Grenze International Journal of Engineering and Technology, January Issue

Security Analysis of Cryptographic Algorithms Including SHA-256, SHA-3 & ECC in the context of Mining in Blockchain Technology

Meruva Ashish[1] and Arun Kumar B.R[2]
[1-2]BMS Institute of Technology & Management,
Dept. of Computer Science and Engineering, Bengaluru, India
Email: 1by20cs110@bmsit.in, arunakumarbr@bmsit.in

***Abstract*—Cryptography stands as a pivotal tool in safeguarding data integrity and confidentiality, particularly in the face of escalating cyber threats including Blockchain technology. This research delves into the realm of cryptographic algorithms, focusing on their pivotal role in ensuring secure communication over networks. The study analyses three prominent algorithms: DES, Triple DES, and RSA, representing both symmetric and asymmetric cryptographic techniques as well as SHA-256 and SHA-3 in the context of Blockchain Technology. The analysis of through rigorous examination and assessments is presented based on their efficacy in securing data, encryption speed, and overall throughput.**

***Index Terms*—Cryptography, Blockchain, Cryptanalysis, Hashing.**

I. Introduction

In the realm of blockchain technology, ensuring robust security measures against potential cyber threats is paramount. Cryptographic algorithms play a pivotal role in safeguarding the integrity and confidentiality of data exchanged within the blockchain network. This study delves into the intricacies of various cryptographic algorithms utilized in blockchain mining operations, encompassing both symmetric (e.g., DES, 3DES, CAST-128, BLOWFISH, IDEA, AES, RC6) and asymmetric (e.g., RSA) techniques.
Furthermore, the research investigates key parameters such as key exchange mechanisms, adaptability, and inherent security vulnerabilities associated with these algorithms. Through a comprehensive analysis, this study aims to provide valuable insights into the efficacy and dependability of cryptographic systems in the context of blockchain mining. By scrutinizing these factors, stakeholders can make informed decisions in selecting the most appropriate cryptographic algorithm, one that not only upholds stringent data security standards but also optimizes performance in real-world blockchain environments.

II. Comparison of Algorithms

A. Data Encryption Standard

The Data Encryption Standard (DES) operates on a cipher framework pioneered by Horst Feistel, an esteemed IBM cryptographer, during the early 1970s. This framework, termed the Feistel block cipher, comprises a series of rounds, each incorporating intricate processes such as bit-shuffling, nonlinear substitutions via S-boxes, and

Grenze ID: 01.GIJET.11.1.348

exclusive OR operations. Operating on 64-bit blocks of data, DES utilizes a singular algorithm and key for both encryption and decryption purposes, with the key spanning 56 bits in length. Notably, specific bit positions, including 8, 16, 24, 32, 40, 48, 56, and 64, are discarded during the encryption process.

DES epitomizes two cardinal principles of cryptography: Diffusion (Substitution) and Confusion (Permutation), manifesting through its architecture comprising 16 rounds. Within each round, the interplay of key and data bits entails intricate shifts, permutations, XOR operations, and traversal through eight S-boxes. Commencing with an initial permutation (IP) stage, where a 64-bit plaintext undergoes partitioning into left and right halves, subsequent rounds see these halves navigating through a series of transformations. Ultimately, after traversing all 16 rounds, the left and right plaintext halves are reunified. The decryption process mirrors the encryption process, albeit in reverse order, with rounds executed in an inverted sequence to unveil the original plaintext.

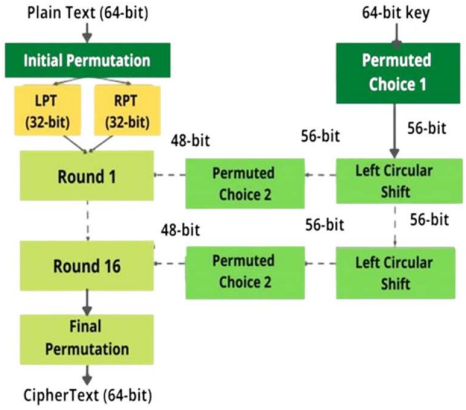


Fig1. Working of DES Algorithm

B. Triple Des (3DES)

Derived from the Data Encryption Standard (DES), Triple DES (3DES) employs three distinct keys, each comprising 56 bits, culminating in a total key size of 168 bits. This cryptographic scheme offers three keying options:

Independent Keys: Each of the three keys operates independently, rendering this option the most robust, with a total of 168 independent key bits.

Identical Keys: All three keys are identical, representing the weakest option and equating to the security level of DES.

Partially Identical Keys: The first and third keys are identical, necessitating 48 rounds of processing as DES is applied thrice for encryption.

