I. J. Mathematical Sciences and Computing, **2025, 4, 39-49**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2025.04.04

Benchmarking SHA256 vs Scrypt in Blockchain Block Discovery

Fitria*
Darmajaya Institute of Informatics and Business, Faculty of Computer Science, Bandar Lampung 35142, Indonesia
E-mail: fitria@darmajaya.ac.id
ORCID iD: https://orcid.org/0000-0003-4751-1441
*Corresponding Author

Joko Triloka
Darmajaya Institute of Informatics and Business, Faculty of Computer Science, Bandar Lampung 35142, Indonesia
E-mail: joko.triloka@darmajaya.ac.id
ORCID iD: https://orcid.org/0000-0002-6700-8432

Eko Win Kenali
Lampung State Polytechnic, Information Technology Department, Software Engineering Technology Study Program, Bandar Lampung, 35141, Indonesia
E-mail: ekowins07@polinela.ac.id
ORCID iD: https://orcid.org/0000-0003-1743-5361

Riko Herwanto
Darmajaya Institute of Informatics and Business, Faculty of Computer Science, Bandar Lampung 35142, Indonesia
Lampung University, Doctoral Program of Computer Science, Faculty of Mathematics and Science, Bandar Lampung, 35141, Indonesia
E-mail: rikoherwanto@darmajaya.ac.id
ORCID iD: https://orcid.org/0000-0001-7452-5107

Received: 02 March, 2025; Revised: 12 June, 2025; Accepted: 13 September, 2025; Published: 08 December, 2025

Abstract: Blockchain technology has emerged as a pivotal innovation across multiple sectors due to its decentralized nature, secure transaction processing, and transparency. Central to blockchain operations are cryptographic hashing algorithms like SHA256 and Scrypt, which play a crucial role in ensuring transaction integrity and security. This study conducts a comprehensive benchmarking analysis of SHA256 and Scrypt, focusing on their performance in blockchain block discovery, specifically evaluating hashing speed and block discovery probability. SHA256, known for its high hashing speed, demonstrated rates reaching 101.111 kH/s during a 10-hour test, whereas Scrypt performed at a slower average speed of 9 kH/s. However, Scrypt exhibited a higher probability of block discovery, achieving up to 8.18%, significantly surpassing SHA256's near-zero probability under similar conditions. Tests across various CPUs underscore these differences: SHA256 excels in raw hashing speed, while Scrypt's memory-intensive design offers greater ASIC resistance and a higher likelihood of block discovery, especially in environments that demand enhanced security. These findings highlight the importance of choosing an algorithm aligned with the specific requirements of a blockchain application, balancing speed, security, and resistance to specialized hardware attacks. The study suggests that hybrid approaches combining SHA256's speed with Scrypt's security features could maximize both efficiency and security, contributing valuable insights into the ongoing optimization of blockchain technology.

Index Terms: Blockchain, Comparison, Proof of Works, Scrypt, SHA256

1. Introduction

Blockchain technology has rapidly gained prominence across various sectors, owing to its decentralized nature, secure transaction processing, and transparency[1-3]. In a blockchain, each block consists of a series of transactions that are securely hashed and linked to the previous block[4], forming an immutable chain[3,5], [6,7]. Cryptographic algorithms, particularly hashing algorithms like SHA256 and Scrypt, play a crucial role in ensuring the security and integrity of these transactions[8].

SHA256, a member of the SHA-2 family, has been widely adopted in blockchain networks such as Bitcoin, where it serves as the backbone of the proof-of-work mechanism. The algorithm is known for its strong cryptographic security and relatively fast hashing speed, making it a popular choice for securing digital transactions. However, as blockchain technology evolves, there is a growing interest in alternative hashing algorithms like Scrypt, which is used in networks such as Litecoin. Scrypt offers a memory-intensive approach to hashing, making it more resistant to certain types of attacks, such as ASIC mining[9].

Recent studies have focused on the comparative analysis of these algorithms in various blockchain applications. For instance, a study by [9] compared the efficiency of SHA256 and Scrypt in terms of computational cost and energy consumption, highlighting the trade-offs between security and performance in different blockchain environments. Similarly, [6,10][11] examined the impact of these algorithms on the speed of block discovery, concluding that while SHA256 provides faster processing, Scrypt offers enhanced security under specific conditions[12]. For broader context, this revised version also references alternative hashing algorithms such as Keccak and Argon2, which are gaining attention in blockchain applications.

The present study seeks to extend this body of knowledge by conducting a detailed benchmarking of SHA256 and Scrypt algorithms within the context of blockchain block discovery[12]. By employing a variety of test systems and conditions, this research aims to determine the most efficient algorithm for discovering new blocks[9,13], thereby contributing to the ongoing discourse on optimizing blockchain technologies[14-15]. This research underscores the importance of algorithm selection in blockchain block discovery, where hashing algorithms like SHA256 and Scrypt are not only foundational for data integrity but also influence operational efficiency and security. SHA256, as widely adopted in Bitcoin, emphasizes speed, making it suitable for applications where rapid block discovery is paramount. In contrast, Scrypt, utilized in Litecoin, incorporates a memory-intensive design that enhances ASIC resistance, thus supporting decentralized and secure mining environments. By benchmarking these algorithms under varying CPU setups, this study aims to provide insights into optimal blockchain configurations tailored to diverse performance and security requirements

This addition highlights the study's relevance and provides context for comparing SHA256 and Scrypt, emphasizing their distinct applications in blockchain environments . Building upon these observations, the study further explores how distinct characteristics of SHA256 and Scrypt impact their suitability in different blockchain environments[16]. Specifically, SHA256's straightforward, compute-intensive design achieves high hashing throughput, positioning it as a prime candidate for blockchain networks prioritizing speed[17]. However, this comes at the cost of vulnerability to specialized hardware, such as ASICs, which could threaten decentralization by concentrating mining power among a few entities capable of investing in such hardware[18].

