IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative Performance Analysis of Lightweight Encryption Algorithms for Securing IoT Networks

Bhanushali Shiv G, Doddi Srija, Jayashree R, Jaswanth Kumar, Dr. Eashwar Sivakumar Department of Electronics and Communication Engineering
Department of Computer Science and Engineering
Faculty of Engineering & Technology JAIN (Deemed-to-be University) Bengaluru, India

Abstract

The Internet of Things (IoT) connects resource-constrained devices that continuously exchange data across networks, making efficient security mechanisms important for reliable communication. This paper presents a comparative performance analysis of five encryption algorithms, namely AES-128, PRESENT, Salsa20, ChaCha20, and SIMON, with emphasis on their suitability for software-based IoT communication scenarios. A controlled Python-based simulation environment was used to evaluate encryption and decryption perfor-mance for input sizes of 1 KB, 10 KB, 100 KB, 500 KB, and 1 MB. The reported execution-time results show that ChaCha20 consistently requires the least encryption and decryption time among the evaluated algorithms, followed closely by Salsa20. At 1 MB, the reported encryption times are 271.1 ms for ChaCha20, 293.6 ms for Salsa20, 551.7 ms for AES-128, 796.3 ms for SIMON, and 931.4 ms for PRESENT. The corresponding decryption times are 269.4 ms, 291.5 ms, 549.8 ms, 842.7 ms, and 983.4 ms, respectively. These results indicate that stream ciphers, particularly ChaCha20, provide a strong software-performance profile for time-sensitive IoT communication, while AES-128 remains a useful reference because of its established security standardization. The study is limited to software simulation and does not claim direct energy measurements or hardware-level performance.

Index Terms - Internet of Things, Lightweight Cryptography, AES-128, PRESENT, Salsa20, ChaCha20, SIMON, IoT Secu-rity

1. Introduction

The Internet of Things (IoT) has expanded the use of interconnected sensing and computing devices in areas such as healthcare, smart homes, industrial automation, environmental monitoring, and smart cities. These de-vices collect and transmit data continuously, often while operating with limited processing capability, memory, storage, and energy resources. Security mechanisms therefore need to provide adequate protection without im-posing excessive computational cost. Recent studies of lightweight cryptography identify resource constraints as a central consideration when selecting cryptographic techniques for IoT environments [1]–[3].

Conventional cryptographic algorithms remain important because they provide well-established

security properties and extensive deployment support. AES, for example, is standardized by the National Institute of Standards and Technology (NIST) and supports 128-bit, 192-bit, and 256-bit keys with a 128-bit block size [4]. However, the suitability of an algorithm in an IoT system depends not only on its security properties but also on execution time, resource requirements, implementation environment, and application constraints.

Lightweight cryptographic research has consequently investigated compact block ciphers and efficient stream ciphers for constrained platforms. PRESENT was designed for extremely constrained environments, while SIMON was developed as a family of lightweight block ciphers. Salsa20 and its related ChaCha family are software-oriented stream ciphers designed with efficiency in mind [5]–[7]. More recent research continues to examine lightweight cryptography as a practical requirement for constrained IoT deployments [1]–[3].

This work focuses on a consistent software-based comparison of five algorithms used in the project: AES-128, PRESENT, Salsa20, ChaCha20, and SIMON. The evaluation uses the execution-time results reported for five data sizes, from 1 KB to 1 MB. The objective is not to identify a universally best cryptographic algorithm, but to determine which algorithms provide favorable performance under the specific software test conditions used in this study.

The main contributions of this study are as follows:

- A comparative evaluation of five encryption algorithms relevant to constrained IoT environments.
- An analysis of encryption and decryption execution time across five data sizes using a common software-based evaluation setting.
- A performance-oriented interpretation of the results for software-based IoT communication.
- An explicit discussion of the limitations associated with simulation-based evaluation and the absence of direct energy measurements.

2. Related Work

Lightweight cryptography has become an important research direction because IoT devices frequently oper-ate under constraints that differ substantially from conventional computing systems. Thakor et al. reviewed lightweight cryptographic algorithms for resource-constrained IoT devices and discussed the trade-off be-tween security, implementation cost, and resource consumption [1]. Ullah et al. surveyed lightweight cryp-tography for resource-constrained machine-type communication and emphasized the need for efficient cryp-tographic mechanisms in autonomous and embedded environments [2]. Rana et al. presented a survey of lightweight cryptography in IoT networks and compared block and stream cipher approaches in the context of constrained devices [3].

