Evaluating Hashing Algorithms in the Age of ASIC Resistance

Oleksandr Kuznetsov [1,2], Yelyzaveta Kuznetsova [2], Oleksii Smirnov [3], Oleksii Kostenko [4] and Volodymyr Zvieriev [5]

[1] *Department of Political Sciences, Communication and International Relations, University of Macerata, Via Crescimbeni, 30/32, 62100 Macerata, Italy*
[2] *Department of Information and Communication Systems Security, School of Computer Sciences, V. N. Karazin Kharkiv National University, 4 Svobody Sq., 61022 Kharkiv, Ukraine*
[3] *Department of cyber security and software, Central Ukrainian National Technical University, 8, University Ave, 25006 Kropyvnytskyi, Ukraine*
[4] *Scientific Laboratory of the Theory of Digital Transformation and Law of the Scientific Center for Digital Transformation and Law of the State Scientific Institution "Institute of Information Security and Law of the National Academy of Legal Sciences of Ukraine", P.Orlyk str., 3, 01024, Kyiv, Ukraine*
[5] *Department of Software Engineering and Cybersecurity, State University of Trade and Economics, Kyoto, 19, 02156 Kyiv, Ukraine*

Abstract

In the intricate matrix of cryptographic hashing, two contrasting paradigms vie for precedence: computational swiftness and resilience against specialized hardware, or ASICs. This study undertakes a meticulous exploration into these dueling priorities, juxtaposing conventional stalwarts like SHA-2 and KECCAK against the newer, ASIC-resistant X series. Leveraging detailed performance metrics and visual analytics, we discern the manifest advantage of SHA-2 and KECCAK in terms of computational alacrity. However, as the paper delves deeper, a compelling narrative unfolds. The deliberate design intricacies of the X series, despite their computational latency, emerge as robust bulwarks against potential centralization threats posed by ASIC miners. These algorithms embody a strategic deceleration, ensuring that the cryptocurrency terrain remains hospitable to a broad diaspora of miners. Through this multifaceted lens, we argue that in the cryptographic domain, raw speed is but one facet of a larger, more nuanced picture. The underlying ethos of decentralization and democratic access must guide our trajectory, even if it comes at the expense of pure efficiency. In essence, this paper endeavors to navigate the delicate balance between rapid computation and the overarching principles of an inclusive digital realm.

Keywords

Cryptographic hashing, ASIC resistance, SHA-2, KECCAK, X series, Computational efficiency, Hashing algorithms

1. Introduction

In the dynamically evolving world of blockchain technology, the reliability, security, and performance of cryptographic hash functions play a paramount role in ensuring the integrity and stability of blockchain systems [1]. The X-series hash algorithms, starting with X11 and extending to X14, have emerged as innovative solutions in response to concerns over the potential vulnerabilities of simpler hash functions [2,3].

Proceedings ITTAP'2023: 3rd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2023, Ternopil, Ukraine, Opole, Poland
EMAIL: kuznetsov@karazin.ua (A. 1); elizabet8smidt12@gmail.com (A. 2); dr.SmirnovOA@gmail.com (A. 3); antizuk@gmail.com (A. 4); zvieriev_vp@knute.edu.ua (A. 5)
ORCID: 0000-0003-2331-6326 (A. 1); 0000-0002-0573-0913 (A. 2); 0000-0001-9543-874X (A. 3); 0000-0002-2131-0281 (A. 4); 0000-0002-0907-0705 (A. 5)

The primary ethos behind the creation of the X-series was to introduce a multi-algorithmic approach to hashing. This not only elevates the security barriers but also attempts to democratize the mining process, making it more resistant to ASIC (Application-Specific Integrated Circuits) domination. ASIC resistance is fundamental in preventing the monopolization of mining by powerful entities, ensuring that the power within the blockchain remains decentralized [4]. Furthermore, a multi-algorithmic approach implies that even if one of the algorithms gets compromised, the entirety of the hash function remains robust, drawing its strength from the collective security of the other algorithms [5–7].

For X11, the journey began with an amalgamation of 11 different cryptographic algorithms [3]: BLAKE [8–11], BMW [12], GROESTL [13,14], J-H [9,15,16], KECCAK [6,14,15], SKEIN [9,17], LUFFA [9,18], CUBEHASH [9], SHAVITE [9,19], SIMD [9,11], and ECHO [9,20]. But as the digital currency ecosystem progressed, the need for additional layers of security became evident. X12 incorporated HAMSI [9], X13 introduced FUGUE [9,21], and X14 brought in SHABAL [9,22] to the ensemble. Each of these added algorithms has its own strengths and rationale, and their integration further enhances the resilience of the hash function.

In the context of blockchain systems, the significance of these algorithms cannot be understated. A blockchain, at its core, is a chain of blocks, and each block contains data that's represented by a hash [23–25]. If the hashing mechanism is compromised, the integrity of the entire blockchain can be questioned, jeopardizing trust in the system. Thus, the robustness of the hash function directly correlates with the trustworthiness of the blockchain.

This paper aims to delve deep into the performance metrics of X11, X12, X13, and X14 hash algorithms. Through rigorous analysis, we intend to shed light on the efficiency, security, and overall implications these algorithms have on modern blockchain systems. Given the profound impact and reliance of blockchain technologies on cryptographic hash functions, understanding the nuances and intricacies of the X-series algorithms becomes not just an academic pursuit but a necessity for the future of decentralized systems.

2. Literature Review

The realm of cryptographic hashing has witnessed a plethora of literature, each bringing forth distinct facets of this multifaceted domain. At its inception, Merkle's revolutionary paper introduced cryptographic hashing, delineating its foundational principles and applications [26]. This seminal work laid the groundwork for numerous endeavors in the cryptographic landscape.

The evolution of hashing algorithms has been systematically chronicled by Rogaway and Shrimpton [27]. Their comprehensive survey provides a diachronic perspective, encompassing early designs to contemporary cryptographic mechanisms. Particularly, they elucidate the importance of attributes like collision resistance, pre-image resistance, and second pre-image resistance in cryptographic hashing.