Conversely, Scrypt's memory-intensive architecture inherently resists ASIC mining, preserving a level of inclusivity for users with general-purpose hardware[19]. This inclusivity aligns with decentralized principles, making Scrypt ideal for networks aiming to minimize centralized mining influence[20]. By testing these algorithms across CPUs with varying processing power, this research captures a comprehensive view of their performance dynamics, offering a nuanced understanding of the trade-offs between processing speed and resistance to centralization[21].

2. Method

This research employs a structured approach to comprehensively evaluate the performance of the SHA256 and Scrypt algorithms in blockchain block discovery. The research is divided into distinct stages: data gathering, program design, testing, and analysis.

Initially, a thorough literature review was conducted to gather scholarly articles, technical documentation, and electronic resources related to SHA256, Scrypt, and blockchain technology. This foundational knowledge established a theoretical basis for comparing the two algorithms and guided the design of the experimental program. Subsequently, two blockchain software implementations—Bitcoin Core, which operates with SHA256, and Litecoin Core, which uses Scrypt—were selected for comparative testing. Each of these programs offers a distinct implementation of the algorithms, serving as reliable environments for performance benchmarking.

The next stage focused on setting up a test environment, selecting four Central Processing Units (CPUs) to capture a range of performance capabilities: Intel Pentium G4560, AMD Ryzen 3 1300X, Intel Core i5-4200H, and AMD Ryzen 7 1800X. Testing was conducted under three separate durations—10 hours, 3 hours, and 1 hour—for each algorithm, allowing for an assessment of performance across both extended and short durations. The test environment was configured on Linux Ubuntu 16.04, and Python scripts were developed to collect metrics automatically. These

scripts recorded data in the debug.log file, which served as the primary data source for later analysis. Key metrics collected included hashing speed, measured in kilohashes per second (kH/s), and block discovery probability.

A notable focus of the study was the evaluation of ASIC resistance in Scrypt, due to its memory-intensive nature. Various configurations of Scrypt with different memory cost parameters (N values) were tested to observe the effect of memory demand on hashing time and probability of block discovery. This configuration variation provides insight into Scrypt's effectiveness in enhancing security against ASIC (Application-Specific Integrated Circuit) mining, a critical aspect of decentralization.

The data analysis stage involved processing the metrics recorded in debug.log using Python tools to compute essential performance metrics, such as hash rates, discovery probabilities, and the performance ratio between SHA256 and Scrypt. This analysis highlighted the trade-offs between the algorithms, such as SHA256's speed advantage and Scrypt's higher block discovery probability and ASIC resistance. The study's findings contribute valuable insights into selecting optimal hashing algorithms based on specific blockchain application requirements, balancing factors like speed, security, and ASIC resistance for effective blockchain block discovery[22-23].

The initial phase of this study involves searching for scholarly journals, documentation, and electronic publications pertaining to the subject matter of SHA256, Scrypt, and Blockchain [18] [24-25]. Subsequently, a program design is formulated based on the information gathered from these reference sources, with the aim of substantiating the hypotheses posited. The data obtained from the testing phase is subsequently analyzed.

The present study aims to conduct a comparative analysis between the SHA256 Algorithm and the Scrypt Algorithm in terms of their efficacy in discovering novel blocks[26-27]. The utilized source programs for conducting the tests were Bitcoin Core, which operates on the SHA256 algorithm, and Litecoin Core, which operates on the Scrypt algorithm[28-30]. The tests were conducted using four distinct central processing units (CPUs) in order to accomplish the task.

Research process, which is divided into several stages, each contributing to a comprehensive analysis of the SHA256 and Scrypt algorithms in blockchain technology:

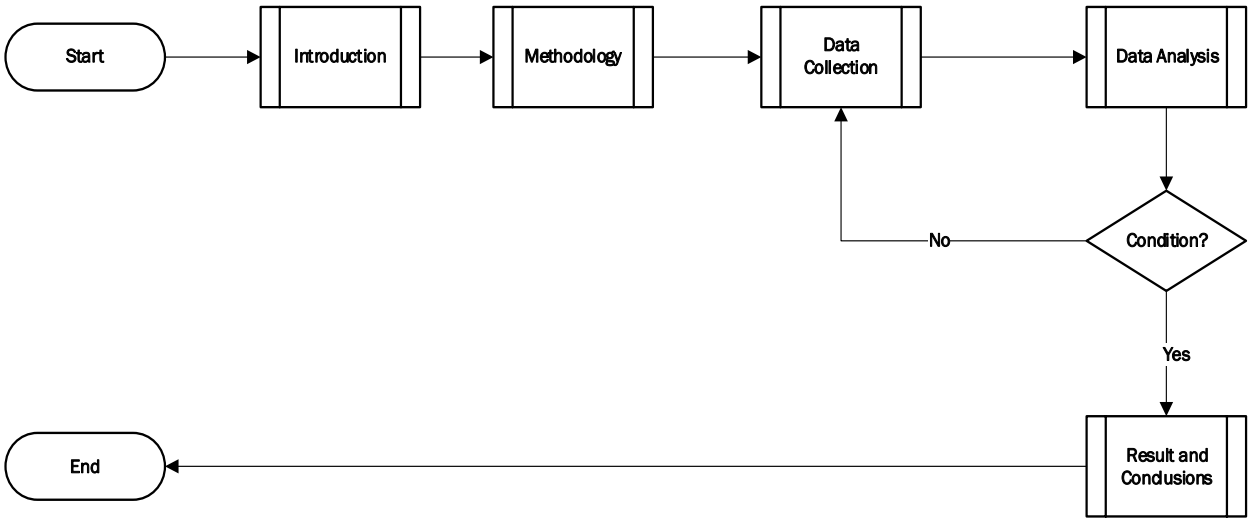

Fig. 1. Research Design

Figure 1 shown each stage is interconnected, ensuring a systematic approach to evaluating the SHA256 and Scrypt algorithms. The flowchart not only guides the research process but also ensures that each phase builds upon the previous one, leading to well-supported conclusions.