AES remains a major benchmark for comparative studies because it is a standardized and widely deployed symmetric block cipher. The current NIST FIPS 197 publication defines AES and its three approved key lengths [4]. ChaCha20 is specified together with Poly1305 in RFC 8439 and is widely recognized as an efficient stream-cipher construction for software environments [6]. The ChaCha family was derived from Salsa20, making the comparison between Salsa20 and ChaCha20 relevant when examining software execution behavior [5].

PRESENT was designed specifically for highly constrained environments, including applications where hardware area and implementation cost are important [7]. SIMON is another lightweight block-cipher family intended for constrained implementations [8]. These designs illustrate that a cryptographic algorithm can be optimized for resource-constrained environments even when its software behavior does not necessarily match the behavior of a hardware implementation.

Recent work also emphasizes that lightweight cryptography should be evaluated with awareness of the target environment. NIST's lightweight cryptography standardization process selected the Ascon family for lightweight authenticated encryption applications, reflecting the continued development of cryptographic stan-dards for constrained environments [9]. Hardware-focused studies further show that the performance ranking of lightweight designs can depend strongly on the target platform and implementation characteristics [10].

3. Research Problem and Objectives

Although many studies compare individual cryptographic algorithms, comparisons can be difficult to interpret when different hardware platforms, software libraries, input sizes, and performance measures are used. A controlled comparison using a common software environment provides a simpler way to observe relative execution behavior.

The research problem addressed in this study is the selection of encryption algorithms that can provide practical performance for IoT communication without introducing unnecessary computational delay. The study specifically examines the following objectives:

1. To compare AES-128, PRESENT, Salsa20, ChaCha20, and SIMON.
2. To measure and compare encryption and decryption execution time.
3. To examine performance as the input size increases from 1 KB to 1 MB.
4. To identify algorithms that show favorable software performance for time-sensitive IoT communication.

4. Methodology

4.1. Simulation Environment

The project uses a software-based simulation approach implemented in Python and executed in a cloud-based environment such as Google Colab. The absence of dedicated IoT hardware means that the study evaluates al-gorithm behavior under controlled software conditions rather than measuring the physical energy consumption or hardware-specific performance of a sensor node.

4.2. System Architecture

The simulated communication model represents an IoT sender, an encryption stage, a communication channel, and a receiver. Input data is generated at the sender, encrypted using the selected algorithm, transmitted conceptually through the communication stage, and decrypted at the receiver. The decrypted output is checked to ensure that the original data can be recovered.

Figure 1: Simplified simulation workflow used for the comparative evaluation.

4.3. Selected Encryption Algorithms

Five algorithms are included in the comparison.

AES-128 is a symmetric block cipher using a 128-bit key and 128-bit blocks. It provides the standardized reference point in the experiment [4].

PRESENT is a lightweight block cipher designed for highly constrained environments. Its design empha-sizes small implementation requirements [7].

Salsa20 is a stream-cipher family designed for efficient software operation [5].

ChaCha20 is a stream cipher derived from the Salsa family and standardized for use with Poly1305 in RFC 8439 [6].

SIMON is a family of lightweight block ciphers designed for constrained implementations [8].

4.4. Input Data

Random input data was evaluated at five sizes:

- 1 KB,
- 10 KB,
- 100 KB,
- 500 KB, and
- 1 MB.

This range provides both small-message and larger-transfer conditions for observing how execution time changes with increasing input size.

4.5. Performance Metrics

The primary reported metric used in this paper is execution time in milliseconds. Encryption time is the time required to convert plaintext into ciphertext, while decryption time is the time required to recover plaintext from ciphertext. Lower execution time indicates lower processing delay under the same test conditions.
Throughput can be derived from the reported execution times using

$$T = \frac{S}{t/1000}, \quad (1)$$

where S is the input size in KB and t is execution time in milliseconds. Because the supplied project results do not provide a complete numerical throughput table, throughput values derived from execution time are treated as secondary calculations rather than independently measured observations.

4.6. Experimental Procedure

For each algorithm and input size, the workflow consists of generating the input, applying encryption, record-ing execution time, applying decryption, recording execution time, and verifying recovery of the original data. The project documentation states that experiments were repeated and average results were calculated. The reported comparison is therefore presented as the project-level averaged result.

