Optimization of SHA 256 with Finetune Pipeline and Parallel Processing with Split Techniques

Bharat S.Rawal[1], Satheesh Naidu Aleti, [2] Saikethan Reddy[3] Podduturi andTriven Kumar Chandu[4]`

[1]Dept. of Computer Science, Capitol Technology University

Washington DC,USA

[2,3&4]Dept. Computer and Information SciencesGannon University ,Erie, USA

[1]bsrawal@captechu.edu

[2,3&4] (aleti007, poddutur002 and chundu002) @gannon.edu

The technique of converting any given key or string ofcharacters into another value is known as hashing, frequently

Article Info
Page Number: 460 – 472
Publication Issue:
Vol. 71 No. 3s (2022)

Article History
Article Received: 22 April 2022
Revised: 10 May 2022
Accepted: 15 June 2022
Publication: 19 July 2022

Abstract

The designBy facilitating the effective transfer of sensitive information across all parties and the system's decentralization, blockchain-based technology could address various security issues in the IT sector, we examine the many strategies for optimizing SHA-256 block hash solving that are available, as even a little increase in block hash solving may have a significant impact on the bitcoin mining industry and decentralization, and authors explain some of the relevant efforts in this paper. The Secure Hash Algorithm-256 (SHA-256) is a cryptographic algorithm found in anything from the Internet of Things microdevices to supercomputers. This paper investigates many SHA-256 implementations and cryptographic hash function. Also, authors propose the finetune pipeline and parallel processing with Split-protocol.

Keywords: SHA-256, Blockchain, Hash, Bitcoin, Cryptography, FPGA, Split-protocol

I. INTRODUCTION

As Bitcoin miners are presently processing 200 Exa hashes per second. By identifying collisions in a Merkle root that goesinto the header, the ASIC miner’s utilization has already reduced the brute force effort by 15%. authors are looking into the methods of Optimizing SHA 256 block hash solving.

A. Cryptography:

The study and practice of tactics for safe communication in the face of hostile conduct. More broadly, cryptography is concerned with developing and evaluating methods that prohibit third parties or the public from accessing private messages, current cryptography emphasizes data secrecy, data integrity, authentication, and non-repudiation, Cryptography ensures safe communication in the face of adversaries, or malevolent third parties. Encryption transforms a plaintext input into an encrypted output using an algorithm and a key (i.e., ciphertext). An algorithm will always convert the same plaintext into the same ciphertext if the same key is used. The technique is considered safe if an attacker cannot extract any plaintext or key properties from the ciphertext. An attacker should not be able to derive anything about the key based on many plaintext/ciphertext combinations that used it.

B. Hashing:

represented by a shorter, fixed-length value or key that reflects

the original string and makes it simpler to locate or use it. The process by which a hash algorithm converts a message with a finite length into one with a predetermined length.

The primary goals of cryptography are to ensure the authenticity, integrity, and non-repudiation of information. Encryption is the process of masking the content of a message's message body from view. A cipher text is produced because of the encryption process (Rachmawati et al., 2018).

C. SHA 256:

The SHA is one of several cryptographic hash algorithms (Secure Hash Algorithm). A data set's signature is analogous to a cryptographic hash. It is usually preferable to hash and compare SHA256 values when comparing two bits of raw data (file source, text, or similar). The method will provide a different hash result even if just one symbol is modified. The SHA256 algorithm creates a 256-bit (32-byte) hash that is considerably unique. The hash function is excellent for data integrity checks, challenge hash authentication, anti-tamper, digital signatures, and blockchain applications since it is a one-way function.

D. Bitcoin:

A decentralized digital currency that keeps track of transactions on a distributed ledger known as a blockchain. Bitcoin miners use high-powered computers to solve difficult problems in order to confirm groupings of transactions known as blocks.

E. Block Chain:

Block chain was invented in 2008 by an anonymous person or group known only as 'Satoshi Nakamoto,' with the intention of serving as a public transaction record for the crypto currency Bit coin.

1) Basic structure:

A block is made up of two parts: the block header and the block content. A block is defined as a collection of legitimate transactions. The preceding block's hash digest, nonce, timestamp, and Merkle root are all included in the block header. The valid transactions are kept in the block's body.