A more specific focus on SHA-2, its design, strengths, and weaknesses, is expertly detailed in the works of Turner [28] and Dang et al [29]. Their meticulous exploration into the National Institute of Standards and Technology (NIST) endorsed algorithms affirms their computational efficacy. However, with the advent of ASICs and their looming threat of centralization, researchers like Harvey-Buschel and Kisagun began highlighting the potential pitfalls of relying solely on computational efficiency [30].

Evidently, the rise of ASICs necessitated an evolution in the cryptographic hashing paradigm [31]. This brought about the exploration of ASIC-resistant algorithms, a topic elaborately discussed in the paper by Bex et al [32]. Their discourse emphasizes the need for balanced trade-offs to ensure decentralization in the cryptocurrency ecosystem.

The Keccak team's treatise on the SHA-3 algorithm offers insights into its design philosophy and underlying intricacies [33]. While it stands as a testament to efficiency, the cryptocurrency community's relentless pursuit of ASIC resistance led to the advent of the X series [2,3]. These, while not as prolifically documented as their predecessors, have been critiqued and discussed in forums and white papers, emphasizing their role in the broader cryptographic panorama.

In summation, the literature presents a compelling narrative of hashing's evolution, from its nascent stages to the contemporary challenges and strategic pivots necessitated by the ever-evolving digital landscape.

Recent our contributions have added depth to this discourse. In our publication, we provided a rigorous performance analysis of cryptographic hash functions specifically geared towards blockchain applications [5]. This study offers both empirical and analytical insights, highlighting the balance between security and efficiency. Our prior work focused on the performance evaluation of hash algorithms on GPUs for blockchain applicability [6]. With the rise of GPU-intensive tasks and their potential for parallelization, this research provides valuable insights into how different hashing algorithms perform in such environments. The importance of this investigation becomes particularly pertinent given the escalating arms race between ASICs and GPUs in the mining sector. A complementary our publication approached the domain from a statistical perspective, assessing the robustness and reliability of blockchain hash algorithms through rigorous testing [7]. Our findings underscored the importance of continuously vetting and validating the hash functions to ascertain their resilience against potential vulnerabilities.

Collectively, these contributions have significantly expanded our understanding of the nuances involved in cryptographic hashing, particularly in the context of blockchain. This current article stands as a logical extension of this series, delving further into the intricacies of cryptographic hashing in light of ASIC resistance and the emerging challenges therein.

3. Methods

3.1. System Configuration and Environment

The benchmark testing for the X11, X12, X13, and X14 hashing algorithms was executed on an Intel Core i9-7980 with a clock speed of 2.60 GHz. The system was complemented by a Windows 10 operating system, ensuring a stable and standardized environment for all tests. This configuration was chosen to provide a balance between modern computing capabilities and reproducibility for future comparative studies.

3.2. Data Sets

An array of data lengths was employed to ascertain the speed characteristics and efficiency of the hashing algorithms over diverse data sizes. The specific byte-lengths chosen were: 1, 2, 4, 8, 16, 32, and 64 bytes. This expansive range aimed to offer insights into the algorithms' behavior across both minimal and expansive data sets, simulating real-world scenarios of varying transaction sizes within blockchain systems.

3.3. Performance Metrics

For each data length, three critical speed characteristics were measured to provide a comprehensive understanding of the hashing algorithms' performance:

- Cycles per Byte (C/B): This metric denotes the number of CPU cycles required to process each byte of data, providing insight into the computational intensity of the hash function for different data sizes.
- Throughput (MB/sec): Illustrating the data processing speed, this parameter conveys the volume (in Megabytes) processed by the hash function every second. A higher throughput signifies more efficient data handling.
- Hash Rate (KHash/sec): This measure reflects the number of hash computations executed every second, expressed in thousands (KiloHashes). It offers a direct lens into the hashing speed, a critical aspect for blockchain applications, especially in contexts like cryptocurrency mining.

All tests were performed multiple times to account for any potential variability and ensure consistent results. The average values were subsequently taken for each data length and metric, bolstering the accuracy and reliability of the findings.

4. The X11 Hashing Algorithms

The X11 hashing scheme, a formidable and intricate construct, amalgamates eleven distinct cryptographic hash functions to create a multi-layered, robust, and adaptable framework. Each individual algorithm within the X11 ensemble has been meticulously chosen due to its proven cryptographic strengths and unique attributes. This section delves deep into each of these algorithms, shedding light on their mechanisms and relevance within the X11 hashing paradigm.

4.1. BLAKE

- *Overview*: BLAKE is an algorithm that was one of the finalists in the NIST SHA-3 competition [34]. It's known for its high speed in software and resistance to differential cryptanalysis.
- *Features*: Incorporates the HAIFA structure (a variant of the Merkle–Damgård construction) and leverages components of the ChaCha stream cipher. Its design principles center around simplicity and security.
- *Relevance to X11*: Provides a rapid start to the chaining of algorithms within X11 and ensures initial resistance to various cryptanalytic vulnerabilities.

4.2. BMW (Blue Midnight Wish)

- *Overview*: Another finalist of the NIST SHA-3 competition [34], BMW stands out due to its intricate design and high security margins.
- *Features*: Utilizes a complex structure involving bitwise operations, lookup tables, and modular arithmetic. Designed for optimal performance on 64-bit platforms.
- *Relevance to X11*: Offers depth to the chained structure with its unique design, ensuring diversity in cryptographic techniques used.

4.3. GROESTL

- *Overview*: A NIST SHA-3 finalist [34], GROESTL is aimed at achieving a high level of security through its unique construction, which minimizes the risk of certain types of cryptanalytic attacks.
- *Features*: Uses a two-fold permutation process and the AES block cipher construct. Provides high assurance against differential and linear cryptanalysis.
- *Relevance to X11*: Introduces AES-based operations into the mix, giving the chain enhanced resilience against potential vulnerabilities.

