Comparative Analysis of Cryptographic Hashing Algorithms for Blockchains

Atharva Dongare[1], Sara Ataullah[2], Sonal Gaikwad[3], Ioan Raicu[4]

[1,2,3,4] Illinois Institute of Technology {adongare@hawk.iit.edu, sataullah@hawk.iit.edu, sgaikwad4@hawk.iit.edu, iraicu@iit.edu}

Cryptographic Hashes

- Cryptographic hashing functions are mainly inherent in blockchain technology, which works as a decentralized and distributed network storing information in a linear chain of blocks.
- Each block comprises the hash of its own block and the previous block, thereby formulating a cryptographically secure chain of blocks.

Hashing Algorithms

SHA-256
Generates a 256-bit (32-byte) hash value from any input data. The security of SHA-256 algorithm is based on its resistance to collision attacks.

Blake3
Operates on variable-length inputs and produces fixed-length 256-bit or 512-bit outputs. It is designed to be memory-hard.

MD5
Produces a fixed-size 128-bit hash value from arbitrary length input messages. Operates by processing the input message in 512-bit blocks and uses a series of logical operations to transform each block into a 128-bit message.

RipeMD-160
Generates a fixed-size 160-bit output using arbitrary length input messages. It is designed to be resistant to collision attacks and pre-image attacks and makes use of arithmetic and logical operations to transform each block of the input message into a 160-bit message digest.

Keccak-256
Takes a variable-length input message and produces a fixed-size 256-bit output, known as the message digest. It is designed to be secure against a wide range of attacks such as collision attack, pre-image attack and birthday attack.

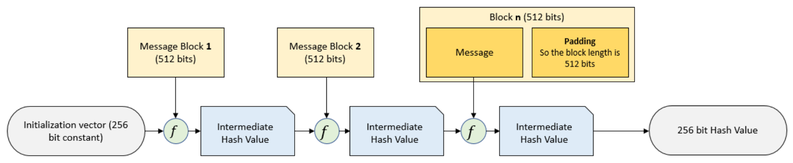
Algorithm	Motivation	Cryptocurrencies
SHA-256	SHA-256 is a widely adopted algorithm and proffers strong resistance to collision attacks while also being fast and efficient with relatively low computational overhead	Bitcoin, Litecoin, NEO
Blake 3	Relatively new and versatile hash function that supports various input and output sizes. Faster and efficient than SHA-256.	Chia, Solana
MD5	MD5 is faster than SHA-256 and SHA-1. Results obtained have the same complexity as SHA-256. MD5 also proffers relatively lower computational head.	Elastic (XEL)
RipeMD-160	RipeMD-160 is a widely adopted algorithm that offers better speed than SHA-256	Bitcoin, Dash
Keccak-256	Keccak-256 is widely used and is part of Ethash and Quark. It also proffers high performance and versatility.	Ethereum, Cardano

Experimental Setup

The codebase was partitioned into two distinct segments, one designed for single-threaded execution and the other for multi-threaded execution. To gauge performance, a set of system monitoring tools, including powertop for power consumption analysis, perf for cache behavior analysis, and htop for thread observation, were employed. Performance metrics such as hash rate, cache hits and power consumption.

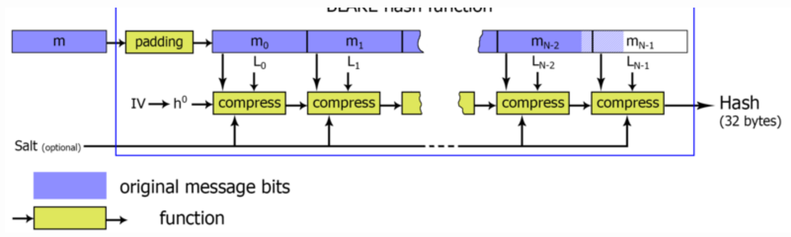
SHA-256

Produces a 256-bit hash value and is part of the Secure Hash Algorithm (SHA) family. Predominantly used in digital security applications such as digital signatures and password hashing. The SHA-256 algorithm processes input messages in 512-bit blocks and uses a set of 64 rounds that manipulate the hash values and block data in a complex way.



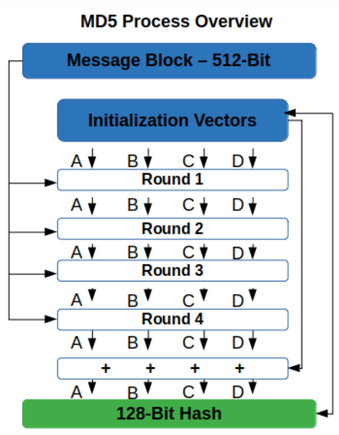
Blake3

Blake3 is based on the Blake2 family and is designed to proffer high performance, ease of use and strong security guarantee. Blake3 processes input messages in fixed-size chunks and uses a binary hash tree to efficiently compute intermediate hashes.



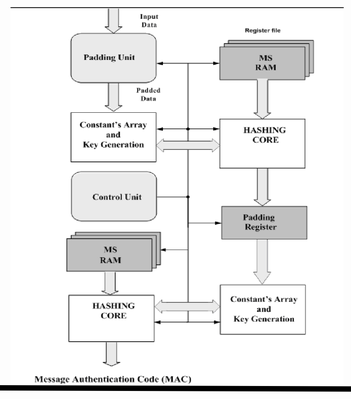
MD5

MD5 is designed to be a one-way function, meaning that it should be computationally infeasible to find two different input messages that produce the same hash value. It is used in some applications, such as checksums for verifying the integrity of downloaded files or for password hashing.