The experiment is software-based. No claim is made that the reported timings represent a particular physi-cal microcontroller. Hardware-level energy consumption was not directly measured.

5. Results and Discussion

This section presents the reported performance results for the selected encryption algorithms. The comparison focuses primarily on encryption and decryption execution time across five data sizes.

5.1. Encryption Time Analysis

Table 1 presents the encryption times reported in the project results.

Table 1: Reported Encryption Time (ms)

Algorithm	1 KB	10 KB	100 KB	500 KB	1 MB
AES-128	0.82	6.14	58.3	284.1	551.7
PRESENT	1.24	9.87	94.2	463.5	931.4
Salsa20	0.41	3.12	29.8	147.3	293.6
ChaCha20	**0.38**	**2.89**	**27.6**	**136.2**	**271.1**
SIMON	1.07	8.43	80.6	396.8	796.3

ChaCha20 has the lowest reported encryption time at every tested input size. Salsa20 is the second fastest throughout the table. AES-128 occupies an intermediate position, while PRESENT and SIMON require more time in this software comparison.

At 1 MB, ChaCha20 requires 271.1 ms compared with 293.6 ms for Salsa20 and 551.7 ms for AES-128. Thus, the measured ChaCha20 encryption time at 1 MB is approximately 51% lower than the AES-128 value under the reported test conditions.

The increasing data size also shows a broadly increasing execution time for every algorithm. This is expected because more input data must be processed. The important observation is the relative difference between algorithms rather than the absolute timing alone.

5.2. Decryption Time Analysis

The reported decryption results are shown in Table 2.

Table 2: Reported Decryption Time (ms)

Algorithm	1 KB	10 KB	100 KB	500 KB	1 MB
AES-128	0.79	5.91	56.2	273.8	549.8
PRESENT	1.31	10.42	99.6	489.7	983.4
Salsa20	0.40	3.09	29.5	146.1	291.5
ChaCha20	**0.38**	**2.87**	**27.4**	**135.1**	**269.4**
SIMON	1.14	8.97	85.4	419.6	842.7

The decryption results follow the same general ordering as encryption. ChaCha20 has the lowest reported decryption time at all five data sizes, followed closely by Salsa20. At 1 MB, ChaCha20 requires 269.4 ms, while Salsa20, AES-128, SIMON, and PRESENT require 291.5 ms, 549.8 ms, 842.7 ms, and 983.4 ms, respectively.

The consistency between encryption and decryption results is relevant because an IoT communication sys-tem normally requires both operations. In the reported experiment, ChaCha20 maintains the lowest execution time in both directions.

5.3. Derived Throughput Analysis

Using the execution-time definition above, the approximate 1 MB throughput can be derived from the reported encryption results. Table 3 gives these calculated values. These are not additional measurements; they are arithmetic transformations of the reported timing data.

Table 3: Derived 1 MB Encryption Throughput

Algorithm	Approx. KB/s
AES-128	1856.1
PRESENT	1099.4
Salsa20	3487.7
ChaCha20	**3777.2**
SIMON	1285.9

The derived values reinforce the execution-time comparison. ChaCha20 has the highest calculated through-put because it has the lowest encryption time at 1 MB. Salsa20 follows closely, while AES-128, SIMON, and PRESENT show lower calculated throughput.

5.4. Overall Performance Comparison

The project results indicate that ChaCha20 provides the strongest execution-time performance among the five evaluated algorithms, with Salsa20 also showing strong performance. The same conclusion is supported by the numerical encryption and decryption tables.

PRESENT and SIMON are designed with constrained implementations in mind, but their reported software timings are higher than those of the two stream ciphers in this experiment. This illustrates why algorithm selec-tion should consider the deployment platform. A cipher that is attractive for a small hardware implementation may not necessarily produce the lowest execution time in a high-level software environment.

AES-128 provides a useful baseline because of its established standardization and widespread use [4]. The results do not imply that AES-128 is unsuitable for IoT systems; instead, they show that ChaCha20 and Salsa20 produced lower execution times in the particular software comparison.

6. Discussion

The results indicate a clear performance advantage for ChaCha20 and Salsa20 in the supplied software mea-surements. Both are stream-cipher designs, and their execution-time behavior in this experiment remains favorable as the data size increases. This agrees with the broader literature that identifies efficient stream and lightweight cryptographic constructions as important options for constrained environments [1]–[3].