4.4. J-H

- *Overview*: A participant of the NIST SHA-3 competition [34], J-H emphasizes cryptographic robustness and adaptability.
- *Features*: Built upon the sponge construction, it uses an expansive S-box and an 8x8 matrix for its cryptographic processes.
- *Relevance to X11*: The sponge construction of J-H ensures varied cryptographic processing, further diversifying the X11 structure.

4.5. KECCAK

- *Overview*: The crowned winner of the NIST SHA-3 competition [34], KECCAK is renowned for its stellar security and efficiency.
- *Features*: Employs a permutation-based sponge construction. Renowned for its high-speed performance and minimalistic design.

- *Relevance to X11*: Being a core component, KECCAK lends its well-established security credentials to the overall robustness of the X11 scheme.

4.6. SKEIN

- *Overview*: A NIST SHA-3 finalist [34], SKEIN is built around the Threefish block cipher and emphasizes both speed and security.
- *Features*: Incorporates the Unique Block Iteration (UBI) chaining mode and offers flexibility in output size.
- *Relevance to X11*: Its flexible nature and the UBI chaining mode provide a versatile cryptographic angle to the X11 chain.

4.7. LUFFA

- *Overview*: A participant in the NIST SHA-3 contest [34], LUFFA is a wide-pipe construction hash function with a layered design.
- *Features*: Contains three parallel streams of operations and combines results in a unique weaving technique.
- *Relevance to X11*: The wide-pipe structure of LUFFA adds an extra layer of cryptographic complexity, further solidifying the security parameters of X11.

4.8. CUBEHASH

- *Overview*: Developed as an entrant to the NIST SHA-3 competition, CUBEHASH offers a balance of security and efficiency.
- *Features*: Uses an iterative permutation-based approach and can be tuned for performance on both hardware and software platforms.
- *Relevance to X11*: Its adaptability and iterative nature introduce additional cryptographic variety into the X11 algorithm.

4.9. SHAVITE

- *Overview*: Another contestant of the NIST SHA-3 race [34], SHAVITE focuses on providing high-speed hashing on a variety of platforms.
- *Features*: Incorporates a fast block cipher in its design and optimizes performance on 32-bit architectures.
- *Relevance to X11*: Ensures that X11 remains efficient across diverse hardware architectures.

4.10. SIMD

- *Overview*: Developed for the NIST SHA-3 competition [34], SIMD stands for "Single Instruction, Multiple Data" reflecting its parallel processing design.
- *Features*: Operates on a wide data path and employs a complex sequence of operations for maximum security.
- *Relevance to X11*: Its parallel processing approach aids in maximizing the throughput of the hashing scheme.

4.11. ECHO

- *Overview*: A NIST SHA-3 competitor [34], ECHO is designed around the AES block cipher, targeting both efficiency and security.
- *Features*: Uses a unique double-pipeline structure and optimizes AES operations for hashing.
- *Relevance to X11*: Contributes AES-based cryptographic strength and introduces an innovative pipeline design to the X11 algorithm.

In culmination, the X11 hashing scheme stands as a testament to the strengths of amalgamating diverse cryptographic techniques. By harnessing the unique attributes of each of its constituent algorithms, X11 ensures a fortified and versatile hashing paradigm, pivotal for blockchain's rigorous demands.

5. Subsequent Evolutions of the X Hashing Algorithms: X12, X13, and X14

The X series of hashing algorithms has been a hallmark of cryptographic progression, offering multi-algorithmic chains that provide a robust defense against potential vulnerabilities. The inception of the X11 algorithm set a precedent in combining 11 individual cryptographic hashes. Its successors, X12, X13, and X14, further build upon this foundation by incorporating additional hashing mechanisms. This section highlights the added algorithms in these subsequent versions and provides a detailed examination of their cryptographic relevance.

5.1. X12 (HAMSI)

Building upon the X11 base, X12 introduces a single additional hashing algorithm (HAMSI):

- *Overview*: A contender in the NIST SHA-3 competition [34], HAMSI is designed to be compact and suitable for constrained environments.
- *Features*: Utilizes a series of S-boxes, linear diffusion layers, and MDS matrices. It exhibits a strong resistance against differential cryptanalysis.
- *Relevance to X12*: HAMSI augments X12's resilience by adding its unique cryptographic structure, making the algorithm more suitable for environments with limited computational resources.

5.2. X13 (FUGUE)

Expanding further on its predecessor, X13 adds yet another layer to the established hash chain (FUGUE):

- *Overview*: Another entrant to the NIST SHA-3 competition [34], FUGUE processes data in a unique matrix format, providing both speed and security.
- *Features*: Implements a 3D extension of the AES S-box, combined with a series of rotations, permutations, and non-linear transformations. This layered approach offers heightened security assurances.
- *Relevance to X13*: FUGUE's 3D matrix processing mechanism introduces a novel dimension to the cryptographic chain, strengthening the diversity of techniques within the X13 algorithm.

5.3. X14 (SHABAL)

The X14 hashing scheme adds one more hash function to the already intricate X13 system (SHABAL):

- *Overview*: Participating in the NIST SHA-3 race [34], SHABAL stands out due to its symmetric structure and ability to process vast amounts of data efficiently.
- *Features*: Employs a unique combination of bitwise operations and modular arithmetic. Its design allows for parallel processing, enhancing throughput.

- *Relevance to X14*: SHABAL's symmetric design and parallel processing capabilities further diversify the cryptographic techniques within the X series, ensuring that X14 maintains high efficiency even with its increased complexity.

In summation, the evolution from X11 to X14 signifies the relentless pursuit of cryptographic perfection. By continuously integrating diverse and proven hash functions, the X series ensures that it remains at the forefront of cryptographic security, meeting the ever-increasing demands of blockchain and other cryptographic applications.

6. Results

In the realm of cryptographic hashing, both speed and security are paramount. This section evaluates the performance metrics of multiple cryptographic hash functions, namely SHA2-256, SHA2-512, KECCAK-256, KECCAK-512, X11, X12, X13, and X14.

