Cover Page

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY EDUCATIONAL RESEARCH
ISSN:2277-7881(Print); IMPACT FACTOR :9.014(2025); IC VALUE:5.16; ISI VALUE:2.286
PEER REVIEWED AND REFEREED INTERNATIONAL JOURNAL
(Fulfilled Suggests Parametres of UGC by IJMER)
Volume:14, Issue:7(2), July, 2025
Scopus Review ID: A2B96D3ACF3FEA2A
Article Received: Reviewed : Accepted
Publisher: Sucharitha Publication, India
Online Copy of Article Publication Available : www.ijmer.in

SECURE DATA MODEL WITH AES & SHA ANALYSIS

[1]Shruthi MG and [2]Dr. Indumathi Karthikeyan
[1]Research Scholar (Part-Time), Dr. Ambedkar Institute of Technology, Affiliated to VTU, Bengaluru, Karnataka , India and Assistant Professor, RV Institute of Technology and Management, Bengaluru, Karnataka
[2]Research Head & Associate Professor, Dr. Ambedkar Institute of Technology, Affiliated to VTU, Bengaluru, Karnataka

ABSTRACT:

In today’s fast-moving world where everything is digitalised, ensuring the security of digitized data has emerged as a critical challenge, necessitating robust cryptographic solutions to protect sensitive information from unauthorized access and tampering. This research paper proposes

- A framework to secure data handling mechanisms by integrating the Advanced Encryption Standard (AES) with Secure Hash Algorithms (SHA) on a blockchain platform where we are storing citizen ID cards like Aadhaar Card, PAN Card, Driving License and Passport.
- The encrypted data and its hashed counterpart are immutably stored, ensuring end-to-end confidentiality, data integrity and tamper-resistance during transmission and storage.
- To further strengthen the framework, a comparative analysis of various Secure Hash Algorithms SHA-1, SHA-256, SHA-512 and SHA-3 is implemented on the same data to secure or to identify the most effective hashing mechanism for ensuring data integrity and authenticity. The evaluation considers parameters such as hash length, processing time and collision resistance. By integrating symmetric encryption with optimized hashing, the framework offers a balanced approach to secure data handling suitable for various applications including cloud storage, digital communications and identity management systems.
- The experimental results validate the efficiency and reliability of the hybrid model, emphasizing its potential for real-world deployment with respect to its collision resistance, speed and security. Through this implementation the framework proves efficient algorithms for further cloud storage, secure transmission and blended encryption.

Keywords: Data Security, AES Encryption, SHA Hashing Algorithms, Blockchain Technology and Digital Identity Protection

1. INTRODUCTION:

In today’s digitally connected world, the storage and exchange of sensitive personal information from national ID cards to communication records has become more frequent, distributed and complex. While this digital transformation has revolutionized accessibility and convenience, it has simultaneously heightened concerns around data breaches, unauthorized access and malicious tampering. Ensuring the confidentiality, integrity and authenticity of digital assets is therefore a cornerstone of secure information systems.

- To address these challenges, this research work proposes a hybrid cryptographic framework that combines Advanced Encryption Standard (AES) for data confidentiality and Secure Hash Algorithms (SHA) for integrity verification [1].
- Implementing these cryptographic mechanisms over a blockchain-based infrastructure
- ensures data immutability and traceability, making the system tamper-evident and trust-oriented. Sensitive identity documents like Aadhaar, PAN, driving licenses, and passports are encrypted and hashed before being stored in a decentralized ledger, ensuring they remain unaltered and secure even in shared or cloud-based environments.

- Hashing plays a crucial role in this framework by producing a unique digest of each data entry, allowing for quick detection of unauthorized changes [11].
- A comparative assessment of SHA variants—SHA-1, SHA-256, SHA-512, and SHA-3 is conducted, focusing on digest size, processing speed and collision resistance. [2] As observed in prior work, SHA-3 with its Keccak-based architecture offers enhanced resistance to cryptanalytic attacks and is suitable for modern cryptographic demands.