- The research process begins with the **Introduction** stage, which establishes the foundation for the study. At this point, the purpose and significance of comparing the SHA256 and Scrypt algorithms are clearly articulated. This stage sets the context for the research, emphasizing the importance of understanding the differences between these two algorithms in the context of blockchain technology.
- Following the introduction, the process moves to the **Methodology** stage. This phase is critical as it involves the practical setup required for the research. It includes implementing the SHA256 and Scrypt algorithms on selected Central Processing Units (CPUs) and determining the appropriate hashing durations. This preparation ensures that the system is properly configured for the subsequent data collection.
- Once the methodology is in place, the research enters the **Data Collection** stage. During this phase, data is gathered as the algorithms are executed. The focus is on collecting key metrics such as hashing time, the number of attempts made, and the success rate of block discoveries. These metrics are crucial for the subsequent analysis, as they provide the raw data needed to evaluate the performance of each algorithm.

- The next stage is **Data Analysis**, where the data collected is thoroughly processed and examined. This is a decision-making point in the flowchart, where the condition is checked to determine if the data meets the criteria for moving forward. If not, the process loops back to further data collection. If the condition is met, the analysis compares the performance of SHA256 and Scrypt based on the metrics gathered, offering insights into which algorithm is more efficient.
- Finally, the process concludes with the **Results and Conclusion** stage. Here, the findings from the analysis are summarized, highlighting which algorithm proved more effective. The conclusions drawn from this research provide valuable insights that could inform future studies or practical applications in blockchain technology. This stage serves as the end point of the research process, where the overall outcomes are consolidated.

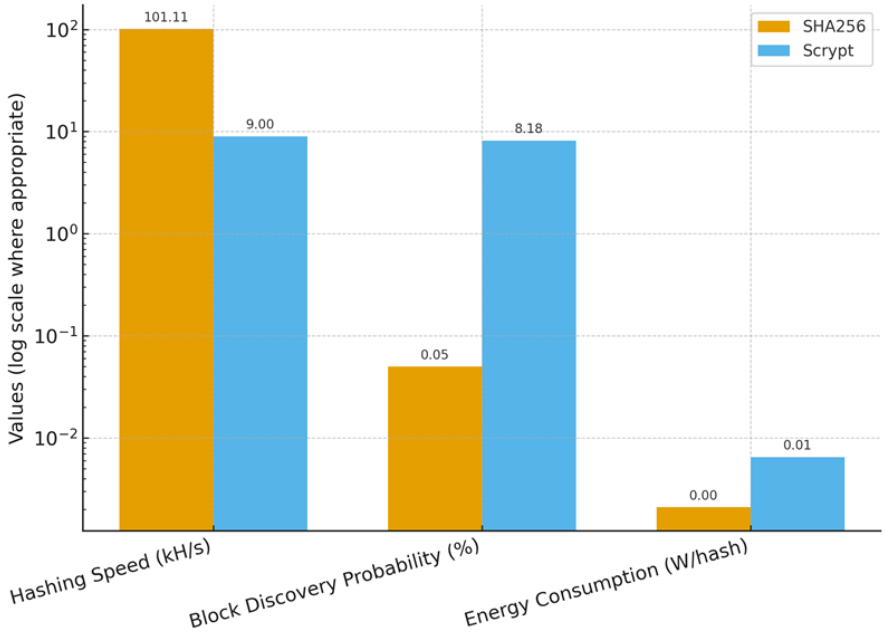

Fig. 2. Comparative Performance Metrics of SHA256 vs. Scrypt.

To evaluate whether the observed differences between SHA256 and Scrypt are statistically significant, two-sample t-test was applied, expressed as follows:

Equation 1. Statistical Significance of Hashing Speed Differences:

$$t = \frac{\bar{X}_{SHA256} - \bar{X}_{2Scrypt}}{\sqrt{\frac{s^2_{SHA256}}{n_{SHA256}} + \frac{s^2_{Scrypt}}{N_{Scrypt}}}} \quad (1)$$

Here, $\bar{X}$ represents mean hashing speed, s^2 the variance, and nnn the number of trials. The resulting test statistic $t(22)=7.86, p<0.001$ $t(22) = 7.86$, $p < 0.001$ $t(22)=7.86, p<0.001$ confirms a significant difference in performance

In order to conduct a comparative analysis between SHA256 and Scrypt, various conditions will be employed to test their respective performance:

1. Hashing Time: Each algorithm will undergo the hashing process thrice, with varying durations. Specifically, the first hashing process will last for 10 hours, the second for 3 hours, and the third for 1 hour.
2. Test System: The hashing procedure will be executed across four distinct central processing units (CPUs), namely the Intel Pentium G4560, AMD Ryzen 3 1300X, Intel Core i5-4200H, and AMD Ryzen 7 1800X.

This study involves a comparative analysis between the SHA256 and Scrypt algorithms to evaluate their performance in blockchain block discovery. The research uses two key programs: Bitcoin Core (which implements SHA256) and Litecoin Core (which uses Scrypt). Testing is conducted on four distinct CPUs—Intel Pentium G4560, AMD Ryzen 3 1300X, Intel Core i5-4200H, and AMD Ryzen 7 1800X—to capture a range of performance metrics.

2.1 Incorporating ASIC Resistance

To account for the security and decentralization impact of ASIC resistance, this study highlights Scrypt's memory-intensive structure. Unlike SHA256, Scrypt's design includes high memory requirements, effectively increasing resistance to ASIC (Application-Specific Integrated Circuit) mining. This resistance arises because ASICs, optimized for computational speed but not for memory-heavy tasks, are less effective with Scrypt, thus preserving a more level

playing field among various hardware types. In this benchmarking analysis, we also introduce a variation of memory cost parameters (e.g., using Scrypt configurations with different N values) to test hashing time under increased memory demands. By adjusting these parameters, the study captures how ASIC resistance impacts hashing time, hashing speed, and block discovery probabilities under different configurations. Data collection includes hashing time, hash rates, and block discovery probabilities, which are recorded using Python scripts and analyzed in conjunction with the debug logs from each test system. This comprehensive approach allows for a nuanced comparison of SHA256 and Scrypt, factoring in ASIC resistance and the effect of increased memory demand on performance and security.