To provide a holistic perspective, we present the results in three distinct tables (Tables 1-3), each representing a different performance metric: Cycles per Byte, Throughput (MB/sec), and Hash Rate (KHash/sec). This structured approach enables a clearer comparison among these algorithms.

Table 1
Comparative Analysis of Algorithms based on Cycles per Byte

Algorithm	1 byte	2 bytes	4 bytes	8 bytes	16 bytes	32 bytes	64 bytes
SHA2-256	848.68	423.30	211.72	105.80	52.84	26.28	25.74
SHA2-512	945.51	472.36	236.28	118.07	59.05	29.48	14.79
KECCAK-256	1307.78	657.99	328.59	159.08	79.54	39.75	19.88
KECCAK-512	1269.54	633.65	316.76	156.76	77.77	38.83	19.42
X11	35972.07	18012.95	8878.42	4439.23	2220.24	1109.69	554.82
X12	43318.45	21838.60	10907.93	5456.44	2694.58	1347.39	682.47
X13	49509.65	24764.23	12380.19	6191.50	3092.92	1546.94	772.80
X14	51112.37	25668.86	12788.01	6392.37	3189.71	1595.09	797.48

Table 2
Throughput Analysis Across Different Hashing Algorithms

Algorithm	1 byte	2 bytes	4 bytes	8 bytes	16 bytes	32 bytes	64 bytes
SHA2-256	3.06	6.13	12.24	24.53	49.07	97.98	100.50
SHA2-512	2.74	5.49	10.98	21.94	43.93	87.89	175.35
KECCAK-256	1.97	3.94	7.88	16.29	32.54	65.20	130.53
KECCAK-512	2.04	4.09	8.18	16.44	33.31	66.58	133.64
X11	0.07	0.14	0.29	0.58	1.17	2.33	4.67
X12	0.06	0.12	0.24	0.48	0.96	1.91	3.80
X13	0.05	0.10	0.21	0.42	0.84	1.68	3.35
X14	0.05	0.10	0.20	0.41	0.81	1.62	3.24

Table 3
Hash Rate Efficiency Across Various Hashing Algorithms

Algorithm	1 byte	2 bytes	4 bytes	8 bytes	16 bytes	32 bytes	64 bytes
SHA2-256	3060.37	3065.65	3059.57	3066.73	3066.73	3061.86	1570.25
SHA2-512	2741.59	2744.53	2744.68	2742.09	2745.54	2746.69	2739.80
KECCAK-256	1974.12	1969.97	1970.41	1970.63	1970.89	1970.56	1970.89
KECCAK-512	2041.02	2045.36	2044.80	2045.45	2044.98	2044.87	2045.01
X11	71.73	72.33	72.98	73.14	73.14	73.12	73.08
X12	59.35	59.34	59.35	59.36	59.36	59.37	59.37
X13	52.36	52.33	52.34	52.34	52.35	52.35	52.36

X14	50.75	50.67	50.67	50.68	50.68	50.69	50.70

The "Cycles per Byte" metric gives an insight into the computational efficiency of a hashing algorithm. Lower values are indicative of better efficiency. As observed, both SHA2 variants, especially SHA2-512, and the KECCAK variants show significantly lower cycle counts compared to the X series (X11 to X14). Amongst the X series, X11 is the most efficient, with X14 being the least. It's crucial to note that the cycle counts for SHA2 and KECCAK increase (i.e., improve in efficiency) as the input block size increases, reaching an optimum at 64 bytes for SHA2-256.

Throughput, measured in MB/sec, reflects the data processing speed of the hashing algorithm. Higher values indicate superior performance. The SHA2-256 algorithm clearly dominates, with its throughput peaking at 100.50 MB/sec for a 64-byte input block size. Its counterpart, SHA2-512, closely follows this trend, but with roughly half the throughput for the same block size. The KECCAK variants, while less efficient than the SHA2 series, substantially outperform the X series in this regard. The consistently low throughput values for the X series, especially for smaller block sizes, might raise concerns in scenarios where rapid data processing is paramount.

The hash rate, presented in KHash/sec, provides a measure of how quickly an algorithm can compute a hash. Similar to throughput, higher values suggest better performance. Notably, SHA2-256 and SHA2-512 show remarkable efficiency, consistently outperforming other algorithms across varying block sizes. However, it's intriguing to note a sharp decline in the hash rate of SHA2-256 for a 64-byte block size. This deviation warrants further investigation. The KECCAK series maintains a steady hash rate across all block sizes, suggesting consistent performance. The X series algorithms, while lagging behind their counterparts, show stability in their hash rates.

In summary, the SHA2 series, especially SHA2-256, generally showcases superior efficiency in terms of cycles per byte, throughput, and hash rate, followed by the KECCAK variants. The X series, while not as efficient, provides a consistent hash rate, which might be suitable for applications that prioritize consistency over raw performance. However, it's essential to consider other factors such as security robustness, application context, and system compatibility when selecting an appropriate hashing algorithm.

7. Discussion

The recent developments in the field of cryptographic hashing algorithms have reignited the discourse surrounding efficiency versus security, particularly in the context of Application-Specific Integrated Circuit (ASIC) resistance. While it's evident from the data that the SHA-2 and KECCAK families demonstrate superior performance metrics in terms of computational efficiency, throughput, and hash rates, their dominance raises pivotal concerns in the ever-evolving arms race against ASIC miners.

ASICs are specialized hardware designed explicitly for a specific computational task. In the cryptocurrency world, ASICs tailored for certain hashing algorithms can perform mining operations orders of magnitude faster than general-purpose computing hardware. This advantage not only centralizes the mining power, often leading to fewer entities controlling the mining process, but also poses a threat to the security and democratization of the network.

In this light, the X series of algorithms, despite their apparent lag in raw performance metrics, emerge as potential game-changers. Their intricate sequence of cryptographic functions is tailored to deter the advantages ASICs traditionally hold. This level of ASIC-resistance means that, despite the slower speed, there's a leveling of the playing field, making the mining process more accessible and distributed.