Triple DES (3DES), with a combined 168-bit key size, significantly enhances security compared to DES. Despite its robustness, 3DES faces scrutiny due to its association with DES. Nevertheless, it's widely used in Internet protocols, albeit vulnerable to certain attacks. Positioned as an upgrade to DES, 3DES applies the encryption method thrice to counter vulnerabilities and boasts proven reliability and a heightened key length.

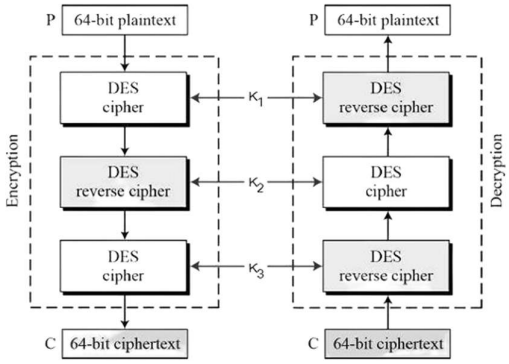


Fig.2 Working of 3DES Algorithm

C. RSA

Developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman, RSA is a renowned public key encryption algorithm known for its versatility in secrecy and digital signature authentication. Utilizing prime numbers for key generation, RSA operates on block-sized data, typically consisting of 1024 bits or 309 decimal digits. Unlike symmetric key algorithms, RSA uses distinct keys for encryption and decryption, enhancing security.

At its core, RSA employs an asymmetrical key structure created through the process of factoring the product of two large prime numbers. Despite its robustness, RSA faces vulnerabilities when small or closely spaced primes are used in key generation, compromising its effectiveness. Nevertheless, RSA remains a pillar of encryption technology, valued for its robust security measures and enduring cryptographic resilience.

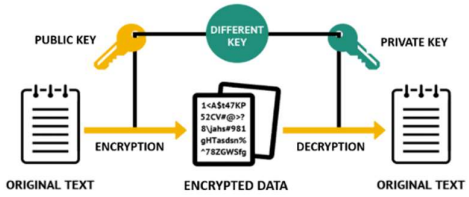


Fig.3 Working of RSA Algorithm

D. CAST-128

CAST-128, a block cipher algorithm rooted in the Feistel function, operates through 12 to 16 processing rounds. With a block size of 64 bits and a variable key length ranging from 40 to 128 bits, CAST-128 employs 16 rounds when the key size exceeds 80 bits. Notably, CAST enhances security by accommodating variable key sizes of 128 and 256 bits, bolstering resilience against both linear and differential attacks.

In cryptographic contexts, CAST-128's security is measured by its resistance to various attack vectors. While it demonstrates robustness against many threats, such as linear and differential attacks, it is not impervious. Notably, CAST-128 can be compromised by a substantial number of chosen plaintexts, which can lead to vulnerabilities. Moreover, the 64-bit key version of CAST-128 is particularly susceptible to differential related-key attacks, highlighting a potential weakness in its security posture.

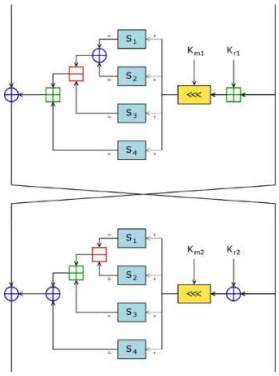


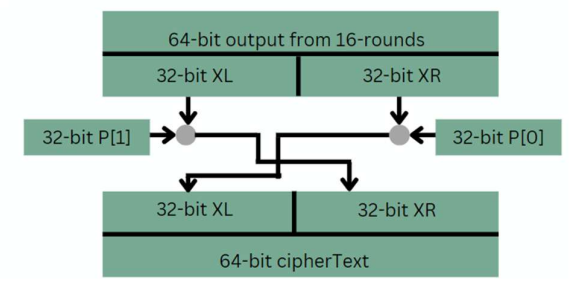
Fig.4 Working of CAST-128 Algorithm

E. Blowfish

Blowfish, a robust block encryption algorithm, utilizes a 64-bit block and offers key sizes from 32 to 448 bits. It employs 16 processing rounds, with key expansion and data encryption as core functions. Unlike some methods, Blowfish's substitution boxes are independent of keys, enhancing security.

While Blowfish ensures long-term data security and lacks known backdoor vulnerabilities, it faces challenges. Its reliance on variable key lengths increases processing time due to intricate sub-key generation. This complexity deters brute-force attacks but increases computational overhead.

Blowfish's reliability can be compromised by weak keys, reducing overall robustness. Additionally, the first four encryption rounds are susceptible to second-order differential attacks, emphasizing the need for comprehensive security measures.



F. International Data Encryption Algorithm (IDEA)

IDEA, the International Data Encryption Algorithm, is a robust symmetric key block encryption system known for its substitution and permutation mechanisms. With 8.5 rounds of processing, IDEA combines XOR operations, additions, and multiplications to create a formidable defense against unauthorized access.