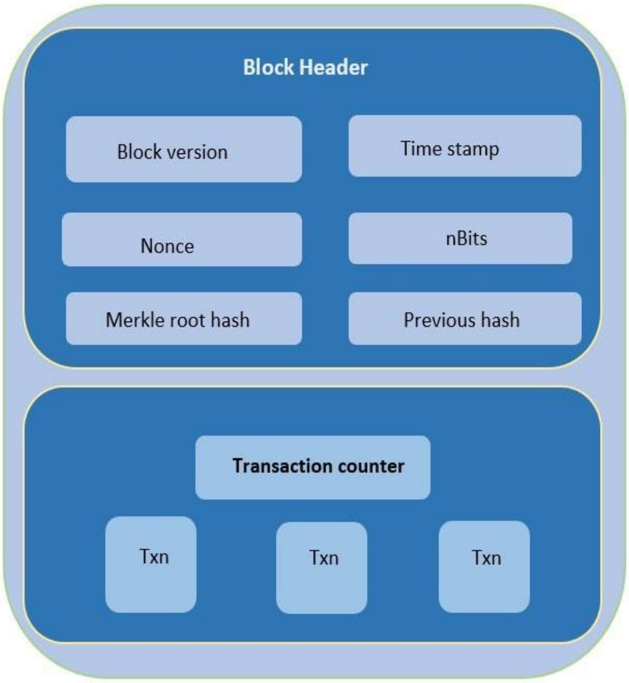

Figure 1: General structure of Block chain [10]

Author listed some of the methods available for optimization of SHA256 as, MATLAB FPGA[1], optimization techniques for Password-Based Key Derivation Function 2 (PBKDF2) Hash- based message authentication codes HMAC-SHA-2 Family and PBKDF2-HMAC-LSH Family[2], PBKDF2 optimization methods using multi-threads, optimization based on Proactive Reconfigurable Computing Architecture PRCA to increase the efficiency and security of the blockchain hash algorithm[3], an effective method for finding practical cases of (semi-free-start) collisions in SHA-256[4], AISC of SHA-256, a high- performance parallel computing hardware design [5], high-speed FPGA implementation of the SHA-256 algorithm as a pipelined design [6]. To boost the speed of the hardware implementation, applied techniques from block cipher cryptanalysis to hash functions, on block-cipher-based hash functions, higher-order differential cryptanalysis [7], reduction in the number of addition operations done in the compression function [8].

The rest of the paper is organized as follow, II Section presents related work, and Section III describe implementation of various SHA 256 optimization algorithm. Section IV describes Split-protocol, and Section V is Discussion. Section VI conclude the paper.

II. RELATED WORK

Methods of Optimization SHA256 to solve block hash:

As Zhang and Hu [1] proposed in paper the iteration bound theory, which is utilized to explore SHA256 optimization on FPGA, with a block diagram to go along with it. Then, based onthe peculiarities of the Bitcoin mining situation, the computing efficiency is increased. Finally, a simulation experiment is usedto evaluate the design's efficiency, and the results suggest that the hash function calculation's efficiency can be successfully enhanced at a lower cost.

H. Choi and S. C. Seo [2], offered many optimization strategies for the PBKDF2-HMAC-SHA-2 and PBKDF2-HMAC-LSH families in this study. The suggested PBKDF2-HMACSHA256 improves performance by around 121.26 percent when compared to KISA (Korea Internet & Security Agency) implementation, while the proposed PBKDF2-HMAC-LSH256 improves performance by about 325.91 percent when compared to KISA implementation.

To improve the efficiency and security of the blockchain hash algorithm, Fu et al., study offers a method of blockchain hash algorithm optimization based on PRCA [3].

According to Nicolas T, Grajek et al., the profitability of bitcoin mining is determined by a cryptographic constant, which is determined to be at most 1.89. Normally, only a small percentage of the population is interested in intricate cryptographic engineering minutiae. However, things are different in this case. Bitcoin miners can save millions of dollars each year because of this finding [4].

