Modify Block Chain Environment based on Post-quantum Algorithms

Rasha Hani Salman[1,*], Hala Bahjat Abdul Wahab[2]

[1]Informatics Institute for Postgraduate studies, Information Technology & Communication University, Baghdad, Iraq

[2]Computer Sciences Department, University of Technology, Baghdad, Iraq

Email; rsalman@uowasit.edu.iq; Hala.B.AbdulWahab@uotechology.edu.iq

Abstract

Blockchain technology provides reliable data storage and secures transactions, however, is not suitable for devices with low resources because of its high computational and resource requirements. As quantum computing develops, it poses concerns regarding a cryptographic integrity of blockchain, making them more vulnerable to attacks. Blockchain technology is being used to enhance security and performance. The application of the post-quantum Ascon algorithm in a blockchain setting is presented in this paper. The Ascon hashing algorithm offers a lightweight, efficient architecture for resource-constrained applications, including mobile devices or Internet of Things-based blockchains. By providing high-speed hashing, authentication features, and defense against quantum attacks, it enhances performance and guarantees strong security without putting a strain on network infrastructure. The experimental results show using the Ascon algorithm in a blockchain environment is successful in reducing resource usage and execution time and significantly increasing randomness and unpredictability. Post-quantum Ascon algorithms overcome the drawbacks of traditional technologies and ensure that blockchain systems continue to withstand the new risks posed by quantum computing while increasing overall efficiency.

Keywords: Keywords- Lightweight Blockchain; Merkle tree; post-quantum algorithms; Metric measure; Ascon

1. Introduction

Recently, blockchain has become one of the most innovative technologies in the world. From the time it was proposed by Nakamoto in 2008 [1,2], it has evolved into an important technology that enhances the security, scalability, and adaptability of many systems [3,4]. Blockchain, which was originally designed for Bitcoin, is a peer-to-peer electronic cash system that eliminates intermediaries for conducting secure payments based on cryptographic algorithms and decentralized data sharing [5-7]. A state machine processes these transactions and programs the blockchain to perform other tasks, such as executing smart contracts [8, 9]. Many different industries have shown interest in blockchain technology due to its unique qualities and potential integration into IoT ecosystems. [10, 11]. Multiple blocks of transactions are stored in a decentralized database called a blockchain. Each block has a unique identifier, a block hash, and the previous block's hash because they contain the hash of previous blocks, these blocks must be linked. Any illegal changes invalidate subsequent blocks [12, 13]. SHA256 is a secure and reliable algorithm in blockchain technology that produces unique keys for transactions. Its functions ensure a secure and permanent hash. The SHA-256 hash algorithm in blockchain technology is susceptible to quantum computing [14].

Attacks pose a significant risk because quantum algorithms like Shor's can efficiently solve mathematical problems [15, 16]. Quantum computers and need a lot of processing power, so they can't be used in places with limited hardware, like Internet of Things devices [17], could attack Blockchain systems, which use the SHA-256 hash

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

method. Blockchain networks could experience slower processing due to SHA-256 processing needs, potentially affecting transaction verification and cryptographic systems' security given by quantum computing development [14]. Quantum computing threatens the security of hash functions a public key encryption, which are essential, parts of blockchains, as Shor's algorithm compromises their cryptographic foundations [18], which is capable of factoring integers rapidly. In Grover's method, the blockchain community explores post-quantum encryption as a potential solution to counter quantum computer attacks that threaten the hash functions that connect the blockchain blocks [19-21]. The current paper proposes a lightweight hash algorithm designed specifically for blockchain applications. It methodically contrasts two hashing algorithms to determine which best fits the particular needs of the blockchain. The paper assesses performance measures like throughput, elapsed time, and memory. Utilization and time complexity. Many reviews studies on the significance of hashing algorithms in blockchain technology, focusing on SHA-256, a robust cryptographic property, and its challenges, particularly in resource-constrained environments.as discuss below:

In [22], the researcher discussed the growing need for secure data management in blockchain systems, focusing on the SHA256 hashing algorithm. The algorithm improves data integrity, confidentiality, and authentication, ensuring tamper-resistant storage and retrieval. The paper quantifies improvements in security percentages and metrics, offering insights for researchers, developers, and practitioners seeking to enhance data layers in blockchain networks. Although the paper addresses practical consequences, it skips over potential difficulties especially when using the SHA256 algorithm in actual blockchain systems. Crucial issues like processing overhead, scalability, and integration with current systems may affect the viability of the suggested method.