The considerably lower performance metrics of the X series, as observed in the tables, aren't flaws but deliberate design decisions. ASIC-resistance often comes at the cost of computational efficiency. By incorporating multiple rounds of hashing with varied algorithms in sequences, the X series ensures that an ASIC designed for one specific function would not necessarily excel in others. This inherent complexity acts as a deterrent, making the ASIC design for such algorithms both challenging and economically unviable.

The slower performance metrics of the X series might raise eyebrows in isolation, but when viewed from the lens of ASIC-resistance, they manifest as necessary trade-offs. In a rapidly evolving digital economy, the balance between efficiency and security is paramount, and the X series embodies this philosophy.

While the lure of superior performance metrics is undeniable, the broader context of network security and democratization cannot be understated. The rise of ASICs threatens the foundational principles of many decentralized systems, and while their influence is undeniable, so too is the necessity for robust countermeasures. The X series, with its deliberative design sacrificing speed for security, underscores the lengths the cryptographic community is willing to go to ensure a level playing field. As we forge ahead, it's vital to remember that the most efficient algorithm is not always the most suitable, especially when the stakes are as high as the security and inclusivity of global decentralized systems.

Discussion on Results with Respect to ASIC Resistance

The Figure 1 illustrates the cycles per byte required for each hashing algorithm across varying input block sizes. Lower values represent higher efficiency.

A cursory glance at the graph indicates that SHA-2 and KECCAK families outperform the X series in terms of raw computational efficiency, requiring fewer cycles per byte. The X series, however, although less efficient, is intentionally designed to mitigate ASIC advantages. The increased computational requirements can be interpreted as a deliberate sacrifice to deter ASIC dominance.

The Figure 2 provides a comparative representation of the throughput (MB/sec) exhibited by each algorithm as the input block size increases.