Naik and Nicolas's suggestion for improving the Bitcoin mining reward halving cycle was also offered, with the goal of bringing linearity to the reward paid to miners for mining new Bitcoins. To completely appreciate what was being presented, the thesis also attempted to organize background material as well as connected information [5].

As per Mendel and Tomislav 'due to multiple contradictory criteria in SHA-2, many discovered differential features are invalid, by identifying crucial bits during the search process andcoupling the search for differential features with the computation of conforming message pairs, this difficulty was overcome'. In comparison to more straightforward designs like MD5 and SHA-1, the search for valid differential features and conforming message pairings in SHA-2 is becoming increasingly challenging and unexpected [6].

Zhang, W. U. Ruizhen et al., presented AISC of SHA-256, a high-performance parallel computing hardware design. Based on device characteristics, the standard SHA-256 is postponed. As a result, the SHA-256 critical path is discovered and pipelined using DFFs. In compared to current SHA-256 designs, the proposed design may boost speed by 3 times at a cost of 2.90times the area cost and gain 50.7 percent in power [7].

Padhi and Ravindra's High-speed FPGA implementation of the SHA-256 algorithm is provided as a pipelined design. The improved hardware techniques are based on a fast-pipelined approach for pre-computing the SHA-256 algorithm's vigorous functions, which improves performance. In terms of maximum frequency, throughput, and efficiency, this design performsadmirably [8].

According to Biryukov, Mario et al., The On block-cipher- based hash functions, higher-order differential cryptanalysis is used and applied techniques from block cipher cryptanalysis to hash functions in their assault. When these strategies were applied to SHA-256, they resulted in a feasible attack for 47 (out of 64) phases of the compression function. So far, the most well-known assault with practical complexity has been for 33 stages[9].

As per Annu, and Bhakthavatchalu the Xilinx ISE 14.2 tool is used to verify the design, which is then implemented on an Artix-7 Low Voltage FPGA. Even a slight reduction in the number of addition operations done in the compression function can greatly improve the process. The longest data path or crucial path is formed by the additions using 7 operands in the computation of working variable A [10].

According to Zhu and Wang the plaintext is mixed in with the chaotic sequence, which results in a random number sequence that is then used to further muddle the image. Site feedback on the plaintext cypher can be determined by an index that changes over time (Zhu et al., 2018). Key space examination, critical sensitivity examination, differential analyses, histograms, data entropy, and association coefficients depict about the picture encryption algorithm is secure and reliable., as well as having high application potential [11].

According to Tran et al., by reducing the amount of time data has to travel between the accelerator and external memory, these memories are effective. A few interesting concepts, such as pipelined ALUs, multimem PEs, and SBi- SBo. Aside from that, data buses aren't only reserved for the SHA-256 accelerator [12].

According to Hakim and Vaze ‘the blockchain for secure medical records storage and medical service framework using SHA 256-verifiable key’, The decentralized interchange of health data in the blockchain system safeguards personal health information, allowing for a broader range of uses in the healthcare industry [13].

According to Gadamsetty et al., The model prefers the idea of hashing because hashing is one-to-one and non-reversible. Since all SAR data originates from a satellite and is initially consolidated, it might become a target and be easily attacked if no security measures are used to safeguard it [14].

A wide range of applications relying on the SHA-2 cryptographic algorithms were explored by Pham et al., (2022). These applications ranged from maintaining information security and integrity in network protection to facilitating the proliferation of blockchain networks [15]. As per Rachmawati et al., The SHA256 algorithm works as input a message of any length that is less than 264 bits in length and creates as output a 256-bit message digest of the output message with the same length as the input message [16].

Wang et al., [17] implemented the compression function has a direct impact on the Hash Algorithm's performance because it is the most critical component for performing compression mapping. It's easy for the attacker to spot a collision if the computational complexity is lower than the cost of an exhaustive attack. more stability in confusion and diffusion, as well as increased resistance to collision.

Pham et al., 2020 [18])., summarizes, a CME double SHA256 hardware circuit is proposed by the author in order to minimize power usage and additional mining speed. There are three

SHA-256 circuits, one of which is a resource-sharing circuit while the other two are completely unrolled data path circuits. High processing speeds can be attained at low hardware costs by using this method. authors provide many compact message expander technologies and hardware architectures to minimize power usage and hardware expenses.