By combining symmetric encryption with robust hash algorithms on a blockchain layer, this research demonstrates a comprehensive and resilient approach to secure digital data handling, aiming to support scalable, real-world identity management systems.

2. BLOCKCHAIN TECHNOLOGY AND SECURE HASH ALGORITHM WITH CRYPTOGRAPHIC HASH FUNCTION

Blockchain is a decentralized network where there is no central authority or single server that manages the entire system. Instead, every participant or a node plays an equal role in storing, validating and sharing data. This architecture eliminates the common weakness found in centralized systems that is a single point of failure. However, with every node acting independently and data moving freely across the network, the risk of security breaches and data tampering increases significantly.

To tackle these challenges and maintain the integrity and confidentiality of the data being shared, a strong and efficient cryptographic foundation is essential. This is where Advanced Encryption Standard (AES) comes into picture.

AES is a symmetric encryption algorithm that has the same key that is used both for encrypting and decrypting information. It works by dividing data into fixed blocks of 128 bits and then transforming each block through a series of operations that include substitution, permutation, and mixing of bits. Depending on the level of security requirement AES allows for 128-bit, 192-bit or 256-bit keys with longer keys offering stronger protection against brute-force attacks.

AES is widely used across many fields like secure communications, financial systems, government data protection and cloud storage for its efficiency and resilience against modern cryptographic attacks. It strikes a practical balance between security and performance, making it suitable for both high-powered servers and low-resource devices like smartphones and embedded systems.

In essence, AES remains a cornerstone of modern encryption, offering a reliable way to ensure that sensitive data stays confidential and protected from unauthorized access. By providing a reliable and efficient encryption method that plays a central role in modern cryptographic frameworks. Whether it's protecting personal data, financial transactions or government communications, AES remains a cornerstone of digital security due to its robustness and proven track record.

2.1 Role of SHA Hash Functions in Ensuring Data Integrity and Authenticity:

Secure Hash Algorithm (SHA-1): The SHA-1 is a cryptographic function designed to take an input of any length and produce a fixed-size output of 160-bit hash value. The output is commonly referred to as a message digest, which acts as a unique digital ID of the original data. The primary goal of SHA-1 is not to encrypt data for confidentiality, but to verify its integrity by ensuring that the content has not been altered in transit or storage process.

The process begins by preprocessing, where the original ID is first converted into binary form and padded. This padding ensures that the final length of the message is 160 bits, which is the block size used by the algorithm. An important detail in this step is that the original length of the message is also embedded at the end of the padded message, helping to preserve integrity.

Once the data or ID number is ready then the SHA-1 algorithm breaks it down into 160-bit blocks and each block goes through a series of operations involving bitwise logical functions, modular addition and left-rotations. SHA-1 maintains five internal 32-bit variables (A, B, C, D, and E) that are updated across 80 rounds of processing. During each round, a specific function and constant are applied and the message block is mixed thoroughly with these values.

At the end of this iterative process, the final hash value is formed by combining the updated values of A through E. The result is a 160-bit hash that uniquely represents the original input. Even the slightest change in the original data such as flipping of a single bit also will produce a completely different hash output, which makes SHA-1 useful for detecting tampering or accidental corruption.

While SHA-1 was widely used in the past, modern cryptographic standards have gradually moved toward stronger alternatives like SHA-256 and SHA-3 due to vulnerabilities discovered in SHA-1. Nonetheless, understanding SHA-1's mechanics remains important for appreciating the evolution of hashing techniques and the foundation of data integrity practices in digital systems.