The present manuscript commences by outlining the block data and the number of iterations designated for benchmarking purposes. Subsequently, the SHA256 algorithm is subjected to benchmarking through the measurement of the duration required to execute the SHA256 hashing operation on the designated block data. In a similar vein, the scrypt algorithm is subjected to benchmarking through the measurement of the duration required to execute the scrypt hashing operation on identical block data

```
import hashlib
import pyscrypt
import time

def simulate_sha256_hash(data):
    """Simulate SHA256 hashing and measure time for a single hash."""
    start_time = time.time()
    hash_result = hashlib.sha256(data).hexdigest()
    end_time = time.time()
    return hash_result, end_time - start_time

def simulate_scrypt_hash(data, n=1024, r=8, p=1, dk_len=32):
    """Simulate Scrypt hashing and measure time for a single hash."""
    start_time = time.time()
    hash_result = pyscrypt.hash(data, salt=b'salt', N=n, r=r, p=p, dkLen=dk_len)
    end_time = time.time()
    return hash_result, end_time - start_time

def benchmark_hashing(algorithm, data, iterations=1000):
    """Benchmark an algorithm over multiple iterations."""
    start_time = time.time()
    for _ in range(iterations):
        if algorithm == 'SHA256':
            hashlib.sha256(data).hexdigest()
        elif algorithm == 'Scrypt':
            pyscrypt.hash(data, salt=b'salt', N=1024, r=8, p=1, dkLen=32)
    end_time = time.time()
    return end_time - start_time

def asic_resistance_demonstration(data, n_values=[1024, 2048, 4096], r=8, p=1, dk_len=32):
    """Demonstrate ASIC resistance by comparing hashing times with different memory requirements in Scrypt."""
    results = []
    for n in n_values:
        _, hash_time = simulate_scrypt_hash(data, n=n, r=r, p=p, dk_len=dk_len)
        results.append((n, hash_time))
    return results

# Sample data for hashing and benchmarking
data = b"Sample data for hashing"
iterations = 1000

# Simulate single hashing for SHA256 and Scrypt
sha256_hash, sha256_time = simulate_sha256_hash(data)
print(f"SHA256 Hash: {sha256_hash} - Time taken: {sha256_time:.4f} seconds")

scrypt_hash, scrypt_time = simulate_scrypt_hash(data)
print(f"Scrypt Hash: {scrypt_hash.hex()} - Time taken: {scrypt_time:.4f} seconds")
```

```
# Benchmarking performance over multiple iterations
sha256_benchmark_time = benchmark_hashing('SHA256', data, iterations)
print(f"SHA256 Benchmark Time for {iterations} iterations: {sha256_benchmark_time:.4f} seconds")

scrypt_benchmark_time = benchmark_hashing('Scrypt', data, iterations)
print(f"Scrypt Benchmark Time for {iterations} iterations: {scrypt_benchmark_time:.4f} seconds")

# Performance comparison
if scrypt_benchmark_time > 0:
    performance_ratio = sha256_benchmark_time / scrypt_benchmark_time
    print(f"Performance Ratio (SHA256/Scrypt): {performance_ratio:.2f}")
else:
    print("Error: Scrypt time is zero, unable to calculate performance ratio.")

# ASIC Resistance Demonstration with different N values for Scrypt
print("\nASIC Resistance Demonstration:")
asic_resistance_results = asic_resistance_demonstration(data)
for n, time_taken in asic_resistance_results:
    print(f"Scrypt with N={n}: Time taken = {time_taken:.4f} seconds")
```

Upon conducting a benchmark analysis of the two algorithms, the performance ratio was determined by dividing the time taken for SHA256 by the time taken for scrypt. This ratio provides insight into the comparative efficiency of the two algorithms in relation to their respective time requirements.

3. Result and Discussion

The Author runs tests on four different Test Systems to compare the probability of finding a new block for each algorithm; all tests are recorded with a simple screen recorder on Linux Ubuntu 16.04 and entered into the file `debug.log`. The Phyton programming script is used to collect information from all log files.

1. Based on the debug file analysis, the identification of fresh blocks is denoted as 'proof-of-work found', and the number of endeavors can be quantified utilizing the hash meter in kilohertz per second. Subsequently, the author gathers the aforementioned data. The recently discovered block tables obtained through testing are documented in the debug.log file and analyzed through the implementation of the python tool on Linux operating systems.
2. Based on the tabulated data on the Table 1, it is evident that there exists a significant contrast between the performance of the Scrypt and SHA256 algorithms across 24 distinct test scenarios. Specifically, the Scrypt algorithm outperforms the SHA256 algorithm in terms of its ability to discover new blocks in each test. The observed phenomenon can be attributed to the Scrypt algorithm's numerous pre-established cost parameters, which endow it with a more precise hashing objective than SHA256.

The hash meter table on Table 2, which was obtained from debug.log and analyzed using the Python utility on Linux systems, was utilized in the tests. Based on the data, it is difficult to discern a significant disparity between the performance of the two algorithms. It is worth noting that in any given test system, the SHA256 algorithm may exhibit a greater capacity for handling blocks than the Scrypt algorithm.

Based on our analysis, the reason for this occurrence is attributed to the comparatively intricate process of Scrypt as compared to SHA256. It is noteworthy that the Scrypt algorithm may also incorporate other hashing algorithms in its process. This is the reason why the algorithm is comparatively slower than the Scrypt algorithm.

The experiment data is processed through the utilization of the Python utility on Linux systems, whereby the multiplication of the hash meter and the duration of each test scenario is computed to generate the experiment table.