Nakamura et al., As a result, authors discovered that by inserting an average of only 18 errors into the SHA-256 compression process, whereby find the entirety of an unrecognized outcome to the compression. They also carried study for an AFA for the SHACAL-2 block cipher as well as anAFA for the SHA-256 compression role, which resulted in the forgery of the chop MDMAC function being virtually universally successful [19]. Sundarakumar and Mahadevan [20], every attempt is made by the data owners to encrypt their data and store it in the cloud. Downloading the encrypted files and decrypting them using your personal key is the first step in gaining access to them Kammoun et al., [21], propose designing an innovative contract platform application to implement the FPGA-based implementation of the main challenge that authorsare dealing with here is how to achieve such an FPGA based onblockchain. This leads us to the topic of blockchains and cryptocurrency. According to Devika, K. N., and Ramesh Bhakthavatchalu [22], SHA-256 accelerator can search for collisions in many large groups of devices, which gives an advantage over brute force search. It has a large memory capacity, allowing it to store up to four times as much data as the enormous RAM in your system.

According to Pham et al., [23] performance of the FPGA cluster concerning the speed of processing time, throughput, andlatency of each node can be measured over time. authors start with an experimental setup with the first block of 100 hashes asthe baseline. Once the nodes are fully integrated, authors applya new hashing algorithm based on the same key value.

According to Bensalem et al., [24], algorithm has been tested on hardware platforms at various hardware and programming levels. It has performed well on legacy architectures and the more recent processor platforms, including x86, PowerPC, and PowerVR. The design approach to hardware execution is simple. The design relies on the following features: The SHA-

256 algorithm can be optimized in a highly parallel, data- parallel programming environment.

According to Wong et al., [25] IoT devices and data are stored in a data center to perform complex computations in real- time. Therefore, the computing environment is divided into different types of nodes known as nodes, and another type is considered a storage node. For example, nodes store data on the client-side, while nodes stored on the server-side are considered storage nodes. Similarly, a data center has four main stages: Storage node is typically a server, and it can store and transmit information to the client.

As per Suresh et al., [26] SHA-1 is a set of 64 rounds of a function with a complexity of 16 bits is provided. Each game consists of block encryption and block decryption using a fixed 64-bit seed value chosen during the selection algorithm, for example, The SHA-256 hash function provides a secure implementation of DES in asymmetric key-dependent setting; the corresponding key lengths in DES have been measured by using a technique known as round-Diffie- Hellman as, the result of SHA-256 block encryption is the product of the previous SHA-256 output of the algorithm.

Martino et al., [27], describes a decentralized, secure, smart contract based on a tamperproof consensus scheme called the smart contract. This contract was deployed on the Ethereum blockchain to improve scalability in large-scale IoT using Ethereum consensus algorithms. The smart contract was implemented in a C++ and JavaScript-enabled client written in Solidity using the EVM and Miasm blockchain.

According to Fotohi and Fereidoon [28] study presents a novel and efficient system that utilizes the SHA-256 hash function to speed up the communication between a sender and a receiver. It provides a single interface to transmit the hash function in both directions. It is based on a modified and testbed system that uses a custom FPGA architecture—the Secure Hash Algorithm for Encryption Using MD5 in a Power-Efficient Microprocessor. A secure hash function is acryptographic algorithm used to create a message’s digital signature.

According to Kieu-Do-Nguyen et al., [29], Hash- Sha-256is used for stream ciphers insecure network, a public network of computers, where it has the security of a one-way communication function in a public network such as the Internet. Hash- Sha-256 can be applied for stream ciphers in web servers to make their user compliant with the requirements and check if the data is correct. Hash- Sha-256 is used on the Internet by creating a hash of the message and content to obtainthe key.

Khasanah F. N [30], presented cryptographic schemes for securing various digital assets and the various cryptographic techniques that authors use for that purpose. The security requirements for any cryptographic system applied to the specified information need to be met, and the critical management and related issues must also be addressed. The cryptographic components of an SSL encryption scheme should be based on a TLS encryption scheme (STS). In TLS, these consist of the cryptographic keys, the secret key, and the random number specified in the certificate. Thus, the key and the private key can be used to derive the corresponding randomnumber, which authors will call a private key.