However, performance alone is not sufficient to select a cryptographic primitive. Security requirements, implementation maturity, nonce management, key management, and hardware support also influence the final decision. ChaCha20, for example, is commonly used together with Poly1305 for authenticated encryption rather than as an isolated confidentiality mechanism [6]. Similarly, the use of AES in practical IoT systems depends on the selected mode of operation and implementation.

The comparison also demonstrates why results from different platforms should not be mixed without quali-fication. The project documentation includes a separate microcontroller comparison based on previously pub-lished results. Those values are not treated as experimental measurements in this paper because the present work does not include physical microcontroller testing.

The study therefore supports a practical conclusion: for the software environment represented by the project, ChaCha20 provides the strongest execution-time profile among the five evaluated algorithms, while AES-128 remains a strong standardized reference. For an actual deployment, the final selection should be validated on the intended IoT hardware and communication protocol.

7. Limitations

The study has several limitations. First, the experiments were performed in a software simulation environment rather than on dedicated IoT hardware. Consequently, the reported execution times should not be interpreted as direct measurements of a specific microcontroller or sensor platform.

Second, energy consumption was not directly measured. Statements about energy efficiency are therefore limited to the general motivation for lightweight cryptography and cannot be presented as experimentally measured energy savings.

Third, the supplied numerical result tables contain detailed encryption and decryption times but do not pro-vide a complete numerical table for CPU utilization, memory usage, or independently measured throughput. The paper therefore avoids introducing unsupported values and treats throughput calculations derived from execution time as secondary analysis.

Finally, the comparison evaluates execution performance rather than providing a complete cryptanalytic security assessment. Security properties of the algorithms are discussed from established literature and stan-dards, but resistance to every class of attack was not experimentally tested in this project.

8. Conclusion and Future Work

This paper presented a comparative performance analysis of AES-128, PRESENT, Salsa20, ChaCha20, and SIMON for software-based IoT communication. The evaluation considered five input sizes ranging from 1 KB to 1 MB and focused primarily on encryption and decryption execution time.

ChaCha20 achieved the lowest reported encryption and decryption times for every tested input size, while Salsa20 consistently followed as the second fastest algorithm. AES-128 showed moderate execution time and provides an important standardized benchmark. PRESENT and SIMON, although designed for lightweight and constrained implementations, showed higher execution times than the stream ciphers in the reported software environment.

The findings suggest that ChaCha20 is a strong candidate when low software execution time is a primary requirement. Nevertheless, the result should not be interpreted as evidence that one algorithm is universally optimal for every IoT device. Actual deployment decisions should consider hardware architecture, energy con-sumption, security requirements, communication protocols, implementation support, and key-management requirements.

Future work should extend the evaluation to physical IoT platforms such as microcontrollers, directly measure energy consumption, and evaluate additional authenticated lightweight cryptographic algorithms. In particular, NIST's selection of the Ascon family for lightweight cryptography provides a relevant direction for extending future comparisons toward modern lightweight authenticated encryption standards [9]. Network-level measurements and real-time IoT protocols can also be incorporated to evaluate end-to-end latency and communication overhead.

References

[1] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021.

[2] S. Ullah, R. Z. Radzi, T. M. Yazdani, A. Alshehri, and I. Khan, "Types of Lightweight Cryptographies in Current Developments for Resource Constrained Machine Type Communication Devices: Challenges and Opportunities," *IEEE Access*, vol. 10, pp. 35589–35604, 2022.

[3] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in IoT networks: A survey," *Future Generation Computer Systems*, vol. 129, pp. 77–89, 2022.

[4] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," FIPS 197, 2023.

[5] D. J. Bernstein, "The Salsa20 family of stream ciphers," in *New Stream Cipher Designs: The eSTREAM Finalists*, 2008.

[6] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC 8439, Internet Research Task Force, 2018.

[7] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," in *Cryptographic Hardware and Embedded Systems (CHES 2007)*, 2007.

[8] R. Beaulieu et al., "The SIMON and SPECK Families of Lightweight Block Ciphers," Cryptology ePrint Archive, Report 2013/404, 2013.

[9] M. S. Turan et al., "Status Report on the Final Round of the NIST Lightweight Cryptography Standard-ization Process," NIST IR 8454, 2023.

[10] K. Tsantikidou and N. Sklavos, "Hardware Limitations of Lightweight Cryptographic Designs for IoT in Healthcare," *Cryptography*, vol. 6, no. 3, p. 45, 2022.