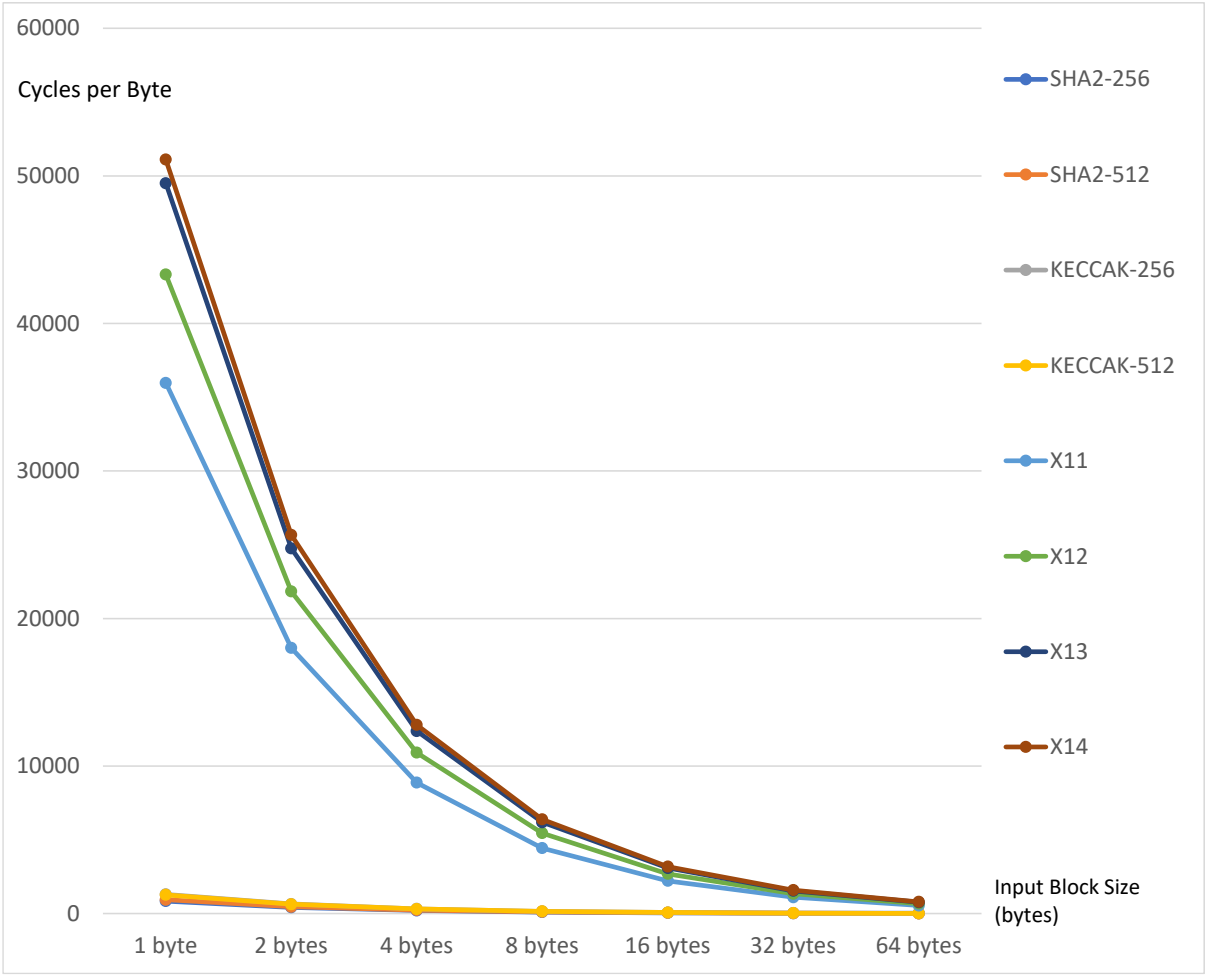

Figure 1: Computational Efficiency Across Hashing Algorithms

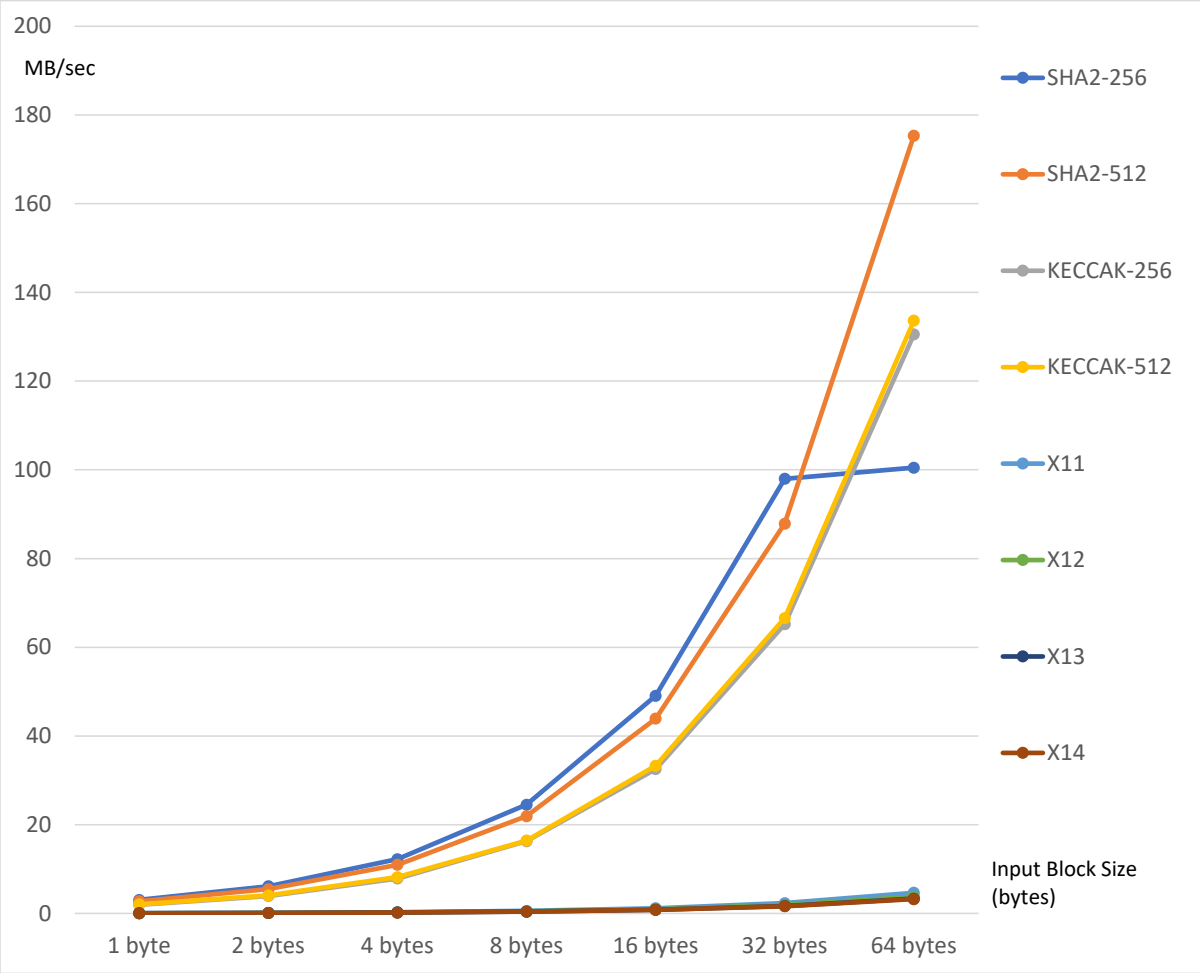

Figure 2: Throughput Performance of Hashing Algorithms

The throughput graph reinforces the notion of SHA-2 and KECCAK's superiority in handling larger data blocks more swiftly. However, the X series' relatively linear scalability showcases its resilience against input variations. This trait might be instrumental in environments where input data size is unpredictable.

The Figure 3 depicts the KHash/sec metrics for each algorithm, providing insights into their hashing speeds.

Once again, the SHA-2 and KECCAK families demonstrate an edge with higher hash rates. The X series, while trailing, maintains consistent performance across input block sizes. The steadiness of the X series, even if slower, might be seen as a testament to its robust and ASIC-resistant design.

The graphs offer a concise yet comprehensive overview of the stark contrasts between traditional, efficient hashing functions and the newer, ASIC-resistant X series. The inherent trade-offs become more pronounced when visualized. While efficiency is an invaluable metric, the need for a decentralized and secure network that resists monopolization by powerful ASIC miners is equally crucial.

The visual aids drive home the point that while some algorithms excel in speed, others are tailored for resilience and security in a constantly evolving digital ecosystem.