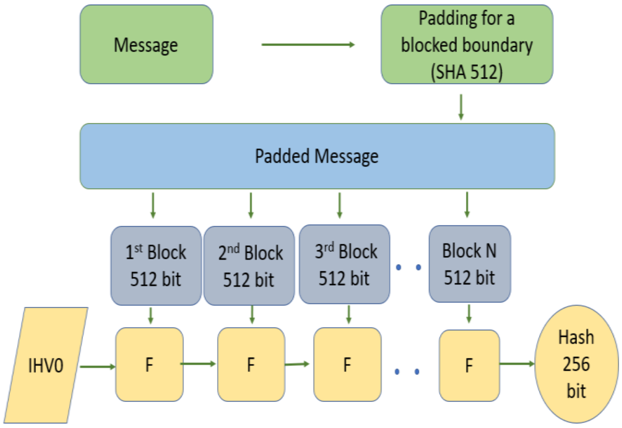

Figure 3: Flow Chart SHA256 [16]

III. IMPLEMENTATION OF SHA 256 USING MATLAB AND ON FPGA BY THE APPLICATION OF BLOCK CHAIN CONCEPTS’

A. FPGA:

Reprogrammable integrated circuits with an array of programmable logic blocks are known as field-programmable gate arrays (FPGAs). The flexibility, hardware-timed speed and dependability, and concurrency of FPGA chips are drivingtheir popularity.

B. Merkle Root Tree:

A Merkle tree, often known as the hash tree, is used to efficiently and securely encode blockchain data. It enables the peer-to-peer blockchain network to quickly verify blockchain data as well as transport massive volumes of data from one computer node to the next.

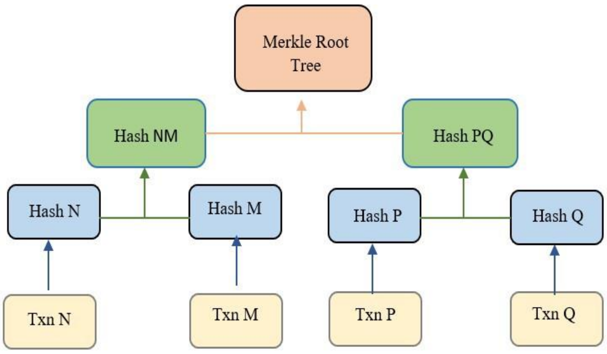

Figure 2: Merkle root tree flow [10]

Three components make up the design:

- The top module is responsible for message pre-processing, constant initialization, and hash value concatenation.
- The second module is responsible for compression andhash value change.
- The third module is responsible for the message scheduleextension.

A smart contract is a protocol where users execute smart contracts. Arrangements are performed on the blockchain with an interface like the traditional web-based applications. This interface allows clients to execute arbitrary code. This process starts with identifying the stakeholders of a blockchain, such as the participants in a community of cryptos, their roles and responsibilities, information security, the main rules of the blockchain, including the protocol, the laws of the execution protocol, the regulations of the voting protocol, and so forth. Atthis stage, a thorough research and development effort is initiated. The development of the relevant software and hardware aims to produce suitable software to be used as the foundation on which to build a blockchain. SHA-256 and a variety of other hash functions and algorithms [7]. Crypto Hash is a public key-based cryptographic hashing system for storing and distributing cryptographic data. It uses a 256-bit block of amessage and a unique hash value for each block. This means the message can be viewed only by the recipient and anyone else who possesses a 256-bit block of the original message. However, using cryptography as a digital signature technology, a message can be authenticated and verified only once sent to the other party.

Ideally, SHA-1 uses 160 bits of key size. On the other hand, SHA-256 utilizes a 256-bit encryption key for the data under transmission. SHA-2 may not have specific bits for the encryption key since it is a US government-developed family for protecting online data. Probably, the SHA-3 would provide more solid security than any other cryptographic encryption. It has a higher cryptographic strength than SHA-256, with a similar key length of 256. Ultimately, SHA-256 is among the most widely used hashing algorithms due to its power which has not been cracked yet; the algorithm enables quick calculation of the hashes, for instance, SHA-512 Most organizations and individuals implement SHA-256 due to its recommended hashing functions for the company's software. The current computing powers of SHA-256, such as SHA-384 and SHA- 512, are significantly secure for the company's information [7].