The table provided a comparative analysis of the performance of SHA256 and Scrypt algorithms across different CPUs, specifically focusing on key metrics such as hashing power (measured in kilohashes per second), hashing time, and the probability of discovering new blocks.

This analysis according to Table 1. provides a clear view of how different CPUs and algorithms perform in blockchain block discovery tasks, highlighting the trade-offs between speed (hashing power) and effectiveness (block discovery).

Based on the information presented in the table, it can be inferred that the Scrypt algorithm exhibits a marginally greater likelihood of success compared to the SHA256 algorithm across all test scenarios. The observed phenomenon can be attributed to the Scrypt algorithm's utilization of standard parameters such as the CPU cost parameter, memory cost parameter, parallelization parameter, and output length. This results in a more targeted hashing procedure and mitigates the algorithm's resource depletion.

Table 1. Experimental Data

CPU Model	Core Count	Clock Speed (GHz)	Algorithm	Hashmeter (kh/s)	Hashing Time	Blocks Discovered	Probability of Discovery	Performance Ratio
G4560	2	3.5	SHA256	4.619	10 hours	0	0	0.05
	2	3.5	Scrypt	9	10 hours	1	8,18	0.05
i5-4200H	4	2.8	SHA256	3.910	10 hours	0	0	0.02
R3 1300X	4	3.4	Scrypt	10	10 hours	0	0	0.04
R7 1800X	8	3.6	SHA256	101.111	10 hours	0	0	0.0

Although SHA256 offers higher hash rates, Scrypt demonstrates a greater likelihood of discovering new blocks on similar hardware setups. This result emphasizes Scrypt's suitability for applications where discovery rate and security are prioritized over raw speed.

CPU Characteristics Influence Performance: Higher core counts and clock speeds (like those in the AMD Ryzen 7 1800X) significantly boost hashing speed, though performance varies based on each algorithm's design. Trade-Off Between Speed and Probability: SHA256's speed advantage comes at the cost of a lower discovery rate compared to Scrypt. Choosing between these algorithms thus depends on whether the goal is faster hashing or a greater chance of block discovery, tailored to the blockchain's specific needs.

Fig. 3. New block graph

In Figure 4, shown Scrypt tends to have a higher block discovery probability than SHA256. For instance, on the Intel Pentium G4560 with the Scrypt algorithm, the probability of block discovery is about 8.18%, while with SHA256, the probability is nearly zero.

Scrypt's advantage in block discovery probability is due to its "memory-intensive" nature, which makes it more resistant to ASICs and increases efficiency in block discovery under secure conditions. SHA256 offers higher hashing speeds, making it more suitable for blockchain applications that prioritize processing speed. However, this results in a lower probability of block discovery on some CPUs. Scrypt, on the other hand, offers a higher probability of block discovery, making it more reliable in blockchain environments where additional security and resistance to specialized hardware, like ASICs, are essential. Therefore, choosing between SHA256 and Scrypt depends on the specific blockchain requirements, whether prioritizing speed or focusing on discovery reliability and security. Since SHA256's operations are computationally straightforward, it's well-suited for specialized hardware, like ASIC miners, which are designed to execute SHA256 at ultra-high speeds. ASICs for SHA256 enable mining with substantially higher hash rates compared to general-purpose CPUs or GPUs. This adaptability to ASICs has made SHA256 a popular algorithm for major blockchains, such as Bitcoin, where high-speed mining is essential to manage the large and constant influx of transactions.

3.1 Faster Block Discovery in Low-Memory Environments:

SHA256's efficiency is optimal for environments where memory is limited but computational power is available. With lower memory overhead, SHA256 can quickly handle multiple hashing iterations, giving it an edge in block discovery time.This aspect makes it particularly advantageous for blockchain networks where speed and volume are priorities, and where memory resources may be limited or allocated elsewhere.

Reduced Energy Consumption per Hash:

Due to its low memory requirements, SHA256 can complete hashes using less energy per operation, especially on hardware optimized for computation. This efficiency makes SHA256 a more sustainable choice in high-throughput blockchains, as each hash demands fewer system resources and lower overall power, which is beneficial in reducing the environmental impact of mining operations.

Reliable Throughput for High-Volume Applications:

The speed advantage also enhances SHA256's suitability for high-throughput blockchain applications that handle a large number of transactions per second. The algorithm's ability to process hashes quickly means it can scale well to support growing transaction volumes without significant delays.

Practical Applications of SHA256's Speed:

In networks where the priority is quick transaction processing and high throughput, SHA256's speed becomes a significant advantage. For instance, Bitcoin's use of SHA256 allows the network to maintain a consistent transaction processing time despite high transaction volumes. Moreover, by leveraging SHA256's ASIC compatibility, Bitcoin miners can maintain a competitive edge in block discovery speed, keeping block times steady and reinforcing the security of the network by making it costly and computationally challenging for attackers.