Operating on a 64-bit block and utilizing a 128-bit key, IDEA demonstrates resilience against differential cryptanalysis and various attacks, bolstered by multiple group operations. The adoption of a 128-bit key size enhances IDEA's security posture, although potential collision attacks remain a concern.

Despite its strengths, IDEA faces vulnerabilities, particularly in the first three processing rounds, where it may be susceptible to key-schedule attacks and key-related differential timing attacks. As such, ongoing vigilance and proactive security measures are crucial for maintaining IDEA's efficacy as a formidable encryption tool.

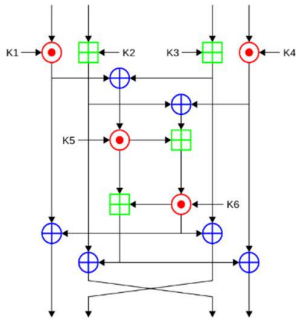


Fig.6 Working of IDEA Algorithm

G. Advanced Encryption Standard (AES)

AES, the Advanced Encryption Standard, is a leading block cipher algorithm known for its innovative cryptographic security. It features a 128-bit block size and offers key lengths of 128, 192, and 256 bits for adaptable security. The number of encryption rounds varies based on the key length, ranging from 10 to 14 rounds.

During encryption, AES executes key expansion, sub-byte generation, column mixing, and round key addition, enhancing resistance against unauthorized access. While AES has withstood numerous attacks, such as square and differential attacks, it remains susceptible to combined boomerang and rectangle attacks on reduced-round versions. Ongoing vigilance and proactive security measures are essential for safeguarding cryptographic systems.

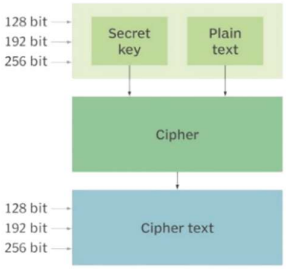


Fig.7 Working of AES Algorithm

H. RC6

RC6, rooted in the trusted Feistel Structure, is a robust cryptographic solution, refining its predecessor RC5 with a 128-bit block size and adaptable key sizes of 128, 192, or 256 bits. Unlike RC5, RC6 employs four registers, enhancing its capabilities and security against various attacks, including differentials and cryptanalysis. The algorithm's strength lies in its efficient manipulation of rotation amounts during processing, offering robust defence against brute-force attacks. However, RC6's security effectiveness depends on factors like key size and the recommended 20 rounds for optimal protection.

Despite its resilience, RC6 has vulnerabilities, especially with small key sizes, making it susceptible to differential and brute-force attacks. Additionally, RC6's processing time can be significant, requiring careful consideration of computational overheads during implementation.

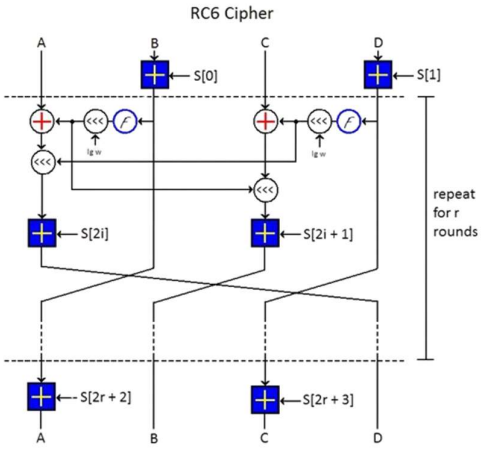


Fig.8 Working of RC6 Algorithm

Blockchain technology is a decentralized and distributed ledger system that enables secure and transparent record-keeping of digital transactions across a network of computers. It gained prominence as the underlying technology behind cryptocurrencies like Bitcoin, but its applications extend far beyond digital currencies to include supply chain management, voting systems, identity verification, and more.

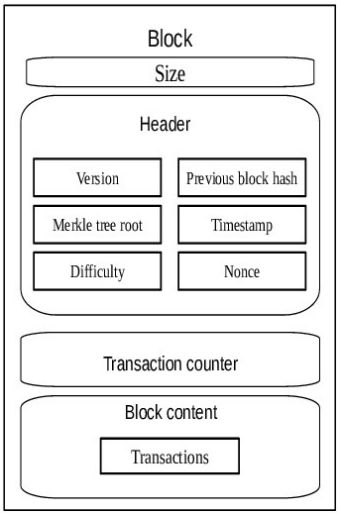


Fig.9 Block Architecture

At its core, a blockchain consists of a chain of blocks, with each block containing a list of transactions. These blocks are cryptographically linked together, forming a continuous and immutable chain. The architecture of a block typically includes the following components:

- Block Header: This contains metadata about the block, including a timestamp, a reference to the previous block (known as the hash of the previous block), and a nonce (a random number used in the mining process).
- Transactions: These are records of digital transactions, such as the transfer of cryptocurrencies or the execution of smart contracts. Each block can contain multiple transactions, grouped together and verified by network participants.
- Merkle Tree Root: A Merkle tree is a hierarchical data structure that summarizes the transactions within a block. The Merkle tree root, also known as the Merkle root, is a cryptographic hash of all transactions in the block, enabling efficient verification of the block's integrity.