The research in [23] highlighted the growing demand for security and privacy when managing personal information. To address these concerns, blockchain technology can be used for its potential. Blockchain operates by linking data blocks using cryptography, ensuring integrity and security. This research used the SHA256 algorithm to secure each block. The paper mentions the SHA256 algorithm in passing as a method for creating blockchains. It did not, however, examine the algorithm's technological complexities or constraints, such as its processing demands or any susceptibilities to quantum computing.

In [24] the usage of hash functions in cryptographic applications and protocols, including digital signatures and message authentication codes, is covered in the paper. SHA (Secure Hash Algorithm) and MD (Message Digest) are the two hash function kinds that are highlighted; SHA-256 was selected because to its efficiency and security. To improve the verification of data integrity, the Merkle-Damgård building approach is presented. One-way cryptographic hash functions ensure secure authentication without storing passwords by transforming input data into a fixed-length output.

In [25] discuss the integration of blockchain technology with robust hashing algorithms, namely SHA-256, is the focus of this study. Because blockchain systems depend on hashing algorithms to preserve the integrity and confidentiality of data, this focus is essential. The primary objective is to produce and evaluate techniques, which improve the precision and trustworthiness of blockchain-based transactions. For businesses that depend on these networks for financial transactions, this is especially important. The study focuses on SHA (Secure Hash Algorithm) and MD (Message Digest) and, two hash function types frequently employed in certificate data security.

In [26] the significance of security in blockchain technology is emphasized throughout this study, especially when it comes to transactional record storage. Recent assaults have shown that the SHA256 algorithm is susceptible, underscoring the necessity for more robust hash functions. The authors suggest SHA288, a new hash function that cuts down processing rounds from 64 to 44 and produces a 288-bit message digest. It is challenging to reverse-engineer input with SHA288 since it maintains important cryptographic characteristics including collision and preimage resistance. According to the research, SHA288. Performs exceptionally well in random tests and is resilient to attacks.

In order to address security and privacy concerns, this work [27] suggests implementing the most popular blockchain hash algorithm, SHA-256, on a field-programmable gate array (FPGA) to increase processing speed and reduce power consumption in Internet of Things (IoT) devices. This method differs from previous papers in the literature in that it runs the SHA-256 algorithm in parallel using clustered cores. An examination of the FPGA's resource usage and specifics of the suggested architecture are provided.

In [28], the research presented a lightweight and scalable blockchain, known as the Light Block. A key purpose of the study is to enable the use of blockchain's technique in the context of the Internet of Things. We achieve Light Block by optimizing the components of an efficient and lightweight blockchain. Several parameters have been taken into account to evaluate the method—throughput, memory usage, time of execution, latency, and security robustness. However, the selected quantities may not include all significant performance characteristics for use in practical applications. For instance, there is a lack of information on such factors as operating energy consumption.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

In this work, a quantum algorithm that won the Best Lightweight Hashing and Encryption Algorithm award in the NIST 2023 competition is implemented. This research introduces the use of the algorithm for the first time as a hashing algorithm within a blockchain environment, although it was previously used as an encryption algorithm in the same environment. This research aims to provide a lightweight environment for the blockchain environment suitable for resource-constrained applications. In comparison with previous works, most of the hashing algorithms used in it, such SHA 256 are primarily based on computationally intensive operations, making them unsuitable for resource-constrained systems. This research contributes to bridging this gap by presenting an innovative model that combines lightweight and security, adapting it to a range of uses, including the Internet of Things, Micropayments over blockchain and smart Supply Chains as the proposed system is highly efficient in terms of resource consumption. The primary differences between the suggested work and the previous studies are listed in Table 1:

Table 1: Related work summary

study	**Area of Focus**	**Utilizing Hashing Algorithm**	**Criteria for Evaluation**	**Consideration of a Lightweight Blockchain**
[22]	Blockchain data security and integrity	SHA-256	Authentication and security metrics	No particular emphasis on lightweight blockchain
[23]	Privacy and security in the handling of personal data	SHA-256	Security of the blockchain in general	Processing overhead and quantum vulnerabilities were not examined.
[24]	Applications for cryptography (MACs, digital signatures)	MD, SHA-256	Security, Efficiency,	Absence of lightweight blockchain optimization
[25]	Combining secure hashing techniques with blockchain technology	MD, SHA-256	Accuracy and reliability in financial dealings	overall emphasis on security as opposed to low-power performance
[26]	SHA-288 for enhancing blockchain security	SHA-288, SHA-256,	Processing round reduction and security resiliency	Not made especially for lightweight blockchain
[27]	FPGA implementation for blockchain applications in the Internet of Things	SHA-256	Power usage and processing speed	IoT-optimized, although memory use and elapsed time were not thoroughly examined
[28]	Lightweight blockchain for IoT	Custom (Light Block)	Memory utilization, security robustness, throughput, and latency	Lightweight features were taken into account, although energy usage information was lacking.
Proposed work	Lightweight blockchain optimization	Ascon	Time, throughput, memory usage, and elapsed time	Specifically created for lightweight blockchain settings, it addresses resource and efficiency limitations.

2. Background

A. SHA-256 Construction

A hash algorithm known as SHA-2 [29]. There are six hashing algorithms in the SHA2 category: SHA512/256, SHA512/224, SHA384 SHA224, SHA512, and SHA256. Although the word lengths, constant parameters, and initial values are different, the overall procedure is similar. For instance, SHA256, a SHA2 family's member that is extensively used. Decentralized blockchain technology, along with HMAC and DSA for data protection, are the subjects of this publication. To compute the hash value of a message with a length of 512, a SHA256 used. If the message is long, it is divided into blocks. Each with a length of 512, and if the length of the final block is less than 512 bits, its hash value is then determined. Stuffing is included.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

A hash is calculated for a message with a large length. For each data block, The SHA-256 technique is used to obtain an intermediate hash value for it, and to calculate the hash of the following data blocks, the hash value of the previous block is used to calculate the first hash value. The message's entire hash value data is used to compute the final block result, a summary of all the operations of the SHA-256 algorithm where its operations include compressor operations (MC) and message expansion operations (ME). The 512-bit input message is expanded by the ME process into 64 32-bit data Wj chunks, where j is between 0 and 63. During the first 16 cycles, the ME breaks up the 512-bit message into 16 32-bit data chunks, called Wj, where j = 0 to 15. In the last 48 cycles, the ME employs equation (1) to determine 48 pieces of 32-bit data, Wj ($16 \leq j \leq 63$). Three 32-bit adders and two logical functions, σ0(x) and σ10(x), are used to compute Wj ($16 \leq j \leq 63$). σ0(x) and σ1(x) are computed using equations (2) and (3). It should be noted that Sn(x) and the Rn(x) signal are both rotations and that the data n bits successively alter x as follows:

$$wj = \sigma 1\ (wj - 2) + wj - 7 + \sigma(wj - 15) + W \quad (1)$$

$$\sigma 0(x) = S^{7}(x) \oplus S^{18} \oplus R^{3}(x) \quad (2)$$

$$\sigma 1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10} \quad (3)$$

The 64 chunks of W_j ($0 \leq j \leq 63$) output by the ME process are used by the message compression (MC) process to calculate the hash value in 256 bits. The process includes two primary steps: hash updates and loops. The step in the loop initializes 8 loop values of hash a , b , c , d , e , f, g ,h using the values of hash at first H_0, H_1..., H_7. It then computes and updates. The number of loops used is 64, whose fragmentation values are as mentioned above, and equations 4 to 9 are used to express each loops [30, 31] as follows:

$$T1 = \sum 1(e) + wj + \text{h} + \text{Ch (e,f,g)} + \text{kj} \quad (4)$$

$$T2 = \sum 0(a) + Maj\ (a, b, c) \quad (5)$$

$$a = T1 + T2 \quad (6)$$

$$e = d + T1 \quad (7)$$

$$b = a; c = b; d = c; f = e; g = f; h = g \quad (8)$$

The equations below are used to calculate logical functions, including: Maj (x, y, z), Ch (x, y, z), 0(x), 1(x).

$$\sum 0\ (x) = S^{13}\ (x) \oplus S(x) \oplus \text{S\^{}22 (x)} \quad (9)$$

$$\sum\ 1\ (x) = S^{6}(x) \oplus S^{11}(x) \oplus S^{25}(x) \quad (10)$$

$$ch(x, y, z) = (x \wedge y) \oplus (\sim x \wedge z) \quad (11)$$

$$Maj(\ x\ ,\ y\ , z\ ) = (y \wedge z) \oplus (x \wedge z) \oplus (\ \text{x} \wedge \text{y}) \quad (12)$$