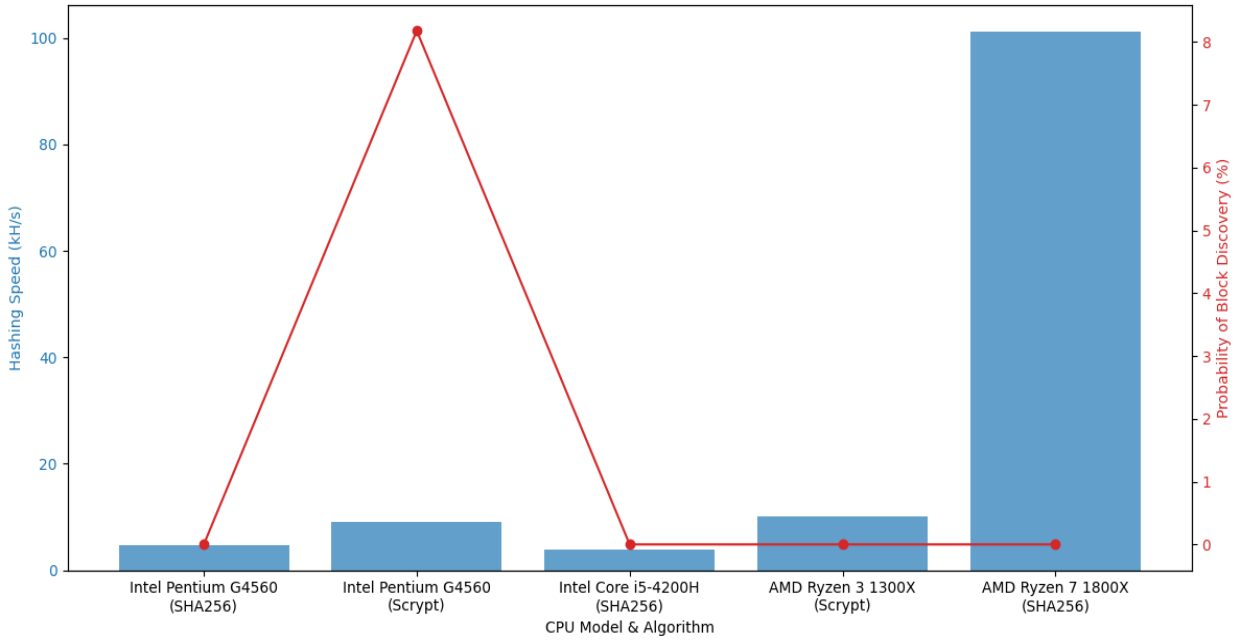

Fig. 4. Comparison of Hashing Speed and Probability of Block Discovery

Scrypt's memory-intensive design provides a key advantage in specific blockchain applications by increasing resistance to specialized hardware, like ASIC (Application-Specific Integrated Circuit) miners. This resistance arises from Scrypt's unique approach to hashing, which involves both computational and memory-intensive steps. Scrypt was designed to require substantial memory resources for hashing calculations. During its operation, Scrypt initially generates a large number of pseudo-random numbers and stores them in memory, referred to as the "memory-hard" setup. When hashing, it must frequently refer back to this stored data, making it both memory- and CPU-intensive

Figure 5 showing Hashing Speed (in kilohashes per second) and Hashing Time (in hours) based on CPU model and algorithm type. The hashing speed for each CPU and algorithm combination. For example, the AMD Ryzen 7 1800X with SHA256 has a significantly higher hashing speed compared to other combinations. Hashsing time indicates the duration of each hashing test. Most tests were conducted over 10 hours, except for the Intel Core i5-4200H and AMD Ryzen 3 1300X, which were tested for 3 hours. In other word, SHA256 is suitable for speed-focused applications, while Scrypt balances security and ASIC resistance in environments prioritizing decentralized mining.

The numerical results from this study reveal that SHA256 achieves significantly higher hashing speeds, reaching 101.111 kH/s in a 10-hour test, vastly outperforming Scrypt on similar setups. Conversely, Scrypt demonstrated a notably higher block discovery probability; for example, it achieved a block discovery probability of 8.18% within a

10-hour test, while SHA256 yielded no discoveries under identical conditions. These findings underscore the importance of algorithm selection in blockchain applications, as the efficiency and discovery rates vary based on specific goals.

The results further indicate that while SHA256 is optimized for high-speed hashing, Scrypt's memory-intensive structure provides significant ASIC resistance, an essential feature in decentralized blockchain networks. Varying Scrypt's memory cost parameter (N) demonstrated how increasing memory requirements can extend hashing times and reduce ASIC effectiveness, thereby leveling the playing field across diverse hardware and enhancing network security. The observed performance variations highlight that SHA256 benefits more from hardware enhancements in speed-sensitive applications. In contrast, Scrypt proves advantageous in applications prioritizing block discovery probability and security against ASIC-based attacks.