Mining adds new blocks to the blockchain by solving complex mathematical puzzles with cryptographic algorithms. Common mining algorithms include:
Proof of Work (PoW): Used in blockchains like Bitcoin, miners solve computationally intensive puzzles to find a nonce that produces a hash value below a target threshold. SHA-256 is a common hash function in PoW mining.
Proof of Stake (PoS): An alternative consensus mechanism where validators are chosen based on their stake in the cryptocurrency. PoS prioritizes validators with higher stakes, reducing the need for intensive computational work.
A few other such algorithms include SHA-256 and SHA-3 which we'll dive more deeper into later.
Overall, blockchain technology offers a decentralized and tamper-resistant framework for digital transactions, with mining serving as the mechanism for maintaining the integrity and security of the blockchain network

I. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a modern cryptographic technique that enhances data encryption using public and private key pairs. Unlike RSA, which relies on prime factorization complexity, ECC utilizes elliptic curves' mathematical properties for robust security. ECC's efficiency lies in generating secure key pairs, enabling seamless encryption across digital platforms. ECC offers smaller key sizes than RSA, making it ideal for resource-constrained environments like mobile devices. Its mathematical foundation ensures high security with compact keys, meeting the demands of secure communication in the mobile era.
As an alternative to RSA, ECC represents a significant advancement in public key cryptography, providing a balance between security and performance. Organizations benefit from ECC's ability to protect data effectively. As ECC adoption grows, continued research and education are crucial for securing cryptographic operations.

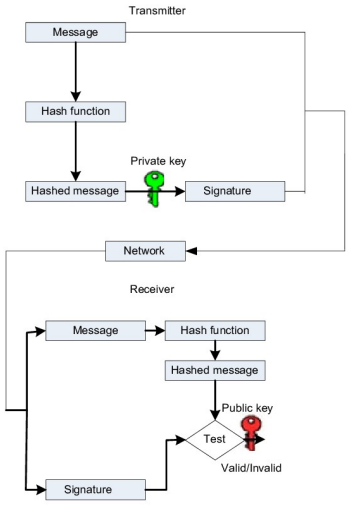


Fig.10 Working of ECC Algorithm

J. SHA-256

The SHA-256 algorithm stands as a robust cryptographic tool renowned for its capacity to safeguard data integrity with its cryptographic hash function. This function generates unique 256-bit hash values for each input, making reverse-engineering exceptionally challenging and ensuring high collision resistance. The algorithm's deterministic nature guarantees consistent verification and authentication processes, striking a harmonious

balance between security and efficiency. Embraced as a NIST-approved standard, SHA-256 finds widespread application across domains like digital signatures, password storage, blockchain technology, and data integrity verification.

SHA-256 operates as a one-way function, making it difficult to reverse-engineer the original input from the hash value. Its intricate mathematical operations generate unique 256-bit hash values, essential for maintaining data integrity, enabling digital signatures, and fortifying digital transactions. With diverse applications, including data integrity, authenticity verification, and secure password hashing, ongoing research aims to enhance its resilience against emerging security threats.

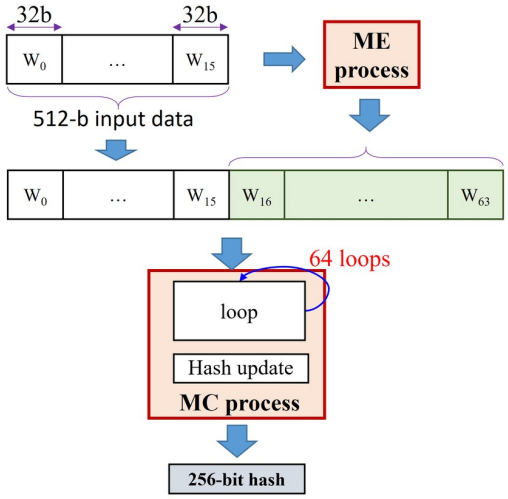


Fig.11 Working of SHA-256 Algorithm

K. SHA-3

SHA-3, or Secure Hash Algorithm 3, is a family of cryptographic hash functions specified in the Federal Information Processing Standards (FIPS) 202. Derived from the KECCAK algorithm, SHA-3 includes SHA3-224, SHA3-256, SHA3-384, and SHA3-512, along with two extendable-output functions (XOFs) called SHAKE128 and SHAKE256. It transforms input data into multidimensional state arrays through multiple rounds of permutations using five distinct permutation functions. These transformations result in fixed-length output strings, ensuring data integrity and security.