IV. SPLIT-ARCHITECURE

Split protocol:

Protocol splitting allows protocols to be separated at the server level without the intervention of the client. For load balancing and faster data transfer, the split-protocol theory was devised. Web services on geographically distributed web servers in the cloud are used in the split-protocol computing paradigm. A cloud made up of enormous split-servers that handle computing and storage tasks that would otherwise use a lot of CPU if dealt with individual servers. Within a normal session, a new split protocol client/server design isolates connections and data transfers fully.

First stage computes x0, and pass it on 2nd stage computes receives x0, computes x1, and passes both x0 and x1 to the next stage. So forth and so on,

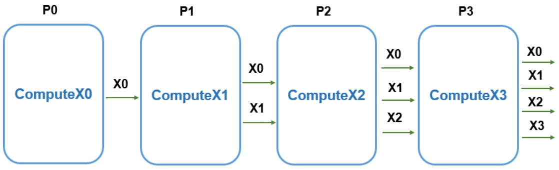

Figure 4. Pipelining back substitution

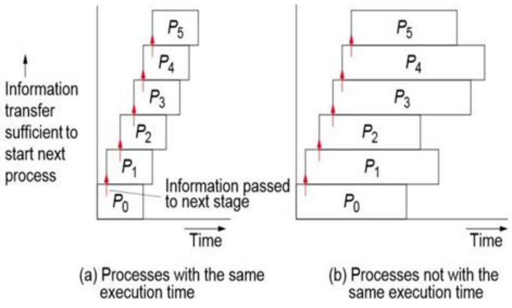
Information transfer sufficient to start next process
Information passed to next stage
P_0
P_1
P_2
P_3
P_4
P_5
Time
Time
(a) Processes with the same execution time
(b) Processes not with the same execution time

Figure 5. Pipelining even and uneven process execution.

As per Rawal et al., this architecture authors assume the key elements and the key features are offered by one meta server. In other words, the Split [34], system contain four identical servers and each server performs only specific functions of the metaverse. Since all are identical it is easy demonstrate role changeover [35]. In case of any server is heavily stressed the migration of task or server can easily implemented with Split- protocol [35]. The role changeover, and migration capability make system more resilient and fault tolerant. A. State transition for role changeover with Split-protocol authors relate the number of roles change over at any state to the number in the previous server states by the difference equation. Since software rejuvenation is a four-step process: functional states, stop, clean internal state and resume operation [32,33].

V. CONCLUSION

In Conclusion as we compared different methods of optimization for SHA 256 for solving block hash, each of the methods described above have their own edge and advantages over the other and have their own disadvantages specific to them per the researcher's view. Even a slight reduction in the number of addition operations done in the compression function can greatly improve the process. This is another area where improvements may be made. This approach might be beneficial in the development of efficient digital signature architectures with excellent hardware security. And AISC of SHA-256, a high-performance parallel computing hardware design was presented. Based on device characteristics, the standard SHA256 is postponed. As a result, the SHA-256 critical path is discovered and pipelined using DFFs. In compared to current SHA-256 designs, the proposed design may boost speed by 3 times at a cost of 2.90 times the area cost and gain 50.7 percent in power, and FPGA architecture, secure mode, in which the block cipher protects FPGA execution. This new security level enables us to securely deliver FPGA-based security, which is the degree of protection that can be provided by employing an FPGA as a compute device, such as a CPU or GPU processing core. The key problem is figuring out how to make a blockchain- based FPGA. This leads to the subject of cryptocurrencies and blockchains. If the process is independent than Split- architecture provides significant percent of throughput, and it is scalable for system of large servers.