The step of updating the hash computes, A 256-bit final hash value will be divided into 8 pieces of 32-bit data: H0, H1..., H7, with the original hashes added H_0, H_1..., H_7 .Equation (13) illustrates how to hash a, b, c, d, e, f, g, and h to the loop:

$$HO_0 = H_0 + a; \ldots;\ H0_7 = H_7 + h \quad (13).$$

B. Post – Quantum Ascon Construction Algorithm

A competition of CAESAR, which sought to develop encryption techniques that are authenticated and appropriate for lightweight devices, gave rise to ASCON. A finalist in the competition garnered praise for its versatility, security, and performance. ASCON also attracted attention as a potential standardizing contender during the NIST competition, winning the rivalry in the lightweight sector [32]. The sponge construction is the hashing mode of operation. Ascon-Hash, which has a fixed output size, internally uses the same hashing mechanism X h, r, a (see Table I).

Table 2: Recommendation for hashing algorithm parameter [32]

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

Name	Algorithm	Bit size of		Round
ASCON	X256, 64, with $\ell = 256$	Hash	Data block	***P2***
		256	64	12

The Ascon algorithm consists of many steps:

Step 1: A constant IV is created to initialize the starting state of the 320-bit Ascon-Hash. The algorithm parameters, similar to those of Ascon, are described by this constant. These variables consist of round number b and a are equal to zero., each of which is an integer with eight bits, k = 0, and a rate r. The h bit is the maximum output length of 32 bits and integer number (for Ascon-Hash, h = 256), is after a 256-bit zero. The state S is initialized using a par round permutation:

$$IV_{h,r,a} \leftarrow r \parallel 0^8 \parallel a \parallel 0^8 \parallel h = \{00400c0000000100 \ for \ ascon - hash\}$$

S← P^a (IV(h ,r ,a) || 0^256)

Recalculating each instance's original 320-bit state S would provide the Ascon-Hash..

S←ee9398aadb67fo3d

8bb21831c6of1oo2

b48a92db98d5da62

43189921b8fe3e8

348fa5cd525e140

Step 2: Absorbing the message

Blocks of r bits are used by Ascon-Hash to process message M. To make the padded message longer than r bits, the padding process adds a single 1 and the fewest feasible 0s to M, just like it does with Ascon's plaintext. We separate the generated padded text into s blocks, with r bits in each block.

$$M_1 \parallel \cdots \parallel M_s$$

$$M_{1,\ldots,}M_s \leftarrow r - bit\ block\ of\ M \parallel 1 \parallel 0^{r-1-(\backslash M \backslash mod\ r)}$$

When processing the message blocks Mi with i = 1..., s, the a-round permutation p^a is applied to S, and each block Mi is then XOR to state S's first r bits, Sr.

$$S \leftarrow P^a((S_r \oplus M_i) \parallel S_c \quad 1 \le i \le S$$

Step 3: Squeezing

Squeezing until the desired output length is reached, the state generates the hash output in r-bit blocks. The completion of blocks t = [ℓ/r] occurs after ℓ ≤ h. The a-round permutation p^a modifies the internal state S following each extraction:

$$Hi \leftarrow S_r$$

$$S \leftarrow P^a(S) \qquad 1 \le i \le t = [\ell/r]$$

H = H1ll …ll H't is returned, unless r divides ℓ, it is minimized to ℓ mod r bit in the last output block:

$$H_{t \leftarrow}[H_t]_{\ell\ mod\ r}$$

The Ascon method's primary components are the two 320-bit permutations, p^a and p^b. The SPN based round transformation p, which has three stages, is applied iteratively by the permutations: PC, PS, and PL:

$$P = PL \circ PS \circ PC$$

The only distinction between P^b and P^a is the number of rounds. Two security factors that can be altered to define and execute the round are the number of rounds (a) and (b).