Fig 2.1: The Flow representation of SHA-1 is as below:

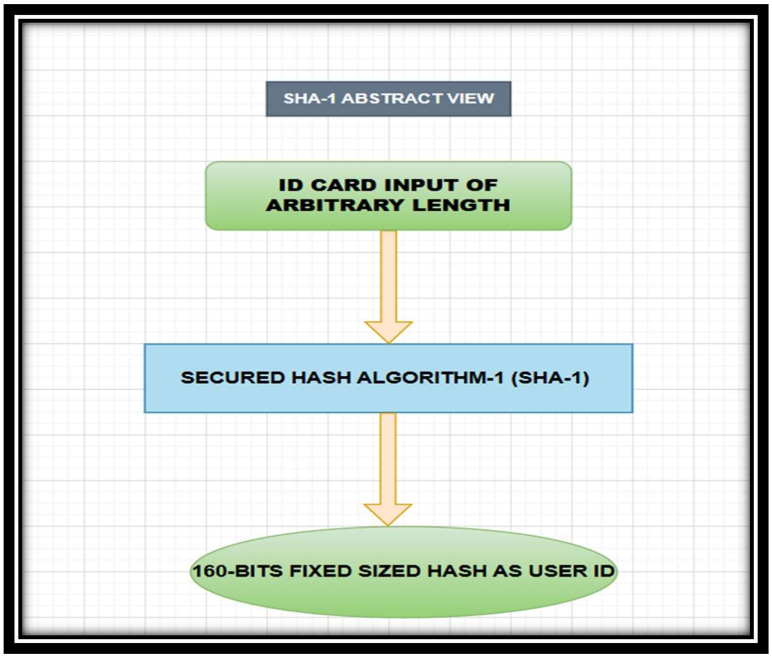

SHA-1 ABSTRACT VIEW
ID CARD INPUT OF ARBITRARY LENGTH
SECURED HASH ALGORITHM-1 (SHA-1)
160-BITS FIXED SIZED HASH AS USER ID

Fig 2.1: Flow Representation of Abstract View of SHA-1

Secure Hash Algorithm (SHA-256): SHA-256 is a member of the SHA-2 family of cryptographic hash functions and it is designed to convert input data of any length into a fixed-size 256-bit (32-byte) hash value. The output often referred to as a message digest, serves as a unique fingerprint for the original input. Unlike encryption, the purpose of SHA-256 is not to conceal data but to ensure its integrity by producing a hash that changes significantly even if the input is altered slightly.

The process begins with preprocessing, where the original input such as an ID number or document it is first converted into binary form. Then, padding is applied to make the total length of the message a multiple of 512 bits, which is the block size used by SHA-256. This padding includes a ‘1’ bit followed by a series of zeros, and finally the original message length is appended at the end in 64-bit binary form. This step ensures the structure needed for secure and consistent hashing.

Once preprocessing is completed, the padded message is divided into 512-bit blocks. Each block undergoes an intensive computation cycle. SHA-256 uses eight working variables (A through H), each initialized with a fixed 32-bit constant. For every 512-bit block, the algorithm performs 64 rounds of operations. These operations include modular additions, logical functions like AND, XOR, and NOT, and circular bit shifts of all carefully designed to mix the input in a nonlinear way that is extremely difficult to reverse.

In each round, a specific constant and a portion of the message schedule are mixed into the current state of variables. The internal state is continually updated in a chain-like process that diffuses the input bits throughout the structure. After all blocks have been processed, the final values of the eight working variables are concatenated to produce the final 256-bit hash output.

Thus, the resulting hash is highly sensitive to input changes even a one-bit modification in the original data will lead to a completely different hash. This property makes SHA-256 an essential tool for verifying data integrity, detecting tampering and securing digital communications.

Due to its robustness and resistance to known attacks, SHA-256 is widely adopted in modern security protocols, including blockchain systems, digital signatures, and certificate verification. It is currently regarded as a reliable and secure hashing method for both academic and commercial applications.

