:

- Performance Variability: Our results demonstrate substantial variability in algorithm performance across different metrics. This variability underscores the importance of selecting encryption algorithms based on specific application requirements rather than relying on generalized performance claims.
- Trade-offs between Security and Efficiency: The inverse relationship often observed between key size and encryption speed highlights the ongoing challenge of balancing security with performance in resource-constrained environments. This trade-off is particularly evident in the comparison between AES variants and newer, streamlined algorithms like STRUMOK and SNOW 2.0.
- Emergence of Specialized Algorithms: The strong performance of algorithms designed for specific scenarios (e.g., TRIVIUM for hardware implementation, HC-128 for software-based stream encryption) suggests a trend towards more specialized cryptographic solutions. This specialization could lead to more efficient security implementations in diverse application areas.
- Importance of Comprehensive Evaluation: The discrepancies observed between stream and packet encryption performance for some algorithms (notably HC-128) emphasize the need for comprehensive evaluation across multiple metrics when selecting encryption solutions for complex systems.
- Potential for Adaptive Encryption Strategies: The varied performance profiles of the tested algorithms across different metrics suggest the potential for adaptive encryption strategies that dynamically select algorithms based on current system conditions and security requirements.
- Challenges in Standardization: The superior performance of newer algorithms like STRUMOK in certain metrics poses challenges for standardization efforts, as it suggests the need for periodic re-evaluation and potential updates to cryptographic standards to incorporate emerging, high-performance algorithms.

These findings have broad implications for the design and implementation of secure systems in various domains, from IoT and edge computing to smart grids and real-time control systems. They highlight the need for a nuanced, context-aware approach to cryptographic implementation in time-critical applications.

9. Conclusions

This study provides a comprehensive performance analysis of symmetric encryption algorithms in the context of time-critical cybersecurity applications. Through rigorous testing and evaluation, we have identified key performance characteristics of a diverse set of algorithms, ranging from well-established standards like AES to emerging ciphers like STRUMOK.

Our findings reveal that no single algorithm excels across all performance metrics, underscoring the importance of tailored encryption strategies for specific application scenarios. Notably, newer algorithms such as STRUMOK and SNOW 2.0 demonstrate impressive performance across multiple metrics, challenging the dominance of traditional standards in certain application areas.

The analysis of application scenarios highlights the potential for significant performance improvements through the strategic selection and implementation of encryption algorithms. In IoT environments, for instance, the use of lightweight algorithms like TRIVIUM could enhance security without overburdening resource-constrained devices. Similarly, the high-speed encryption capabilities of HC-128 and STRUMOK offer promising solutions for real-time control systems and smart grid data aggregation.

However, this study also reveals the complexities involved in balancing security, performance, and resource efficiency. The trade-offs between key size, encryption speed, and setup efficiency necessitate careful consideration in algorithm selection, particularly in heterogeneous environments with diverse security requirements.

Looking forward, our results suggest several directions for future research:

- Further investigation into the performance characteristics of emerging algorithms on diverse hardware platforms, particularly in edge computing environments.
- Exploration of adaptive encryption strategies that leverage the strengths of multiple algorithms to optimize security and performance dynamically.
- Development of standardized benchmarking methodologies for evaluating encryption performance in time-critical applications, facilitating more direct comparisons between studies.
- Investigation of the energy consumption implications of different encryption algorithms, particularly in the context of battery-powered devices in IoT and WSN scenarios.

In conclusion, this study contributes to the ongoing dialogue on the selection and implementation of symmetric encryption algorithms in time-critical cybersecurity applications. By providing a comprehensive performance analysis and discussing its implications across various application scenarios, we aim to facilitate more informed decision-making in the design and deployment of secure systems in an increasingly interconnected and time-sensitive digital landscape.

Acknowledgments

- This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 101007820 - TRUST.
- This publication reflects only the author's view and the REA is not responsible for any use that may be made of the information it contains.

References

[1] V. Kampourakis, V. Gkioulos, S. Katsikas, A Systematic Literature Review on Wireless Security Testbeds in the Cyber-Physical Realm, Comput. Secur. 133 (2023) 103383. doi: 10.1016/j.cose.2023.103383.

[2] V. Rudnytskyi, et al., Cryptographic Encoding in Modern Symmetric and Asymmetric Encryption, Procedia Computer Science 207 (2022) 54–63. doi: 10.1016/j.procs.2022.09.037.