Table 3: Specification for hashing algorithm [32]

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

P^{12}	P^{8}	P^{6}	Constant Cr	P^{12}	P_{8}	P_{6}	Constant Cr
0			00000000000000f0	6	2	0	0000000000000096
1			00000000000000e1	7	3	1	0000000000000087
2			00000000000000d2	8	4	2	0000000000000078
3			00000000000000c3	9	5	3	0000000000000069
4	0		00000000000000b4	10	6	4	000000000000005a
5	1		00000000000000a5	11	7	5	

Including constants, in round I, the state S's register word X2 receives a round constant Cr from the constant addition step PC. We utilize For P^b, r = i + a - b, while for P^a, r = i. Both the r and i indices begin at zero, per Table2:

$X_2 \leftarrow X_2 \oplus C_r$

The 64 concurrent applications of the 5-bit S-box S(x) are used to carry out the replacement layer. The substitution layer Ps updates the state S for each bit-slice of the five registers x0...x4. Typically, operations are performed on the entire 64-bit words in this bit-sliced form, as seen in the sample code. modifications, the 320-bit state S is divided into five 64-bit registers, or words xi: $S = X_0 \parallel X_1 \parallel X_2 \parallel X_3 \parallel X_4$

Constants are added in round I; the round constant Cr is added by the constant addition step PC to the register word x2 of the state S. The indices r and i both begin at zero, and for p^a and p^b, we utilize r = i and r = a − b, respectively (see Table 2) [32].

The PL layer provides a layer of linear diffusion for every 64-bit register word Xi. We use the linear function i (Xi) for every word Xi [33][34]:

$$X_i \leftarrow \sum i(xi), \quad 0 \leq i \leq 4$$

C. Comparison the Performance of the Ascon and SHA256 Algorithms

The SHA-256 and ASCON hashing algorithms are thoroughly compared in this section. The comparison was made using the data and analysis from the preceding section.

Table 4: Compilation of the main performance indicators of SHA-256 with Ascon

Measure	**SHA-256**	**ASCON**	**observation**
Size	Moderate	Small (lightweight)	Faster processing, especially for lower input sizes, is made possible by ASCON's lightweight design.
Strength	Equivalent	Equivalent	ASCON can withstand threats from quantum technology.
Energy Use	Baseline	20% less expensive than SHA-256	Because ASCON uses less energy, it is better for battery-operated and Internet of Things systems.
Memory Usage	32	16 KB	ASCON uses less memory, making it suitable for resource-constrained environments.
Each Hash Round	64	12	Its reduced number of rounds enhances the computational efficiency of ASCON.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

Time of Access	0.8 ms	0.5 ms	ASCON shows reduced latency in single-operation hashing.
Rate of Transfer	900 MB/s	1.1 GB/s	Higher throughput due to fewer computational steps in ASCON's algorithm.
Speed of Hashing	850 MB/s	1.2 GB/s	Faster processing is made possible by ASCON's lightweight design, especially for smaller input sizes.

Table 4 show that Ascon performs better than SHA-256 in terms of resource usage, energy consumption, and efficiency. Its design is lightweight, its energy consumption is 20% lower, its memory requirements are 16 KB, and its latency is 0.5 ms. It is also perfect for modern applications and systems with limited resources because it provides greater transfer rates (1.1 GB/s) and hashing speeds (1.2 GB/s)

3. Proposed system

A. Light weight proposed system model

Blockchain technology provides reliable data storage and secure transactions; however, traditional hashing algorithms, such as SHA-256, limit it. Excessive usage of memory and other resources characterize these algorithms, which compromises privacy and security. The proposed system uses lightweight post-quantum hashing in the blockchain as an alternative to traditional hashing algorithms; this is the primary phase in the suggested blockchain construction scheme. The suggested lightweight blockchain is mostly made up of multiple stages, as seen in Figure 1.

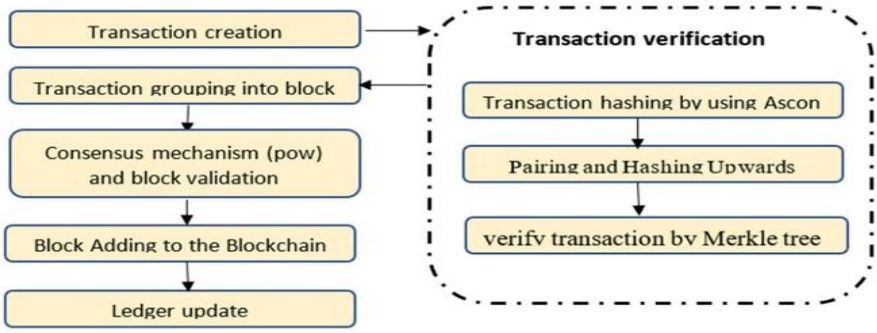

Figure 1. Lightweight blockchain proposed system.