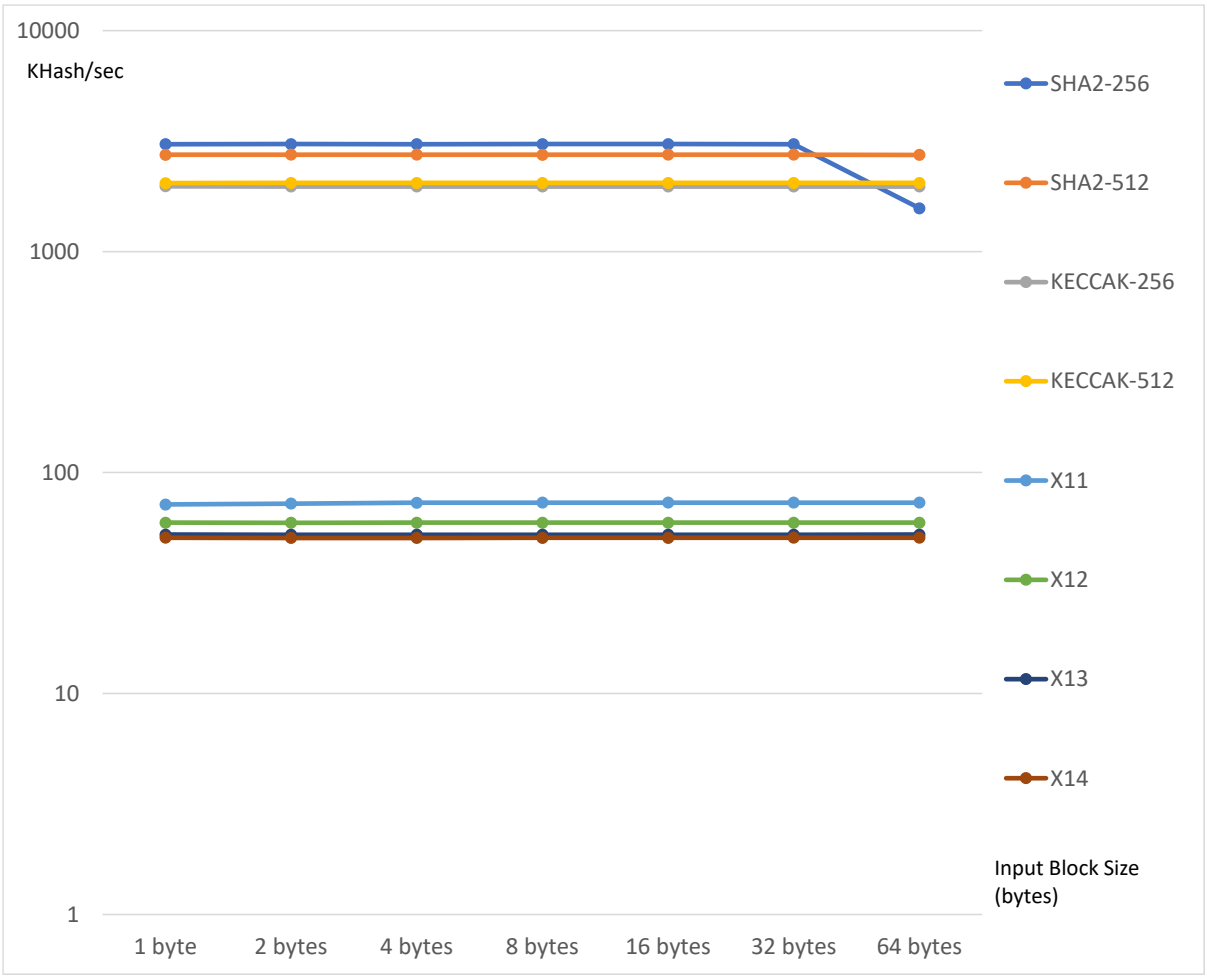

Figure 3: Hash Rate Comparison Across Algorithms

8. Conclusion

In the rapidly evolving landscape of cryptographic hash functions, the quest for efficiency and security operates in tandem, often at cross purposes. Through this study, we endeavored to elucidate the performance dynamics of a diverse array of hashing algorithms, juxtaposing renowned ones like SHA-2 and KECCAK against the ASIC-resistant X series. Our findings underscore a palpable tension between raw computational efficiency and the deliberate design choices made to thwart potential centralization by powerful ASIC miners.

The visual representations generated from our performance metrics unequivocally indicate the superiority of SHA-2 and KECCAK in terms of cycles per byte, throughput, and hash rates. Such proficiency is advantageous in scenarios demanding swift computational outcomes. However, the digital world is not merely governed by the swiftness of computations. The principles of decentralization, equitable access, and resistance to potential monopolization hold paramount importance in ensuring a sustainable and inclusive digital ecosystem.

The X series of hashing algorithms, despite their apparent computational sluggishness, are emblematic of this ideological pivot. Their resilience and consistent performance, even at the cost of raw speed, represent a concerted effort to ensure that the realm of cryptocurrency remains accessible to a broad spectrum of miners, not just those wielding the power of ASICs.

In sum, while efficiency is indisputably a coveted attribute, the broader objectives of decentralization and equitable access necessitate a balanced approach. It's imperative to recognize that in the world of cryptographic hashing, the fastest isn't always the most fitting. As technology continues to advance, the

ongoing challenge for researchers and developers will be to harmonize efficiency with egalitarian principles, ensuring that the digital future remains inclusive and secure for all.

References

[1] A. Tiwari, Chapter 14 - Cryptography in blockchain, in: R. Pandey, S. Goundar, S. Fatima (Eds.), Distributed Computing to Blockchain, Academic Press, 2023: pp. 251–265. https://doi.org/10.1016/B978-0-323-96146-2.00011-5.

[2] What is Hashing Algorithm?, Changelly.Com. (2020). https://changelly.com/blog/hashing-algorithms-explained/ (accessed September 7, 2023).

[3] coinguides, X11 Coins - List of Cryptocurrencies based on X 11 hashing algorithm, Coin Guides. (2019). https://coinguides.org/x11-algorithm-coins/ (accessed September 7, 2023).

[4] coinguides, ASIC resistance explained - Importance of coins with ASIC resistant, Coin Guides. (2018). https://coinguides.org/asic-resistance-explained/ (accessed September 7, 2023).

[5] A. Kuznetsov, I. Oleshko, V. Tymchenko, K. Lisitsky, M. Rodinko, A. Kolhatin, Performance analysis of cryptographic hash functions suitable for use in blockchain, Int. J. Computer Netw. Inf. Security. 13 (2021) 1–15. https://doi.org/10.5815/IJCNIS.2021.02.01.

[6] A. Kuznetsov, K. Shekhanin, A. Kolhatin, D. Kovalchuk, V. Babenko, I. Perevozova, Performance of Hash Algorithms on GPUs for Use in Blockchain, in: IEEE Int. Conf. Adv. Trends Inf. Theory, ATIT - Proc., Institute of Electrical and Electronics Engineers Inc., 2019: pp. 166–170. https://doi.org/10.1109/ATIT49449.2019.9030442.

[7] A. Kuznetsov, M. Lutsenko, K. Kuznetsova, O. Martyniuk, V. Babenko, I. Perevozova, Statistical testing of blockchain hash algorithms, in: Fedushko S., Gnatyuk S., Peleshchyshyn A., Hu Z., Odarchenko R., Korobiichuk I. (Eds.), CEUR Workshop Proc., CEUR-WS, 2019. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083260414&partnerID=40&md5=e65f8fb02e0a57799a5aaea7ccfe195e.