With digest lengths ranging from 128 to 512 bits, SHA-3 caters to diverse cryptographic needs in various applications. SHAKE128 and SHAKE256 provide extendable-output functions, enhancing their versatility. SHA-3's comprehensive suite of algorithms makes it indispensable in modern cryptographic protocols, ensuring robust security and facilitating secure data processing.

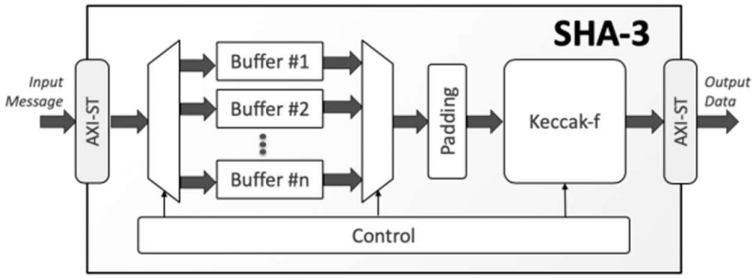


Fig.12 Working of SHA-3 Algorithm

III. Comparison of Algorithms

From the aforementioned Cryptographic Algorithms, we will compare different parameters such as structure, security, flexibility, key length, block size, cipher type, speed, efficiency and process rounds.

TABLE I. QUALITATIVE ANALYSIS OF ALGORITHMS

Algorithm	Structure	Flexibility	Known Attacks	Key Length
DES	Feistel	NO	Brute Force Attack	56
3DES	Feistel	YES	Brute Force Attack, Chosen Plaintext, Known Plaintext	112-168
CAST-128	Feistel	YES	Chosen Plaintext Attack	40-128
BLOWFISH	Feistel	YES	Dictionary Attack	32-448
IDEA	Substitution-Permutation	NO	Differential Timing Attack, Key-Schedule Attack	128
AES	Substitution-Permutation	YES	Side Channel Attack	128-256
RC6	Feistel	YES	Brute Force Attack, Analytical Attack	128-256
RSA	Factorization	YES	Factoring the Public Key	1024
SHA-256	Merkle–Damgård construction	YES	Brute Force Attack, Chosen Plaintext, Known Plaintext, Length Extension, Side-Channel, Differential Cryptanalysis	256
SHA3	Sponge	YES	Brute Force Attack, Chosen Plaintext, Known Plaintext, Length Extension, Side-Channel, Differential Cryptanalysis	224-512
ECC	Elipse	YES	Invalid Curve, Side-Channel, Quantum, Random Number	128-512

TABLE II. QUALITATIVE ANALYSIS OF ALGORITHMS

Algorithm	Rounds	Block size	Speed	Security
DES	16	64 bits	21.340 MB/s	Not Secure Enough
3DES	48	64 bits	0.002663 MB/s	Highly Secure
CAST-128	16	64 bits	50 MB/s to 200 MB/s	Relatively Secure
BLOWFISH	16	64 bits	8.3MB/s	Robust Security
IDEA	8	128 bits	50MB/s to 300MB/s	Good Security
AES	10-14	128 bits	55MB/s	Best Security
RC6	20	128 bits	50 MB/s to 300 MB/s	Secure
RSA	1	Bit Dependent	4MB/s	Least Secure
SHA-256	1	512	512 MB/s to 1024MB/s	Great Security
SHA3	1	1088 bits	50 MB/s to 200 MB/s	Great Security
ECC	1	NIL	0.1 MB/s to 1MB/s	Best Security

The security of cryptographic algorithms hinges on their resilience against diverse attacks. Key factors influencing algorithmic performance include structure, key length, block size, the number of rounds employed, and cryptographic processing time. These elements collectively determine the security efficacy of a given algorithm. Of particular significance is the block size, serving as the fundamental unit of data for encryption and decryption processes.

Increasing the block size can enhance security, all else being equal, in certain algorithms. For instance, AES employs a block size of 128 bits, which is double the size of blocks utilized in other symmetric algorithms under consideration.