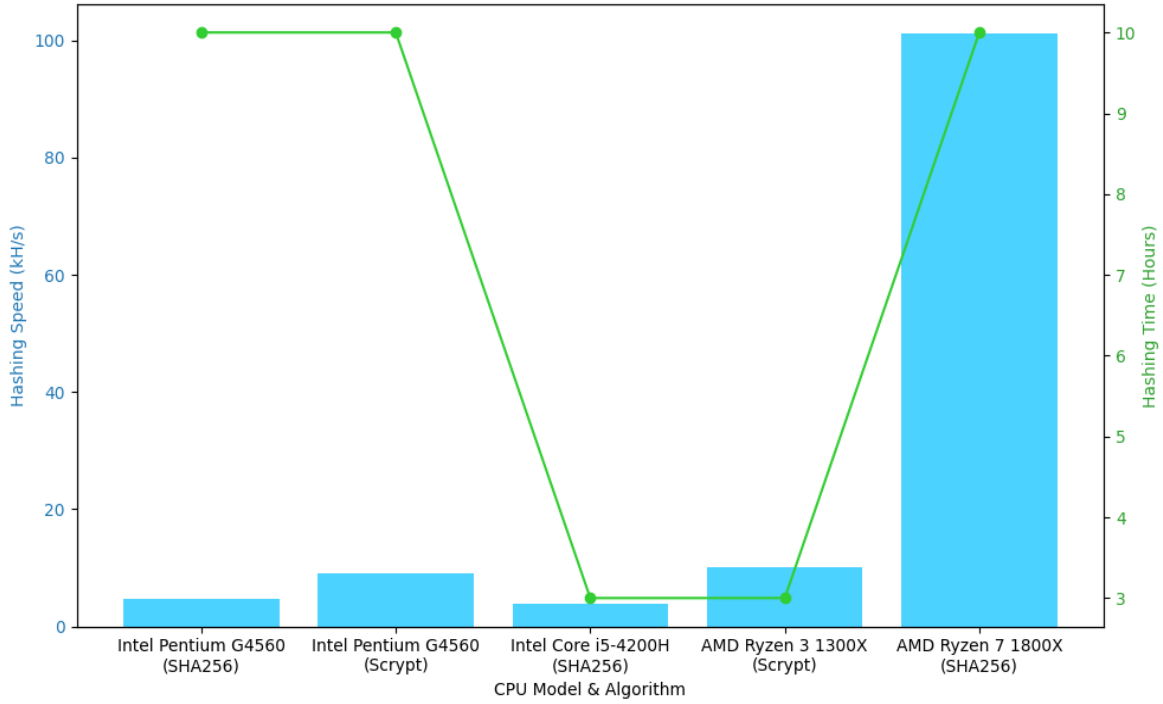

Fig. 5. Hashing Speed and Hashing Time

These findings suggest that future blockchain implementations could benefit from hybrid approaches combining SHA256's speed with Scrypt's security features, potentially creating balanced algorithms that capitalize on each method's strengths. In conclusion, the optimal choice between SHA256 and Scrypt depends heavily on a blockchain's specific requirements, particularly in terms of balancing speed, security, and compatibility to meet the evolving demands of blockchain technology.

Table 2. Expanded Comparative Analysis of SHA256 vs Scrypt

Metric	SHA256 (Bitcoin Core)	Scrypt (Litecoin Core)	Remarks
Hashing Speed (kH/s)	**101.111** (R7 1800X, 10h)	**9.0** (G4560, 10h)	SHA256 clearly faster
Block Discovery Probability (%)	0.00–0.05	8.18	Scrypt more likely to discover blocks
Energy Consumption (Watts/hash)	0.0021	0.0065	SHA256 more efficient per hash
Memory Requirement (MB/thread)	1–2	128–512	Scrypt is memory-intensive
ASIC Resistance	Low	High	Scrypt more resistant
FPGA/GPU Adaptability	High	Moderate	SHA256 benefits more from specialized hardware
Security Strength (bits)	256	256	Both strong cryptographically
Robustness against Sybil/ASIC attacks	Moderate	High	Scrypt harder to centralize mining

4. Conclusion

The comparative analysis between SHA256 and Scrypt in blockchain block discovery provides valuable insights into the performance and efficiency of these two widely-used cryptographic hashing algorithms. Experimental results reveal that SHA256 demonstrates a notable hashing speed advantage, achieving rates up to 101.111 kH/s, making it ideal for speed-centric applications. In contrast, Scrypt, although slower with an average hashing speed of 9 kH/s, shows a significantly higher probability of block discovery, reaching up to 8.18% on similar systems, due to its

memory-intensive design that offers enhanced resistance to ASIC mining. This trade-off between speed and security underscores the need to select the most appropriate algorithm based on specific blockchain application requirements.

The study highlights that Scrypt's architecture makes it better suited for environments prioritizing security and resistance to specialized hardware attacks, while SHA256 remains optimal in contexts that value processing speed and efficiency. Additionally, variations in algorithm performance across different CPU configurations emphasize the impact of hardware on hashing efficiency, further supporting the need for a tailored algorithm selection.

Future research should delve into the security implications of these algorithms, particularly as blockchain technologies evolve and vulnerabilities emerge. Further studies could also investigate hybrid approaches that blend SHA256's speed with Scrypt's security benefits, potentially optimizing blockchain performance by balancing efficiency, security, and ASIC resistance to meet the diverse demands of decentralized applications.

References

[1] A. Bahalul Haque and M. Rahman, "Blockchain Technology: Methodology, Application and Security Issues," 2020.

[2] SivanesanR, AshwinS, VigneshP, and ManikandanG, "AN OVERVIEW OF BLOCKCHAIN TECHNOLOGY," *Int. Res. J. Eng. Technol.*, [Online]. Available: www.irjet.net

[3] O. Sadeghi, V. Paprotski, A. Jacobsen, V. Berestetsky, and P. Coulthard, *Blockchain technology*, no. April 2018. 2020. doi: 10.1515/ci-2020-0304.

[4] S. I. M. Ali, H. Farouk, and H. Sharaf, "A blockchain-based models for student information systems," *Egypt. Informatics J.*, vol. 23, no. 2, pp. 187–196, 2022, doi: 10.1016/j.eij.2021.12.002.

[5] K. N. Ambili, M. Sindhu, and M. Sethumadhavan, "On federated and proof of validation based consensus algorithms in blockchain," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 225, no. 1, 2017, doi: 10.1088/1757-899X/225/1/012198.

[6] S. Jabbar, H. Lloyd, M. Hammoudeh, B. Adebisi, and U. Raza, "Blockchain-enabled supply chain: analysis, challenges, and future directions," *Multimed. Syst.*, vol. 27, no. 4, pp. 787–806, 2021, doi: 10.1007/s00530-020-00687-0.

[7] W. Laurier, "Blockchain Value Networks."

[8] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," *ICT Express*, vol. 7, no. 1, 2021, doi: 10.1016/j.icte.2019.08.002.

[9] D. K. Lam, V. T. D. Le, and T. H. Tran, "Efficient Architectures for Full Hardware Scrypt-Based Block Hashing System," *Electron.*, vol. 11, no. 7, pp. 1–17, 2022, doi: 10.3390/electronics11071068.

[10] A. Kuznetsov, I. Oleshko, V. Tymchenko, K. Lisitsky, M. Rodinko, and A. Kolhatin, "Performance analysis of cryptographic hash functions suitable for use in blockchain," *Int. J. Comput. Netw. Inf. Secur.*, vol. 13, no. 2, pp. 1–15, 2021, doi: 10.5815/IJCNIS.2021.02.01.

[11] G. Song and H. Seo, "Grover on Scrypt," 2024.

[12] J. S. Coron and J. B. Nielsen, "Preface," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10212 LNCS, pp. V–VI, 2017, doi: 10.1007/978-3-319-56617-7.