Fig 2.2: The Flow representation of SHA-256 is as below:

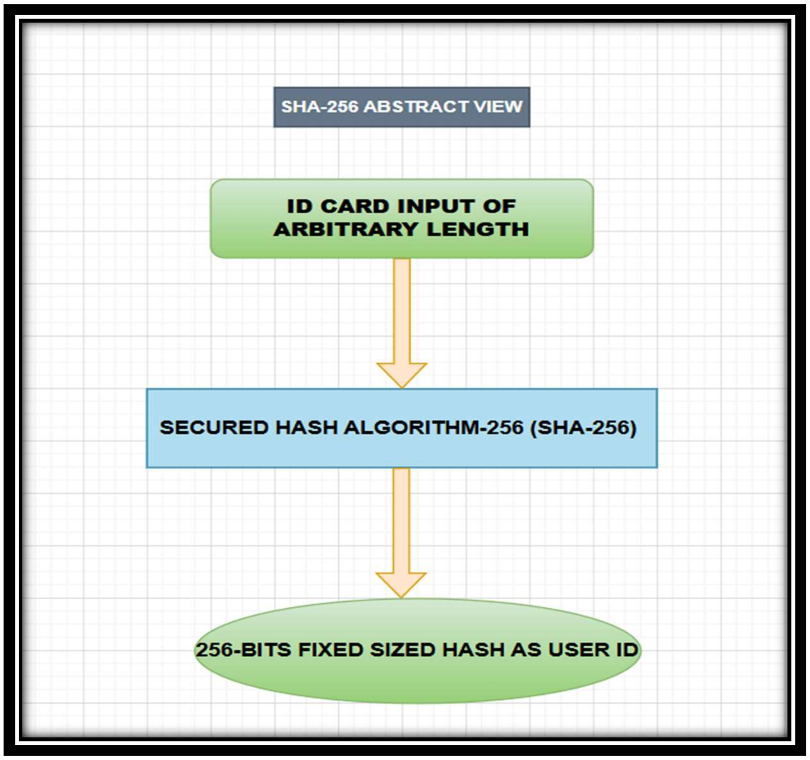

SHA-256 ABSTRACT VIEW
ID CARD INPUT OF ARBITRARY LENGTH
SECURED HASH ALGORITHM-256 (SHA-256)
256-BITS FIXED SIZED HASH AS USER ID

Fig 2.2: Flow Representation of Abstract View of SHA-256

Secure Hash Algorithm-512 (SHA-512): SHA-512 belongs to the SHA-2 family of cryptographic hash functions and it is designed to generate a fixed sized 512-bit hash value from input data of any length. The output commonly referred to as a message digest which acts as a unique digital signature for the input, enabling verification of data integrity. Unlike encryption, SHA-512 does not hide information but confirms whether it has been altered.

The process starts with preprocessing, where the input data such as a document, ID, or digital message—is converted into binary and then padded. This padding ensures that the total message length is a multiple of 1024 bits, which is the block size used by SHA-512. The padding includes a '1' bit, followed by zeros and ends with the original message length encoded in 128 bits. This structure ensures consistency and security during processing.

After preprocessing, the message is broken down into 1024-bit chunks. Each chunk is processed through 80 rounds of operations involving logical functions, modular additions, and bitwise rotations. SHA-512 uses eight 64-bit working variables (labelled A through H), which are initialized with specific constants. These variables are continuously updated in each round using predefined constants and message schedule values derived from the input block.

In each round, a combination of functions and various shift and rotate operations are used to scramble the data. These steps ensure that the final output is a complex and irreversible representation of the input. After all chunks are processed, the eight updated variables are combined to form the final 512-bit hash.

The most powerful feature of SHA-512 is its collision resistance and even the smallest change in input data such as flipping a single bit will result in a vastly different hash output. This makes it highly suitable for detecting tampering, accidental errors or unauthorized modifications.

Due to its high level of security and longer hash length, SHA-512 is often used in environments where a higher degree of protection is required, such as in military-grade applications, secure password storage, cryptographic key management and blockchain systems. Though it requires more computational power and memory than SHA-256, it provides stronger resistance to certain types of attacks.