Phase 1: Creation of the transaction

A user (or node) starts a blockchain transaction that transfers data. Sender and recipient information, as well as the transaction amount, are frequently broadcast to the network as part of the transaction data.

Phase 2: Transaction Verification

Each transaction is hashed during creation to generate a unique transaction identifier, thus making it easily recognizable, similar to a "fingerprint.". This is a hash of the transaction, and since nodes check these hashes to validate the integrity of the data, it is often used to prove that data has not been changed.

In the proposed system case, In order to meet the needs of a blockchain that is lightweight and provides protection against quantum computing attack techniques, the Ascon algorithm is used in the hashing process, creating a future-proof blockchain.

Phase 3: Transaction grouping into a block

A blockchain data structure called a Merkle Tree addresses the grouping of transactional records through blocks. It involves transaction hashing whereby each transaction is assigned as a digital fingerprint. The process of pairing and hashing upwards is repeated until a Merkle root is reached. This efficient verification system keeps the block header updated securely and efficiently.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

Phase 4: Mechanism of Consensus

The proposed system's structure uses POW, a blockchain consensus technique, to validate blocks. Miners need to solve a cryptographic issue; it requires computational effort for security hashing efforts and involves locating a value of the block header whose hash is smaller than a specific predefined number.

Phase 5: Block Adding to the Blockchain

After being validated, the block is appended to the series of blocks that have already been verified, creating an unchangeable, continuous ledger. To ensure the chain's integrity, the new block is cryptographically hashed to the prior block.

Phase 6: Updates to ledger

All the updates are sent to every node to reach the consistency of ledger files. The most recent version of the ledger that distributed will be available on every node. The transactions in block are regarded as final once they are uploaded to the blockchain. Multiple confirmations, or fresh blocks put on top of the block, further protect the transaction in the majority of blockchains.

B. Data set

The system utilizes the medical data [35], hashed using the Ascon algorithm and stored within the blockchain. The medical datasets consist of electronically generated medical records for a group of hypothetical patients. The records include patient IDs, names, dates of birth, genders, medical problems, prescriptions, and allergies, last names, and dates of the appointments. Dataset It is essential to test the selected algorithms on the dataset to assess their performance in a scenario and data distributions. It assesses the extent of generalizing the results and identifies if there are any biases or limitations. [36] In a practical scenario, the system employing such an algorithm is made more reliable because it does not focus on specific features of data. Therefore, using a dataset helps to get a more or less accurate understanding of the system's advantages, flaws, and potential.

4. Expert result and decision

The suggested system refers to an energy-efficient blockchain that is lightweight and secure. This method improves certain features that characterize the functionality of a blockchain, such as the processing speed, memory usage, Throughput, Latency, Elapsed time by combining the original blockchain with Ascon's lightweight hashing algorithm. Table 5 compares the traditional blockchain system with the novel lightweight blockchain based on Ascon's algorithm.

Table 5 Evaluation of the original system versus the suggested system based on medical datasets

Metric measure	Traditional block chain (SHA256)	Lightweight block chain (ASCON)
Time	0:00:26.739767	0:00:23.306198
Memory usage	22067	9432
Throughput	60.184506	69.0590
Latency	0.00830	0.007240
Elapsed time	26.742763	23.6590

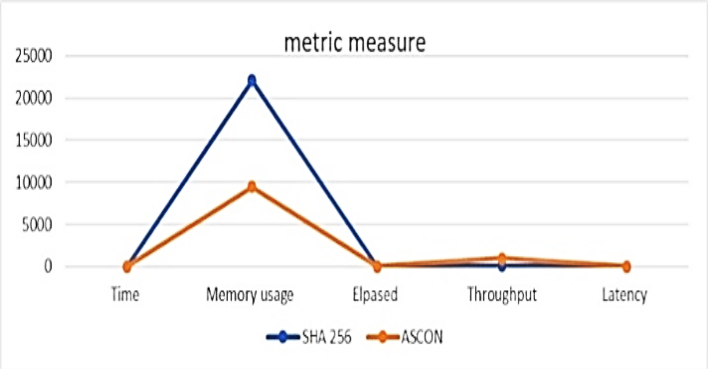

Figure 2. Evaluation of the original system versus the suggested system based on medical datasets.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