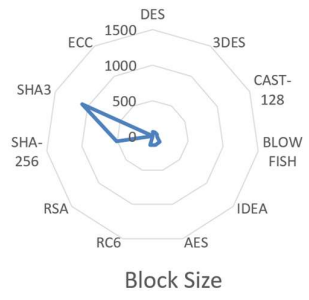


Fig.13 Block Size Comparison of Algorithms

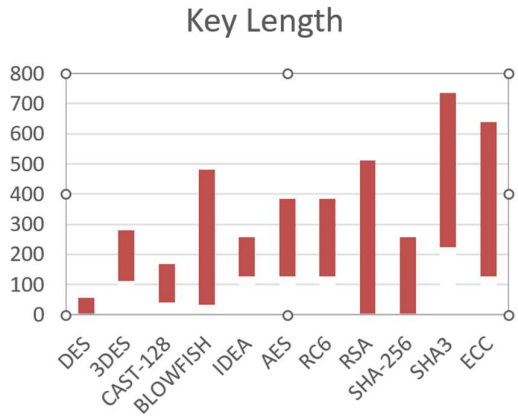


Fig. 14 Key Length Comparison of Algorithms

An essential aspect to consider is the number of rounds employed in the encryption and decryption process. Augmenting the number of processing rounds significantly enhances security, as a solitary Feistel round typically fails to provide sufficient protection. Notably, DES and BLOWFISH utilize 16 rounds of processing, while 3DES triples this count to 48 rounds. The number of rounds in AES varies depending on the key size, ranging from 10 to 14 rounds. However, RC6 emerges as the optimal choice in this regard, boasting a formidable 20-round processing cycle. A critical vulnerability faced by symmetric key algorithms is the brute force attack, wherein every possible key is systematically tested until the correct one is discovered to decrypt the message.

When considering the speed of cryptographic algorithms, DES operates relatively fast due to simplicity, clocking in at 21.340 MB/s. However, 3DES, performing triple encryption, is notably slower, with a speed of 0.002663 MB/s. CAST-128 achieves moderate to high speeds owing to its simple structure, ranging from 50 MB/s to 200 MB/s, while BLOWFISH is slower due to its complex key scheduling, operating at 8.3MB/s. IDEA strikes a balance between simplicity and efficiency, offering moderate to high speeds, ranging from 50MB/s to 300MB/s. AES, recognized for its balance between security and performance, achieves relatively high speeds of 55MB/s. RC6 also operates at moderate to high speeds, benefiting from its efficient structure, with speeds ranging from 50 MB/s to 300 MB/s. However, RSA, relying on complex mathematical operations, operates significantly slower than symmetric algorithms, at 4MB/s. These variations highlight trade-offs between speed and security in cryptographic operations.

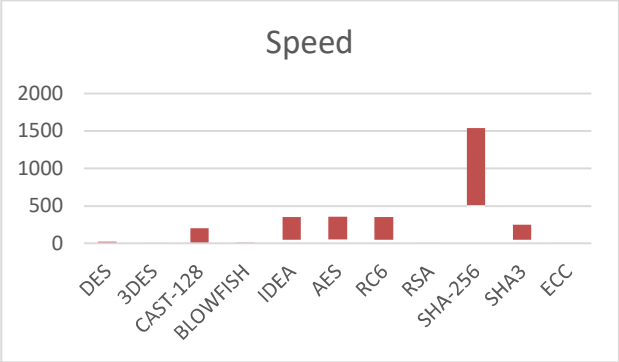


Fig. 15 Encryption Speed of Algorithms

SHA256 (Secure Hash Algorithm 256-bit) is extensively used in blockchain technology, where it serves as the backbone for hashing blocks of transactions. In blockchain, each block contains a header, which includes various pieces of information like the previous block's hash, a timestamp, and a nonce (a random number). Miners compete to find a valid nonce that, when combined with other block data, produces a hash value below a certain target, satisfying the difficulty requirement of the network.

TABLE II. COMPARISON OF DIFFERENT SHA ALGORITHMS

Function	Output Size	Security Strength in Bits		
		Collision	Preimage	2nd Preimage
SHA256	256	128	256	256 - log2(len(M)/B)
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512

SHA3 (Secure Hash Algorithm 3) is a newer hash function compared to SHA256. While not as widely adopted in blockchain as SHA256, it offers similar functionality in terms of generating secure hash values. However, SHA3 uses a different underlying algorithm (KECCAK) compared to SHA256, which may lead to differences in performance and security properties.

IV. SHA-256 Vs SHA-3 Vs ECC

SHA-256, SHA-3, and ECC (Elliptic Curve Cryptography) represent distinct cryptographic algorithms with unique characteristics that cater to different security requirements, performance considerations, and application scenarios.

SHA-256, a member of the SHA-2 family, generates a fixed-size 256-bit hash value from input data of arbitrary size. It's widely employed for data integrity verification, digital signatures, and password hashing. SHA-256 is known for its computational efficiency and robust resistance against known cryptographic attacks, although it's susceptible to length extension attacks.