Finally, SHA-512 stands as a trusted and robust cryptographic tool for safeguarding the integrity of digital data in today's complex and interconnected world**.**
Fig 2.3: The Flow representation of SHA-512 is as below:

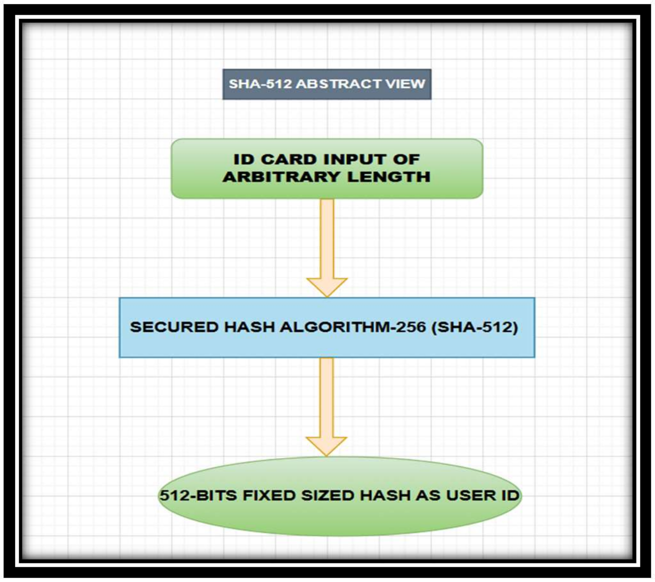

SHA-512 ABSTRACT VIEW
ID CARD INPUT OF ARBITRARY LENGTH
SECURED HASH ALGORITHM-256 (SHA-512)
512-BITS FIXED SIZED HASH AS USER ID

Fig 2.3: Flow Representation of Abstract View of SHA-512

Secure Hash Algorithm (SHA-3): SHA-3 is the latest member of the Secure Hash Algorithm family which is designed as a cryptographic alternative to the earlier SHA-2 series. SHA-3 is based on a completely different approach known as the Keccak algorithm. This fundamental difference gives SHA-3 unique characteristics and makes it highly resistant to known cryptographic attacks.

The purpose of SHA-3 remains the same to take input data of any size and produce a fixed-length hash commonly 224, 256, 384 or 512 bits that uniquely represents the original message. Rather than encrypting data, SHA-3 ensures data integrity, verifying that information hasn't been modified during transmission or storage.

SHA-3 starts by absorbing the input, which involves breaking the message into small parts and processing them into a sponge-like structure. This structure consists of a large internal state, typically 1600 bits, which is divided into two parts that is the rate and the capacity. The rate determines how much of the message is processed at a time while the capacity determines the algorithm's resistance to collisions and other attacks.

The input message is padded and then fed into this sponge structure. Through multiple rounds (typically 24), the algorithm mixes the bits using bitwise operations including XOR, rotation, and permutation. Unlike SHA-2, which has a fixed number of steps per round, SHA-3 performs its mixing in a highly non-linear and parallel-friendly manner, making it efficient and secure.

After the input is fully absorbed and the internal state is thoroughly mixed, SHA-3 enters the squeezing phase, where the final hash value is extracted bit by bit from the sponge. The amount of output depends on the desired hash size.

One of SHA-3's most important features is its independence from SHA-2. Even if future weaknesses are found in SHA-2, SHA-3 remains unaffected because it uses a completely different structure. This makes it an excellent backup standard and a reliable choice for next-generation secure applications, especially in scenarios involving lightweight devices, IoT security, and high-assurance cryptography.