Table 5 and Figure 2 highlight the following observations:

By cutting down the time of creating blocks and enhancing system effectiveness, Ascon enhances the speed of blockchains by 13% greater than SHA-256. Due to the lightweight, it can be effectively used in environments with limited memory, which is the case for mobile applications and blockchains based on the Internet of Things. Ascon outperforms SHA-256 in the high-load use of blockchain applications with more transactions or data simultaneously, with a throughput that is greater by 14.7%. Furthermore, it ensures lower latency for quicker responses, which is essential for real-time applications like micropayments and high frequency trading. Ascon provides notable timesavings with an estimated 11.5% reduction in the total amount of time spent on extended blockchain activities. Because of these improvements, Ascon is a better option for high-demand blockchain scenarios, allowing for the effective management of growing transaction and data volumes.

The proposed system employs a variety of criteria to assess the efficacy of two hashing algorithms within a blockchain setting. These criteria, which address distribution, dependency, sensitivity to change, and similarity of the overall output, provide a multi-faceted evaluation of hashing algorithms. As evidenced in Table V below, when combined, they can help determine the best algorithm for a given application, ensuring maximum performance in areas such as data integrity, collision resistance, and cryptographic security.

Table 6: Demonstrating the correlation coefficient, mean square error based on medical dataset

Metric measures	**Result**
Correlation coefficient	0.00167
Mean Square Error distance	1046.0920
Levenshtein -Distance	168644
RMSE-Coefficient _determination	32.3443
Hamming- distance	543413
Bray -Curtis-Dissimilarity	0.16880

Table6 shows how the correlation coefficient is used to measure the linear relationship between the hash outputs of two algorithms. Its value of 0.00167, which is near zero and indicates nearly no linear connection, improves security by preventing the results of one algorithm from being predicted from those of the other. To reduce collisions in cryptographic hashing, the MSE—, which represents the average squared differences between the methods' comparable outputs—highlights notable numerical variances. Significant character-level variations and the algorithms' sensitivity to input changes are highlighted by the high value of 168644 for the Levenshtein distance, which counts the number of single-character changes required to change one hash output into another. This ensures that hash outputs are both unique and long lasting. With a value of 32.3443, RMSE, a scaled version of MSE, provides output deviation in the same units as the original data. The combination of high MSE and moderate RMSE confirms the uniqueness and consistency of the outputs within a quantifiable range, bolstering the algorithms' dependability. The algorithms' hash output distributions are similar due to the low Bray-Curtis dissimilarity. The distributed values are evenly located in the hash space, which minimizes the possibility of collisions and ensures maximum distribution throughout the chain, which is a great quality while avoiding bottlenecks in blockchain or hash-based solutions.

5. Conclusion and future scope

Blockchain systems rely on cryptographic hash functions to guarantee data integrity and security. In this work, a hash function is utilized within a blockchain environment for the first time despite being used as a cryptographic algorithm in the environment itself, which is one of the quantum algorithms (ASCON) that won the best algorithm award in the competitive competition organized by NIST. This approach seeks to provide a lightweight blockchain environment suitable for resource-constrained applications. To improve the robustness and performance of blockchain technology. Ascon offers several benefits over the traditional SHA-256 hash function, such as increased network security, long-term defense against quantum computing risks, and improved security and efficiency. The greater flexibility of Ascon in resource-constrained contexts is demonstrated by a comparative analysis between Ascon and SHA-256 that evaluates their performance across important metrics, such as time complexity, memory consumption, computational efficiency, throughput, and latency. Furthermore, the simplified

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

design of Ascon increases blockchain utilization by reducing memory usage and transaction times. Metrics such as Bray– curtis divergence, correlation coefficient, mean square error, cosine similarity, and Hamming distance are used in cryptographic evaluations to demonstrate that Ascon is capable of generating highly secure and unpredictable hash outputs, making it suitable for secure blockchain applications. This study can be expanded in the future by leveraging artificial intelligence to accelerate solution discovery and incorporating quantum-resistant technologies to enhance algorithm security and conduct more comprehensive evaluations in real-world environments.

References

[1] M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and Dapps*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," 2014. [Online]. Available: https://ethereum.org/en/whitepaper/

[4] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[5] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[6] Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. New York, NY, USA: Penguin, 2016.

[7] K. Wüst and A. Gervais, "Do you need a Blockchain?," in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, Switzerland, 2018, pp. 45-54.