In contrast, SHA-3, the latest addition to the Secure Hash Algorithm family standardized by NIST, utilizes the Keccak sponge construction. Offering various output lengths (e.g., 224, 256, 384, 512 bits), SHA-3 provides enhanced security compared to its predecessors. It's designed to resist cryptanalytic attacks, including collision and preimage attacks, while maintaining computational efficiency and memory usage similar to SHA-256.

ECC, based on the algebraic structure of elliptic curves over finite fields, offers smaller key sizes and comparable security to traditional asymmetric algorithms like RSA. It's particularly suitable for resource-constrained environments due to its efficient key generation, encryption, and digital signature operations. ECC excels in computational efficiency, especially in scenarios requiring shorter key lengths, making it ideal for mobile devices and constrained environments where computational resources are limited.

In terms of security, ECC provides strong security with smaller key sizes compared to SHA-256 and SHA-3, making it less vulnerable to brute-force attacks. However, SHA-256 and SHA-3 are designed to offer high levels of cryptographic security against known attacks when used properly. Regarding performance, SHA-256 and SHA-3 are generally faster in hash computation compared to ECC, especially for larger data sizes. ECC, on the other hand, excels in key generation and cryptographic operations, offering efficient performance in resource-constrained environments. Additionally, ECC's shorter key lengths contribute to its efficiency, reducing bandwidth and storage requirements compared to SHA-256 and SHA-3. However, SHA-256 and SHA-3 are favoured for data integrity verification, digital signatures, and other cryptographic applications where hash functions are essential.

In blockchain technology, a nonce (number used once) is a crucial component of the mining process, particularly in proof-of-work (PoW) consensus algorithms like those used in Bitcoin. The nonce is a 32-bit (or larger) integer value added to the block header during mining, and its purpose is to generate a hash value that meets a specific target difficulty level. Miners continuously adjust the nonce value and recompute the block hash until they find a nonce that, when combined with the rest of the block header data, produces a hash value below the target threshold.

The importance of the nonce lies in its role in securing the blockchain network. By requiring miners to expend computational effort (in the form of hashing power) to find a valid nonce, PoW consensus algorithms deter malicious actors from attempting to alter the blockchain's transaction history. This process ensures the integrity and immutability of the blockchain, as altering any block would require recalculating the nonce for that block and all subsequent blocks, an exceedingly difficult and resource-intensive task.

Now, let's discuss the computational requirements and performance of the SHA-256, SHA-3 and ECC hashing algorithms in the context of finding a nonce:

A. SHA-256:

- Hash Rate: Typically measured in hashes per second (H/s) or terahashes per second (TH/s) for high-performance mining rigs.
- For example, ASIC miners like Antminer S9 can achieve a hash rate of approximately 14 TH/s, while more advanced models like Antminer S17 can reach around 70 TH/s.
- Time to Find a Nonce: The time required to find a valid nonce varies based on the mining difficulty and hash rate. It could range from minutes to hours or longer depending on network conditions and hardware efficiency.

B. SHA-3:

- Hash Rate: Like SHA-256, the hash rate for SHA-3 would also be measured in hashes per second (H/s) or similar units.
- Specific hash rates for SHA-3 in the context of mining are less commonly available due to its limited adoption in blockchain mining compared to SHA-256.
- Time to Find a Nonce: Similar to SHA-256, the time to find a nonce with SHA-3 would depend on factors like mining difficulty, hash rate, and hardware efficiency.

C. ECC:

- Computational Requirements: ECC involves solving elliptic curve discrete logarithm problems, which can be computationally intensive.
- While specific hash rates may not be directly applicable to ECC, the computational requirements for nonce generation would depend on factors such as the size of the elliptic curve and the specific implementation.
- In protocols like proof-of-stake where ECC is commonly used for digital signatures and key exchange, the computational requirements for nonce generation may be lower compared to proof-of-work mining algorithms like SHA-256 and SHA-3.

While SHA-256, SHA-3, and ECC each offer strong cryptographic security and performance advantages, their suitability depends on specific application requirements, resource constraints, and security considerations. RSA, on the other hand, provides robust cryptographic security but tends to have slower performance, particularly for encryption and decryption operations, especially with larger key sizes. Organizations must carefully evaluate these factors to select the most appropriate algorithm for their use case. ECC (Elliptic Curve Cryptography) typically takes longer to find the nonce due to the inherent computational complexity involved in solving elliptic curve discrete logarithm problems. Unlike hash functions like SHA-256 and SHA-3, which primarily involve iterative hashing operations, ECC relies on mathematical properties of elliptic curves, making the nonce discovery process more intricate. As a result, the computational requirements for nonce generation in ECC are generally higher, particularly in proof-of-work systems where ECC-based mining is less prevalent.