[8] BLAKE2, (n.d.). https://www.blake2.net/ (accessed September 6, 2023).

[9] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, A. Szekely, High-Speed Hardware Implementations of BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, and Skein, (2009). https://eprint.iacr.org/2009/510 (accessed September 7, 2023).

[10] E. Andreeva, A. Luykx, B. Mennink, Provable Security of BLAKE with Non-Ideal Compression Function, (2011). https://eprint.iacr.org/2011/620 (accessed September 7, 2023).

[11] S. Neves, J.-P. Aumasson, Implementing BLAKE with AVX, AVX2, and XOP, (2012). https://eprint.iacr.org/2012/275 (accessed September 7, 2023).

[12] M.E. Hadedy, D. Gligoroski, S.J. Knapskog, Single Core Implementation of Blue Midnight Wish Hash Function on VIRTEX 5 Platform, (2010). https://eprint.iacr.org/2010/571 (accessed September 7, 2023).

[13] M. Rogawski, K. Gaj, Groestl Tweaks and their Effect on FPGA Results, (2011). https://eprint.iacr.org/2011/635 (accessed September 7, 2023).

[14] B. Jungk, S. Reith, J. Apfelbeck, On Optimized FPGA Implementations of the SHA-3 Candidate Groestl, (2009). https://eprint.iacr.org/2009/206 (accessed September 7, 2023).

[15] Hash Function JH; designed by Hongjun Wu, (n.d.). https://www3.ntu.edu.sg/home/wuhj/research/jh/ (accessed September 7, 2023).

[16] R. Bhattacharyya, A. Mandal, M. Nandi, Security Analysis of the Mode of JH Hash Function, in: S. Hong, T. Iwata (Eds.), Fast Software Encryption, Springer, Berlin, Heidelberg, 2010: pp. 168–191. https://doi.org/10.1007/978-3-642-13858-4_10.

[17] J.-P. Aumasson, C. Calik, W. Meier, O. Ozen, R.C.-W. Phan, K. Varici, Improved Cryptanalysis of Skein, (2009). https://eprint.iacr.org/2009/438 (accessed September 7, 2023).

[18] T. Oliveira, J. López, Improving the performance of Luffa Hash Algorithm, (2010). https://eprint.iacr.org/2010/457 (accessed September 7, 2023).

[19] O. Dunkelman, E. Biham, The SHAvite-3 - A New Hash Function, in: H. Handschuh, S. Lucks, B. Preneel, P. Rogaway (Eds.), Symmetric Cryptography, Schloss Dagstuhl – Leibniz-Zentrum

für Informatik, Dagstuhl, Germany, 2009: pp. 1–39. https://doi.org/10.4230/DagSemProc.09031.18.
[20] J. Jean, M. Naya-Plasencia, M. Schläffer, Improved Analysis of ECHO-256, (2011). https://eprint.iacr.org/2011/422 (accessed September 7, 2023).
[21] S. Halevi, W.E. Hall, C.S. Jutla, The Hash Function "Fugue," (2014). https://eprint.iacr.org/2014/423 (accessed September 7, 2023).
[22] Shabal, (2009). https://web.archive.org/web/20091209170807/http://www.shabal.com/ (accessed September 6, 2023).
[23] S.-W. Lee, I. Singh, M. Mohammadian, eds., Blockchain Technology for IoT Applications, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-4122-7.
[24] K. Xu, J. Zhu, X. Song, Z. Lu, eds., Blockchain Technology and Application: Third CCF China Blockchain Conference, CBCC 2020, Jinan, China, December 18-20, 2020, Revised Selected Papers, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-6478-3.
[25] S. Van Hijfte, Bitcoin, in: S. Van Hijfte (Ed.), Blockchain Platforms: A Look at the Underbelly of Distributed Platforms, Springer International Publishing, Cham, 2020: pp. 71–146. https://doi.org/10.1007/978-3-031-01804-6_2.
[26] R.C. Merkle, A Digital Signature Based on a Conventional Encryption Function, in: C. Pomerance (Ed.), Advances in Cryptology — CRYPTO '87, Springer, Berlin, Heidelberg, 1988: pp. 369–378. https://doi.org/10.1007/3-540-48184-2_32.
[27] P. Rogaway, T. Shrimpton, Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance, in: B. Roy, W. Meier (Eds.), Fast Software Encryption, Springer, Berlin, Heidelberg, 2004: pp. 371–388. https://doi.org/10.1007/978-3-540-25937-4_24.
[28] S. Turner, Using SHA2 Algorithms with Cryptographic Message Syntax, Internet Engineering Task Force, 2010. https://doi.org/10.17487/RFC5754.
[29] D.R.L. Brown, T. Polk, S. Santesson, K. Moriarty, Q. Dang, Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA, Internet Engineering Task Force, 2010. https://doi.org/10.17487/RFC5758.
[30] J. Harvey-Buschel, C. Kisagun, Bitcoin Mining Decentralization via Cost Analysis, (2016). https://doi.org/10.48550/arXiv.1603.05240.
[31] K. Ivy, The Looming Threat of Centralized Cryptocurrency, iM Intelligent Mining. (2022). https://medium.com/im-intelligent-mining/the-looming-threat-of-centralized-cryptocurrency-562dcc460eae (accessed September 7, 2023).
[32] L. Bex, F. Turan, M. Van Beirendonck, I. Verbauwhede, Mining CryptoNight-Haven on the Varium C1100 Blockchain Accelerator Card, (2022). https://doi.org/10.48550/arXiv.2212.05033.
[33] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Keccak, in: T. Johansson, P.Q. Nguyen (Eds.), Advances in Cryptology – EUROCRYPT 2013, Springer, Berlin, Heidelberg, 2013: pp. 313–314. https://doi.org/10.1007/978-3-642-38348-9_19.
[34] I.T.L. Computer Security Division, SHA-3 Project - Hash Functions | CSRC | CSRC, CSRC | NIST. (2017). https://csrc.nist.gov/projects/hash-functions/sha-3-project (accessed September 6, 2023).