[8] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," *Applied Innovation Review*, no. 2, pp. 6-19, 2016.

[9] S. Underwood, "Blockchain Beyond Bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15-17, 2016.

[10] Bashir, *Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications*. Birmingham, UK: Packt Publishing, 2017.

[11] M. Pilkington, "Blockchain Technology: Principles and Applications," in *Research Handbook on Digital Transformations*, F. Xavier Olleros and M. Zhegu, Eds. Cheltenham, UK: Edward Elgar Publishing, 2016, pp. 225-253.

[12] R. Hani, "Using Lotka-Volterra Equations and Lightweight Post-Quantum Algorithm to Develop Lightweight Blockchain Security," 2023. [Online]. Available: https://www.researchgate.net/publication/389352224_Using_Lotka-Volterra_Equations_and_Lightweight_Post-Quantum_Algorithm_to_Develop_Lightweight_Blockchain_Security

[13] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 2012. [Online]. Available: https://peercoin.net/assets/paper/peercoin-paper.pdf

[14] Biryukov and D. Khovratovich, "Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem," in *Proceedings of the 2016 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 2016.

[15] Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in *Proceedings of the 12th Annual International Cryptology Conference (CRYPTO)*, Santa Barbara, CA, USA, 1992, pp. 139-147.

[16] Juels and J. Brainard, "Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks," in *Proceedings of the 1999 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 1999, pp. 151-165.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025

[17] S. Popov, "The Tangle," 2018. [Online]. Available: https://iota.org/IOTA_Whitepaper.pdf

[18] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI)*, New Orleans, LA, USA, 1999, pp. 173-186.

[19] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382-401, 1982.

[20] L. Lamport, "Paxos Made Simple," *ACM SIGACT News*, vol. 32, no. 4, pp. 18-25, 2001.

[21] M. Vukolić, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," in *Proceedings of the 2015 International Workshop on Open Problems in Network Security (iNetSec)*, Zurich, Switzerland, 2015, pp. 112-125.

[22] Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the 13th EuroSys Conference (EuroSys)*, Porto, Portugal, 2018, pp. 1-15.

[23] S. Meiklejohn et al., "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names," in *Proceedings of the 2013 Internet Measurement Conference (IMC)*, Barcelona, Spain, 2013, pp. 127-140.

[24] M. R. Anwar, D. Apriani, and I. R. Adianita, "Hash Algorithm in Verification of Certificate Data Integrity and Security," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 3, no. 2, pp. 181–188, Sep. 2021. [Online]. Available: https://doi.org/10.34306/att.v3i2.212

[25] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*. Springer, 2019.

[26] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol for Open Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, pp. 17–30, 2016.

[27] Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18)*, 2018.

[28] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy (SP)*, 2015, pp. 104–121.

[29] P. McCorry, S. F. Shahandasti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," in *Financial Cryptography and Data Security*, vol. 9603, Springer, 2016, pp. 357–375.

[30] Y. Liu, S. Dai, J. Zhang, and W. Wang, "From Bitcoin to Ethereum: A Security Analysis of Proof-of-Stake in Blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 285–299, 2022.

[31] H. Wang, Y. Ma, and K. Lu, "A Survey on Blockchain Security: Foundations, Security Model, Consensus, and Future Trends," *IEEE Access*, vol. 9, pp. 115–141, 2021.

[32] M. Xie, T. Wang, and W. Shi, "Blockchain for IoT Security and Privacy: The Case Study of Smart Home," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3275–3291, 2021.

[33] S. Iyer, K. Nandakumar, and J. K. Liu, "Blockchain-Based Privacy-Preserving Federated Learning in Healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 1, pp. 314–325, 2022.

[34] Fan, L. Han, and S. L. Shrestha, "Blockchain-Enabled Secure and Decentralized IoT System," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1503–1513, 2022.

[35] P. Ghimire and M. Levi, "A Comprehensive Survey on Blockchain for Healthcare: Challenges and Future Research Directions," *IEEE Access*, vol. 10, pp. 30011–30035, 2022.

[36] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on Blockchain Based Smart Contracts: Applications, Opportunities, and Challenges," *Journal of Network and Computer Applications*, vol. 177, p. 102857, 2021.

DOI: https://doi.org/10.54216/JCIM.160112
Received: November 29, 2024 Revised: January 31, 2025 Accepted: February 28, 2025