In essence, SHA-3 is not a replacement for SHA-2, but a complementary alternative offering a fresh and modern approach to cryptographic hashing. Its flexible design and strong theoretical foundation make it a trusted option for ensuring data authenticity and tamper-proof verification in the digital age.
Fig 2.4: The Flow representation of SHA-3 is as below:

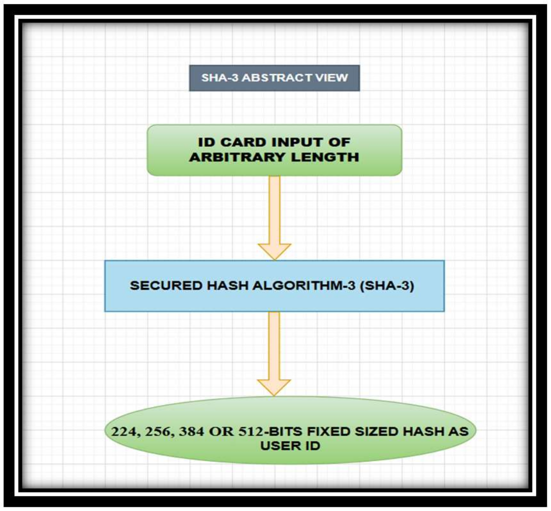

SHA-3 ABSTRACT VIEW
ID CARD INPUT OF ARBITRARY LENGTH
SECURED HASH ALGORITHM-3 (SHA-3)
224, 256, 384 OR 512-BITS FIXED SIZED HASH AS USER ID

Fig 2.4: Flow Representation of Abstract View of SHA-3

3. METHODOLOGY

Fig 3.1 The Framework Encrypting Using Different Variants of Encryption describes to create secure and unique digital identities using personal identification documents. It begins with a user-friendly interface that accepts four ID inputs that is Passport, PAN Card, Aadhaar Card and Driving License. After validating these inputs, the system processes the data through a hashing mechanism by applying the four different hashing algorithms that is SHA-1, SHA-256, SHA-512 and SHA-3. Each of these generates a distinct, fixed-size output that serves as a digital user ID. These hashed outputs ensure that the original details remain private while providing a tamper-proof way to verify identity. Using multiple SHA algorithms not only strengthens security but also prepares the system for future adaptability. This method ensures that sensitive identity information is transformed into secure digital representations without ever revealing the actual data.