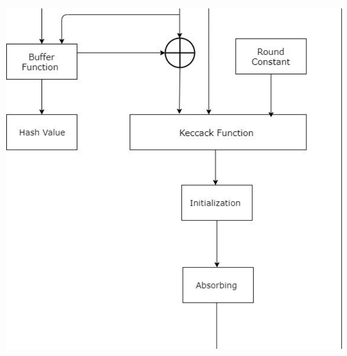
RIPEMD-160

It processes input messages in 512-bit blocks and uses a set of five rounds that manipulate the hash values and block data in a complex way. RIPEMD-160 is designed to be a one-way function.



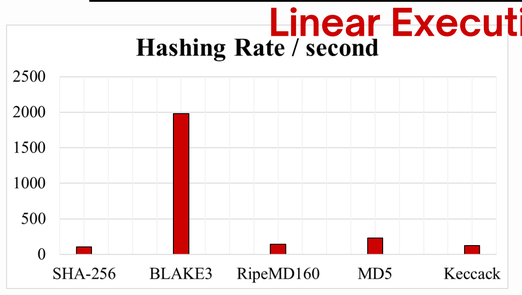
Keccak-256

Keccak-256 is designed to resist a variety of cryptanalytic attacks, including collision attacks, preimage attacks, and length extension attacks.

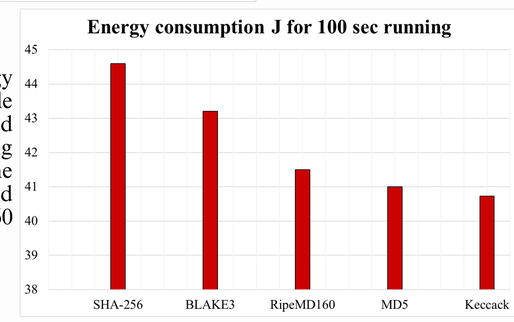


Comparative Analysis

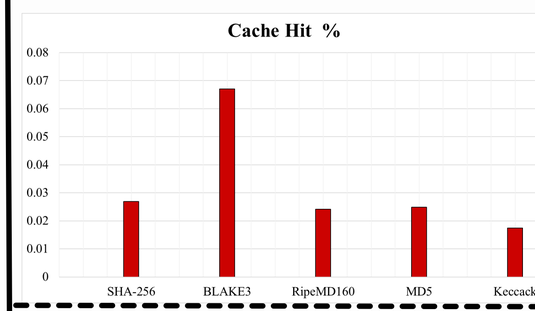
Linear Execution



During Single thread execution for finding the hashing rate, have observed best performance in the case of BLAKE3 algorithm, which displayed about 10 times increase.

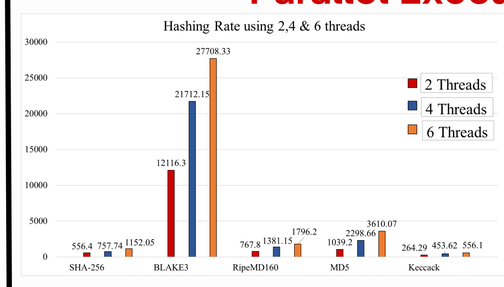


Observing the energy consumption while performing single thread execution led to observing best performance in the case of Keccak, followed by MD5 and RipeMD160 algorithms.



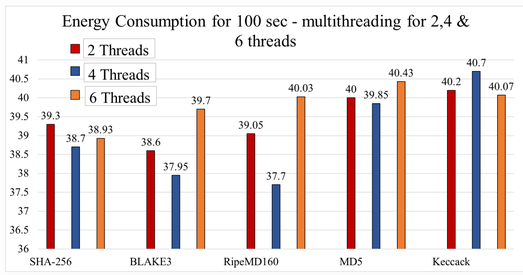
During the analysis of the algorithm with respect to Cache rate, observed that BLAKE3 achieved best cache access performance and was sometimes also observed that the performance increased as compared to the normal percentage of Cache access rate.

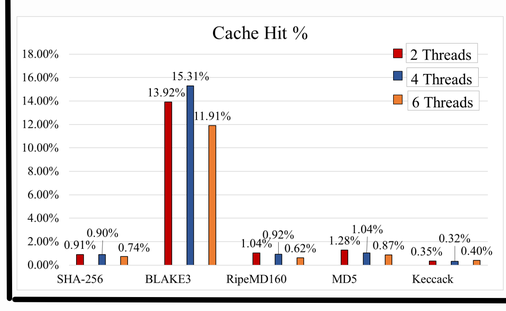
Parallel Execution



In the case of parallel execution for observing the Hashing rate, saw a steady rise for 2, 4 and 6 threads and thus saw similar results where BLAKE3 achieved best performance among the chosen algorithms.

During the analysis of Energy consumption across a parallel execution approach, found better performance in the case of 4 threads at most times. Also, the best algorithms with respect to energy consumotion turned out to be RipeMD160 and BLAKE3.





Suprisingly, during the parallel approach observed better than normal Cache access percentage in the case of BLAKE3. This improved performance is atleast 10% better as compared to the others.

Conclusion

The study, analysis and experimentation confirmed some expected findings like better hashing rate in the case of BLAKE3, but also led us to unexpected findings like BLAKE3's better than average cache performance in the parallel approach. Thus from the perspective of Hashing rate and Cache performance noted that BLAKE3 is the best among the chosen algorithms. While based on power consumption, in a single thread execution observed best performance for Keccak algorithm and in the case of the parallel approach observed least energy was consumed by RipeMD160 and BLAKE3.

ILLINOIS TECH