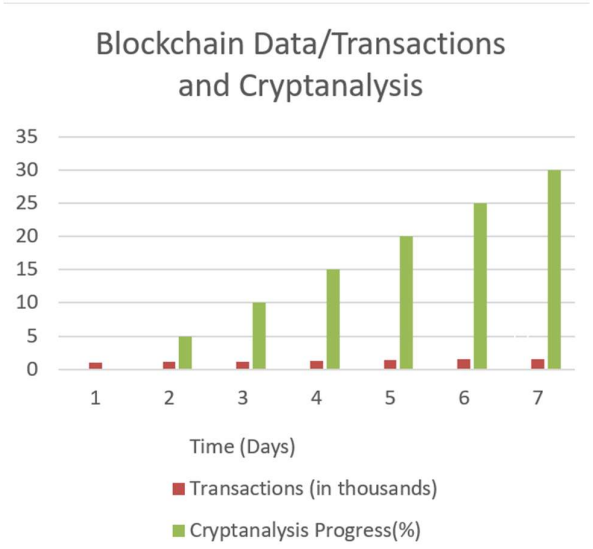


Fig. 16 Transactions and Cryptanalysis Progress for ECC

V. CONCLUSIONS

This paper focuses into a comprehensive examination of both symmetric and asymmetric encryption algorithms, encompassing DES, 3DES, CAST-128, BLOWFISH, IDEA, AES, RC6, RSA, SHA-256, SHA3, and ECC. By scrutinizing their underlying architectures, security attributes, and constraints, it furnishes stakeholders with invaluable insights into their efficacy. The findings underscore that while asymmetric algorithms offer heightened security, they entail increased processing durations and memory requisites. Asymmetric counterparts such as RSA primarily serve for key exchange, whereas symmetric algorithms manage encryption and decryption tasks. Additionally, the discussion extends to implementation nuances, including hardware versus software deployments, and underscores the significance of robust key management practices. Proposing the adoption of hybrid cryptosystems amalgamating diverse algorithms, the study advocates for bolstered security measures to facilitate resilient data transmission amidst the evolving cybersecurity landscape.

REFERENCES

[1] [O.P Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi, "Performance Analysis of Data Encryption Algorithms", IEEE Delhi Technological University India, 2011.
[2] Ketu File white papers, "Symmetric vs Asymmetric Encryption", a division of Midwest Research Corporation.
[3] Tingyuan Nie and Teng Zhang," A Study of DES and Blowfish Encryption Algorithm", IEEE, 2009.
[4] Aamer Nadeem and Dr M. Younus Javed, "A Performance Comparison of Data Encryption Algorithms", IEEE,2005.
[5] Himani Agrawal and Monisha Sharma, "Implementation and analysis of various symmetric cryptosystems", Indian Journal of Science and Technology Vol. 3 No. 12, December 2010.
[6] Diaa Salama, Abdul Minaam, Hatem M. Abdual-Kader, and Mohiy Mohamed Hadhoud, "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types", International Journal of Network Security, PP.78-87, Sept. 2010.
[7] Comparison of Various Encryption Algorithms for Securing Data Dr. Kiramat ullah, Bibi Ayisha, Farrukh Irfan, Inaam Illahi, Zeeshan Tahir Pakistan Institute of Engineering and Applied Sciences (PIEAS)
[8] T. Jamal, and SA Butt, "Cooperative Cloudlet for Pervasive Networks", in Proc. of Asia Pacific Journal of Multidisciplinary Research, Vol. 5, No. 3, PP. 42-26, Aug 2017.
[9] Implementation of Text Encryption using Elliptic Curve Cryptography Laiphrakpam Dolendro Singh∗ and Khumanthem Manglem Singh National Institute of Technology, Manipur, Imphal East 795 001, India
[10] Research on the Security of Elliptic Curve Cryptography Jiaxu Bao Queen Mary University of London, London, UK,E1 4NS
[11] https://people.cs.rutgers.edu/~pxk/419/hw/a-13.html
[12] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Where Is Current Research on Blockchain Technology? —A Systematic Review
[13] Blockchain technology in the energy sector: A systematic review of challenges and opportunities Merlinda Andonia, Valentin Robua David Flynna, Simone Abramb, Dale Geachc, David Jenkinsd, Peter McCallumd, Andrew Peacockd
[14] Dmitry Efanov and Pavel Roschin, The All-Pervasiveness of the Blockchain Technology
[15] Boucher, Philip. "What if blockchain technology revolutionised voting." Unpublished manuscript, European Parliament (2016).
[16] Iansiti, Marco, and Karim R. Lakhani. "The truth about blockchain." Harvard Business Review 95.1 (2017): 118-127.
[17] Pilkington, Marc. "Blockchain technology: principles and applications. Research handbook on digital transformations, edited by f. xavier olleros and majlinda zhegu." (2016)
[18] Nofer, Michael, et al. "Blockchain." Business & Information Systems Engineering 59.3 (2017): 183-187.