VI. REFERENCES

1. Zhang, Xiaohan, and Honggang Hu. "Optimization of hash function implementation for bitcoin mining." In 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019), pp. 448-452. Atlantis Press, 2019.
2. H. Choi and S. C. Seo, "Optimization of PBKDF2 Using HMAC-SHA2 and HMAC-LSH Families in CPU Environment," in IEEE Access, vol. 9, pp. 40165-40177, 2021, doi: 10.1109/ACCESS.2021.3065082.
3. Fu, Jinhua, Sihai Qiao, Yongzhong Huang, Xueming Si, Bin Li, and Chao Yuan. "A study on the optimization of blockchain hashing algorithm based on PRCA." Security and Communication Networks 2020 (2020).
4. Courtois, Nicolas T., Marek Grajek, and Rahul Naik. "Optimizing sha256 in bitcoin mining." In International Conference on Cryptography and Security Systems, pp. 131-

144. Springer, Berlin, Heidelberg, 2014.

5. Naik, Rahul P., and Nicolas T. Courtois. "Optimising the sha256 hashing algorithm for faster and more efficient bitcoin mining." MSc Information Security Department of Computer Science UCL (2013): 1-65.
6. Mendel, Florian, Tomislav Nad, and Martin Schläffer. "Finding SHA-2 characteristics: searching through a minefield of contradictions." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 288-307. Springer, Berlin, Heidelberg, 2011.
7. Zhang, Xiaoyong, W. U. Ruizhen, Mingming Wang, and Lin Wang. "A high-performance parallel computation hardware architecture in ASIC of SHA-256 hash." In 2019 21st International Conference on Advanced Communication Technology (ICACT), pp. 52-55. IEEE, 2019.
8. Padhi, Meelu, and Ravindra Chaudhari. "An optimized pipelined architecture of SHA-256 hash function." In 2017 7th International Symposium on Embedded Computing and System Design (ISED), pp. 1-4. IEEE, 2017.
9. Biryukov, Alex, Mario Lamberger, Florian Mendel, and Ivica Nikolić. "Second-order differential collisions for reduced SHA-256." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 270-287. Springer, Berlin, Heidelberg, 2011.
10. Thomas, Annu, and Ramesh Bhakthavatchalu. "Implementation of SHA 256 using MATLAB and on FPGA by the Application of Block Chain Concepts." In 2021 International Conference on Communication, Control, and Information Sciences (ICCISc), vol. 1, pp. 1-5. IEEE, 2021.
11. Zhu, S., Zhu, C., & Wang, W. (2018). A new image encryption algorithm based on chaos and secure hash SHA-256. Entropy, 20(9), 716. Retrieved from: https://www.mdpi.com/1099-4300/20/9/716/htm
12. Tran, T. H., Pham, H. L., & Nakashima, Y. (2021). A high-performance multimem SHA-256 accelerator for society 5.0. IEEE Access, 9, 39182- 39192. Retrieved from:
13. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumbe r=9367201
14. Hakim, P. D., & Vaze, V. M (2021). Blockchain for Secure Medical Records Storage and Medical Service Framework using SHA 256– Verifiable Key. Retrieved from: https://web.archive.org/web/20211030214423id_/https://inass.org/wpcontent/uploads/2021/10/2021123101.pdf
15. Gadamsetty, S., Ch, R., Ch, A., Iwendi, C., & Gadekallu, T. R. (2022). Hash-Based Deep Learning Approach for Remote Sensing Satellite Imagery Detection. Water, 14(5), 707. Retrieved from: https://www.mdpi.com/2073-4441/14/5/707/htm
16. Pham, H. L., Tran, T. H., Le, V. T. D., & Nakashima, Y. (2022). A HighEfficiency FPGA-Based Multimode SHA-2 Accelerator. IEEE Access,
17. 10, 11830-11845. Retrieved from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9691379
18. Rachmawati, D., Tarigan, J. T., & Ginting, A. B. C. (2018, March). A comparative study of Message Digest 5 (MD5) and SHA256 algorithm. In Journal of Physics:

Conference Series (Vol. 978, No. 1, p. 012116). IOP Publishing. Retrieved from: https://iopscience.iop.org/article/10.1088/1742-6596/978/1/012116/pdf