[3] N. Kolokotronis, et al., An Intelligent Platform for Threat Assessment and Cyber-Attack Mitigation in IoMT Ecosystems, IEEE Globecom Workshops (2022) 541–546. doi: 10.1109/GCWkshps56602.2022.10008548.

[4] S. F. Ahmed, et al., Industrial Internet of Things Enabled Technologies, Challenges, and Future Directions, Comput. Electrical Eng. 110 (2023) 108847. doi: 10.1016/j.compeleceng.2023.108847.

[5] N. Ghafoori, A. Miyaji, Higher-Order Differential-Linear Cryptanalysis of ChaCha Stream Cipher, IEEE Access 12 (2024) 13386–13399. doi: 10.1109/ACCESS.2024.3356868.

[6] X. Huang, L. Li, J. Yang, IVLBC: An Involutive Lightweight Block Cipher for Internet of Things, IEEE Syst. J. 17 (2023) 3192–3203. doi: 10.1109/JSYST.2022.3227951.

[7] V. R. Kebande, Extended-Chacha20 Stream Cipher with Enhanced Quarter Round Function, IEEE Access 11 (2023) 114220–114237. doi: 10.1109/ACCESS.2023.3324612.

[8] R. La Scala, S. K. Tiwari, Stream/block Ciphers, Difference Equations and Algebraic Attacks, J. Symbolic Comput. 109 (2022) 177–198. doi: 10.1016/j.jsc.2021.09.001.

[9] N. Mishra, S. Hafizul Islam, S. Zeadally, A Survey on Security and Cryptographic Perspective of Industrial-Internet-of-Things, Internet of Things 25 (2024) 101037. doi: 10.1016/j.iot.2023.101037.

[10] S. Urooj, et al., Cryptographic Data Security for Reliable Wireless Sensor Network, Alexandria Eng. J. 72 (2023) 37–50. doi: 10.1016/j.aej.2023.03.061.

[11] A. Bessalov, et al., Modeling CSIKE Algorithm on Non-Cyclic Edwards Curves, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 1–10.

[12] A. Bessalov, et al., Multifunctional CRS Encryption Scheme on Isogenies of NonSupersingular Edwards Curves, in: Workshop on Classic, Quantum, and Post-Quantum Cryptography, vol. 3504 (2023) 12–25.

[13] A. Bessalov, et al., CSIKE-ENC Combined Encryption Scheme with Optimized Degrees of Isogeny Distribution, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 36–45.

[14] A. Bessalov, V. Sokolov, S. Abramov, Efficient Commutative PQC Algorithms on Isogenies of Edwards Curves, Cryptography 8(3), iss. 38 (2024) 1–17. doi:10.3390/cryptography8030038.

[15] L. Zhao, et al., Block Cipher Identification Scheme Based on Hamming Weight Distribution, IEEE Access 11 (2023) 21364–21373. doi: 10.1109/ACCESS.2023.3249753.

[16] A. Caforio, et al., Perfect Trees: Designing Energy-Optimal Symmetric Encryption Primitives, Iacr Transactions On Symmetric Cryptology (2021). doi: 10.46586/tosc.v2021.i4.36-73.

[17] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, New York: Wiley (1996).

[18] A. Andrushkevych, et al., A Prospective Lightweight Block Cipher for Green IT Engineering, Springer International Publishing (2019). doi: 10.1007/978-3-030-00253-4_5.

[19] A. A. Kuznetsov, et al., Analysis of Standardized Algorithms for Streaming Cryptographic Convention, Defined in ISO/IEC 18033-4, Springer Science and Business Media Deutschland GmbH (2022). doi: 10.1007/978-3-030-79770-6_7.

[20] Y. Qian, F. Ye, H.-H. Chen, Cryptographic Techniques, in: Security in Wireless Communication Networks, IEEE (2022) 51–76. doi: 10.1002/9781119244400.ch4.

[21] A. Tiwari, Chapter 14. Cryptography in blockchain, Distributed Computing to Blockchain, Academic Press (2023) 251–265. doi: 10.1016/B978-0-323-96146-2.00011-5.

[22] A. A. Kuznetsov, et al., Criteria and Indices Substantiation of the Stream Cryptoconversion Efficiency, Springer Science and Business Media Deutschland GmbH (2022). doi: 10.1007/978-3-030-79770-6_2.