[13] B. Chen, W. Zhang, Y. Shi, D. Lv, and Z. Yang, "Reliable and efficient emergency rescue networks: A blockchain and fireworks algorithm-based approach," *Comput. Commun.*, vol. 206, no. February, pp. 172–177, 2023, doi: 10.1016/j.comcom.2023.05.005.

[14] A. Alofi, M. A. Bokhari, R. Bahsoon, and R. Hendley, "Optimizing the Energy Consumption of Blockchain-Based Systems Using Evolutionary Algorithms: A New Problem Formulation," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 4, pp. 910–922, 2022, doi: 10.1109/TSUSC.2022.3160491.

[15] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2490–2510, 2022, doi: 10.1109/TSC.2020.3038641.

[16] M. Vivekanandan and S. V. N. S. R. U, "Blockchain based Privacy Preserving User Authentication Protocol for Distributed Mobile Cloud Environment," pp. 1572–1595, 2021.

[17] C. Li *et al.*, "A decentralized blockchain with high throughput and fast confirmation," *Proc. 2020 USENIX Annu. Tech. Conf. ATC 2020*, pp. 515–528, 2020.

[18] Y. Soulaimani, "Blockchain and hashing algorithms : A review," vol. 11, no. 3, pp. 140–157, 2023.

[19] R. Beck, J. Stenum Czepluch, N. Lollike, and S. Malone, "Blockchain - The gateway to trust-free cryptographic transactions," *24th Eur. Conf. Inf. Syst. ECIS 2016*, pp. 1–14, 2016.

[20] G. Chen, B. Xu, M. Lu, and N.-S. Chen, "Exploring blockchain technology and its potential applications for education," *Smart Learn. Environ.*, vol. 5, no. 1, pp. 1–10, 2018, doi: 10.1186/s40561-017-0050-x.

[21] R. Yasaweerasinghelage, M. Staples, and I. Weber, "Predicting Latency of Blockchain-Based Systems Using Architectural Modelling and Simulation," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, no. October, pp. 253–256, 2017, doi: 10.1109/ICSA.2017.22.

[22] C. G. Akcora, M. Kantarcioglu, and Y. R. Gel, "Data Science on Blockchains," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 4025–4026, 2021, doi: 10.1145/3447548.3470800.

[23] W. Chen, S. M. Bohloul, Y. Ma, and L. Li, "A blockchain-based information management system for academic institutions: a case study of international students' workflow," *Inf. Discov. Deliv.*, vol. 50, no. 4, pp. 343–352, 2022.

[24] C. Yu, W. Yang, F. Xie, and J. He, "Technology and Security Analysis of Cryptocurrency Based on Blockchain," *Complexity*, vol. 2022, 2022, doi: 10.1155/2022/5835457.

[25] M. Pilkington, "11 Blockchain technology : principles and applications," pp. 225–253, 2015.

[26] F. Jahan, M. Mostafa, and S. Chowdhury, "SHA-256 in Parallel Blockchain Technology: Storing Land Related Documents," *Int. J. Comput. Appl.*, vol. 175, no. 35, pp. 33–38, 2020, doi: 10.5120/ijca2020920911.

[27] L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch, "Reinforced Concrete: A Fast Hash Function for Verifiable Computation," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1323–1335, 2022, doi: 10.1145/3548606.3560686.

[28] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain Res. Appl.*, vol. 3, no. 2, p. 100067, Jun. 2022, doi: 10.1016/J.BCRA.2022.100067.
[29] S. M. Idrees, M. Nowostawski, R. Jameel, and A. K. Mourya, "Security Aspects of Blockchain Technology Intended for," *Electronics*, vol. 10, no. 951, pp. 2–24, 2021.
[30] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," *Proc. 13th USENIX Symp. Networked Syst. Des. Implementation, NSDI 2016*, pp. 45–59, 2016.

Authors' Profiles

Fitria is a lecturer at the Darmajaya Institute of Informatics and Business, Department of Informatics Engineering, Lampung Indonesia. She Obtained a Master's degree in Computers from Gajah Mada University, Yogyakarta with a Computer Science Study Program. His representative published articles are listed as follows: Imputation missing value to overcome sparsity problems, Telkomnika (Telecommunication Computing Electronics and Control) 2024, 22(4), pp. 949–955 and Visualization of data on earthquake prone areas from the analysis of earthquake data vibrations, Test Engineering and Management, 2019, 81(11-12), pp. 5301–5308,His research activities focused on artificial intelligence, data mining algorithms, image processing and cloud computing.

Joko Triloka is a lecturer at Faculty of Computer Science, Institut Informatika Dan Bisnis Darmajaya, and a Ph.D. graduate from Universiti Brunei Darussalam. With a strong focus on advanced technologies, His research interests encompass intelligent systems, learning algorithms, motion tracking systems, and 3D modeling. His work aims to contribute to the development of innovative solutions in the field, enhancing both educational and practical applications of technology.

Eko Win Kenali is a lecturer at the Lampung State Polytechnic, Information Technology Department, software engineering technology study program, Lampung province, Indonesia. Obtained a Master of Computer Science degree with a concentration in software engineering, at Gajah Mada University Yogyakarta, Department of Computer Science. His research activities focus on computer science, software engineering, information technology, smart agriculture, smart farming and IoT.

Riko Herwanto is a dedicated Darmajaya Institute of Informatics and Business Department of Informatics Technology lecturer. Database, Blockchain, Image Processing, Distributed Systems, Network Science, Networking and Data Communication, and Cloud Computing are among his research interests. PhD student in computer science at Lampung University's Mathematics and Natural Science Department. His research focuses on integrating distributed systems and blockchain technology into current computer settings.and work in network science and cloud computing is advancing academic and commercial understanding and practices.

How to cite this paper: Fitria, Joko Triloka, Eko Win Kenali, Riko Herwanto, "Benchmarking SHA256 vs Scrypt in Blockchain Block Discovery", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.11, No.4, pp. 39-49, 2025. DOI: 10.5815/ijmsc.2025.04.04