19. Wang, J., Liu, G., Chen, Y., & Wang, S. (2021). Construction and analysis of SHA-256 compression function based on chaos S-box. IEEE Access,
20. 9, 61768-61777. Retrieved from: https://ieeexplore.ieee.org/abstract/document/9398665
21. Pham, H. L., Tran, T. H., Phan, T. D., Le, V. T. D., Lam, D. K., & Nakashima, Y. (2020). Double SHA-256 hardware architecture with compact message expander for bitcoin mining. IEEE Access, 8, 139634- 139646. Retrieved from: https://ieeexplore.ieee.org/abstract/document/9151160
22. Nakamura, K., Hori, K., & Hirose, S. (2021). Algebraic Fault Analysis of SHA-256 Compression Function and Its Application. Information, 12(10), 433. Retrieved from: https://www.mdpi.com/2078-2489/12/10/433/htm
23. Sundarakumar, M. R., and G. Mahadevan. "Authorization for secured cloud storage through SHA-256." (2019)
24. Kammoun, M., Elechi, M., Abid, M., & BenSaleh, M. S. (2020, June). FPGA-based implementation of the SHA-256 hash algorithm. In 2020 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS) (pp. 1-6). IEEE.
25. Devika, K. N., and Ramesh Bhakthavatchalu. "Parameterizable FPGA implementation of SHA-256 using blockchain concept." 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2019.
26. Pham, H. L., Tran, T. H., & Nakashima, Y. (2021, April). Highperformance multicore SHA-256 accelerator using fully parallel computation and local memory. In 2021 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS) (pp. 1-3). IEEE.Bensalem, Hachem, Yves Blaquière, and Yvon Savaria. "Acceleration of the secure hash algorithm-256 (SHA-256) on an FPGA-CPU cluster using OpenCL." 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2021.
27. Wong, Ming, Vikramkumar Pudi, and Anupam Chattopadhyay. "Lightweight and high-performance SHA-256 using architectural folding and 4-2 adder compressor." 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC). IEEE, 2018.
28. Suresh, V., Satpathy, S., Mathew, S., Anders, M., Kaul, H., Agarwal, A.,
29. ... & Krishnamurthy, R. (2018, September). A 230mv-950mv 2.8 tbps/w unified sha256/sm3 secure hashing hardware accelerator in 14nm tri-gate CMOS. ESSCIRC 2018-IEEE 44th European Solid-State Circuits Conference (ESSCIRC) (pp. 98-101). IEEE.
30. Martino, Raffaele, and Alessandro Cilardo. "A configurable implementation of the SHA-256 hash function." International Conference on P2P, Parallel, Grid, Cloud, and Internet Computing. Springer, Cham, 2019.
31. Fotohi, Reza, and Fereidoon Shams Aliee. "Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT." Computer Networks 197 (2021): 108331.
32. Kieu-Do-Nguyen, B., Hoang, T. T., Pham, C. K., & Pham-Quoc, C. (2021, October). A

Power-efficient Implementation of SHA-256 Hash Function for Embedded Applications. The 2021 International Conference on Advanced Technologies for Communications (ATC) (pp. 39-44). IEEE.

33. Khasanah, F. N. (2022). Application of Hash Sha-256 Algorithm in Website-Based Sales Software Engineering. Journal of Applied Data Sciences, 3(1), 24-32.
34. Suman, Rajiv Ranjan, Bhaskar Mondal, and Tarni Mandal. "A secure encryption scheme using a Composite Logistic Sine Map (CLSM) and SHA-256." Multimedia Tools and Applications (2022): 1-22.
35. Trivedi, Kishor S., and Andrea Bobbio. Reliability and availability engineering: modeling, analysis, and applications. Cambridge University Press, 2017.
36. Rawal, Bharat S., Qiang Duan, and Pandi Vijayakumar. "Dissection of the experimental outcome of split-protocol." International Journal of Advanced Intelligence Paradigms 10, no. 1-2 (2018): 23-44.
37. Rawal, Bharat S., Ramesh K. Karne, and Alexander L. Wijesinha. "Splitting HTTP requests on two servers." In 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), pp. 1-8. IEEE, 2011.
38. Rawal, Bharat S., Ramesh K. Karne, Alexander L. Wijesinha, Harold Ramcharan, and Songjie Liang. "A split protocol technique for web server migration." In 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), pp. 1-6. IEEE, 2012.