[23] A. A. Kuznetsov, et al., Comparison of Stream Modes in Block Symmetric Ciphers, Springer Science and Business Media Deutschland GmbH (2022). doi: 10.1007/978-3-030-79770-6_6.

[24] The eSTREAM Portfolio Page, (n.d.). https://www.ecrypt.eu.org/stream/e2-rabbit.html

[25] J. Daemen, V. Rijmen, Specification of Rijndael, The Design of Rijndael: The Advanced Encryption Standard (AES) (2020) 31–51. doi: 10.1007/978-3-662-60769-5_3.

[26] N.I. of S. and Technology, Advanced Encryption Standard (AES), U.S. Department of Commerce (2001). doi: 10.6028/NIST.FIPS.197.

[27] M. Bahadori, K. Järvinen, V. Niemi, FPGA Implementations of 256-Bit SNOW Stream Ciphers for Postquantum Mobile Security, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 29 (2021) 1943–1954. doi: 10.1109/TVLSI.2021.3108430.

[28] A. N. Alekseychuk, S. M. Koniushok, M. V. Poremskyi, A Method of Evaluating the Security of Snow 2.0-Like Ciphers Against Correlation Attacks Over the Finite

Extensions of Two Element Field, Cybern Syst Anal 56 (2020) 40–52. doi: 10.1007/s10559-020-00220-1.
[29] D. J. Bernstein, The Salsa20 Family of Stream Ciphers, New Stream Cipher Designs: The eSTREAM Finalists, (2008) 84–97. doi: 10.1007/978-3-540-68351-3_8.
[30] T. R. Campbell, Daence: Salsa20 and ChaCha in Deterministic Authenticated Encryption with no noNCEnse (2020).
[31] H. Wu, A New Stream Cipher HC-256, Fast Software Encryption (2004) 226–244. doi: 10.1007/978-3-540-25937-4_15.
[32] S. Babbage, M. Dodd, The MICKEY Stream Ciphers, New Stream Cipher Designs: The eSTREAM Finalists, (2008) 191–209. doi: 10.1007/978-3-540-68351-3_15.
[33] B. Li, M. Liu, D. Lin, FPGA implementations of Grain v1, Mickey 2.0, Trivium, Lizard and Plantlet, Microprocessors and Microsystems 78 (2020) 103210. doi: 10.1016/j.micpro.2020.103210.
[34] M. Boesgaard, et al., The Rabbit Stream Cipher - Design and Security Analysis (2004).
[35] C. Berbain, et al., Sosemanuk, a Fast Software-Oriented Stream Cipher, New Stream Cipher Designs: The eSTREAM Finalists (2008) 98–118. doi: 10.1007/978-3-540-68351-3_9.
[36] C. De Cannière, Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles, Information Security (2006) 171–186. doi: 10.1007/11836810_13.
[37] I. Gorbenko, et al., Strumok Keystream Generator, in: IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), IEEE (2018) 294–299. doi: 10.1109/DESSERT.2018.8409147.
[38] A. A. Kuznetsov, et al., Stream Symmetric Cipher "Strumok," Studies in Systems, Decision and Control 375 (2022) 467–516. doi: 10.1007/978-3-030-79770-6_16.
[39] C. Berbain, et al., Decimv2, New Stream Cipher Designs: The eSTREAM Finalists (2008) 140–151. doi: 10.1007/978-3-540-68351-3_11.
[40] A. Barradas, A. Tejeda-Gil, R.-M. Cantón-Croda, Real-Time Big Data Architecture for Processing Cryptocurrency and Social Media Data: A Clustering Approach Based on k-Means, Algorithms 15 (2022) 140. doi: 10.3390/a15050140.
[41] L. Deng, et al., Certificateless Anonymous Signcryption Scheme with Provable Security in the Standard Model Suitable for Healthcare Wireless Sensor Networks, IEEE Internet of Things Journal 10 (2023) 15953–15965. doi: 10.1109/JIOT.2023.3266335.
[42] J. C. Gonzalez-Arango, et al., Performance Evaluation of Symmetric Cryptographic Algorithms in resource constrained hardware for Wireless Sensor Networks, IEEE Latin America Transactions 19 (2021) 1632–1639. doi: 10.1109/TLA.2021.9477225.
[43] D. Wang, et al., Multi-Keyword Searchable Encryption for Smart Grid Edge Computing, Electric Power Systems Research 212 (2022) 108223. doi: 10.1016/j.epsr.2022.108223.