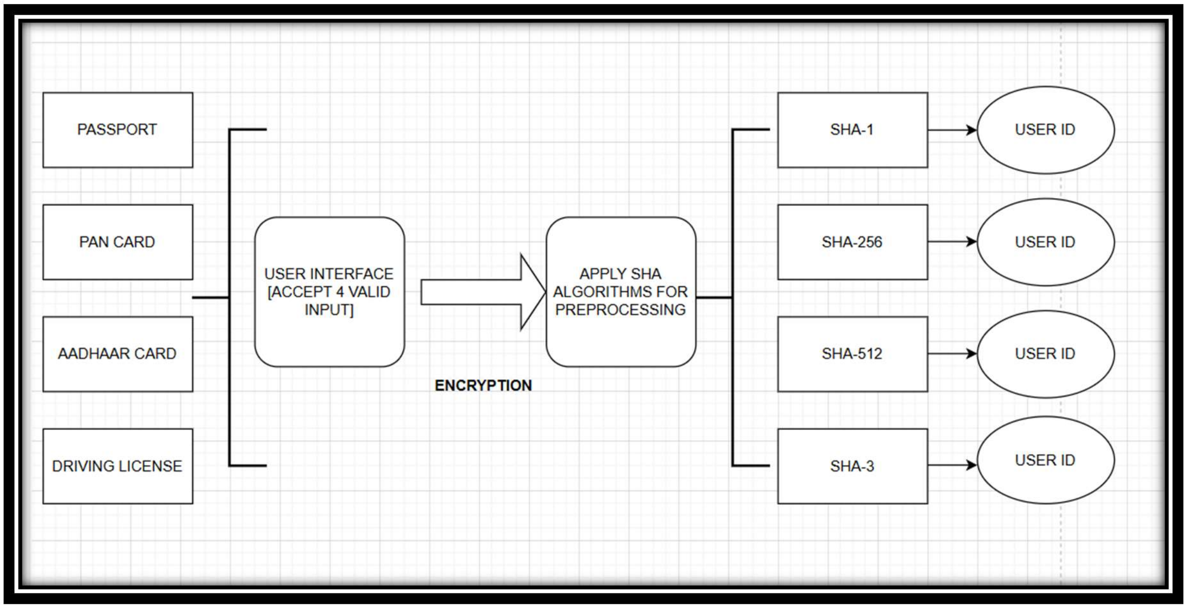


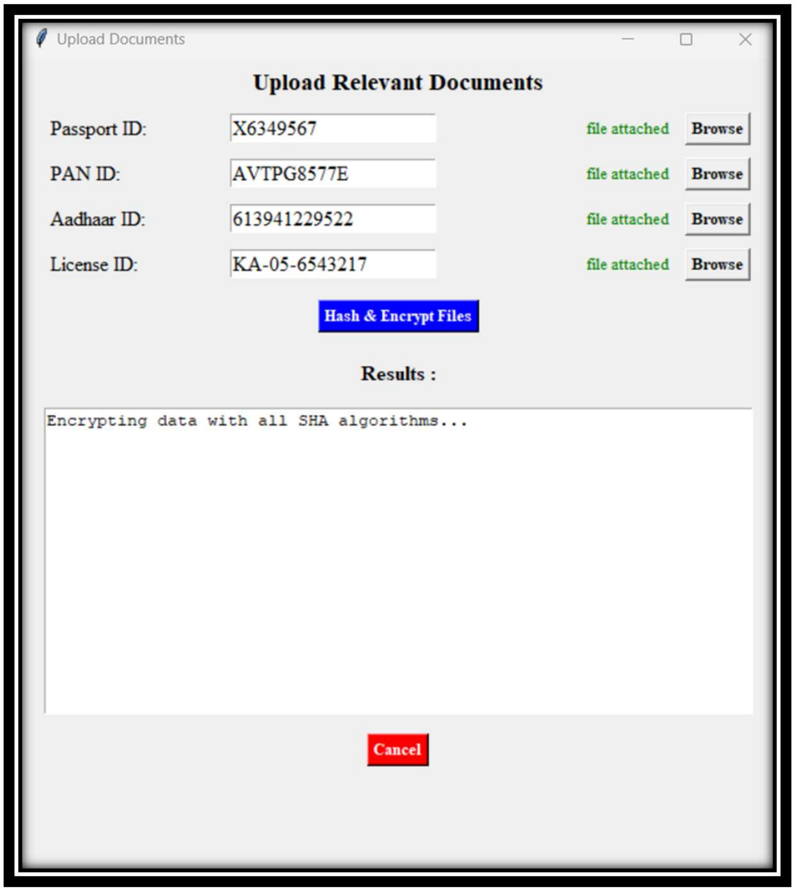
Fig 3.1 The Framework Encrypting Using Different Variants of Encryption

The first stage of implementation initially focuses on gathering and validating user input through a clear and structured interface. As shown in Fig 3.2 in the displayed application window, users are prompted to upload four essential identity documents that is Passport ID, PAN ID, Aadhaar ID and Driving License ID. Each of these fields accepts a specific type of input and validation is performed to ensure that the data provided is both correct and complete.

Once the user enters the values, the system checks for proper formats:

- The Passport ID is validated for alphanumeric structure.
- The PAN ID must follow the standard 10-character Indian PAN format.
- The Aadhaar ID is checked to confirm it contains exactly 12 numeric digits.
- The Driving License ID is validated for correct formatting, which includes both letters and numbers, as per the transport department standards.

Beside each field, there's an indicator confirming whether a file has been attached to ensure the required document is actually uploaded.



Upload Relevant Documents

Passport ID: X6349567 file attached

PAN ID: AVTPG8577E file attached

Aadhaar ID: 613941229522 file attached

License ID: KA-05-6543217 file attached

Results :

```
Encrypting data with all SHA algorithms...
```

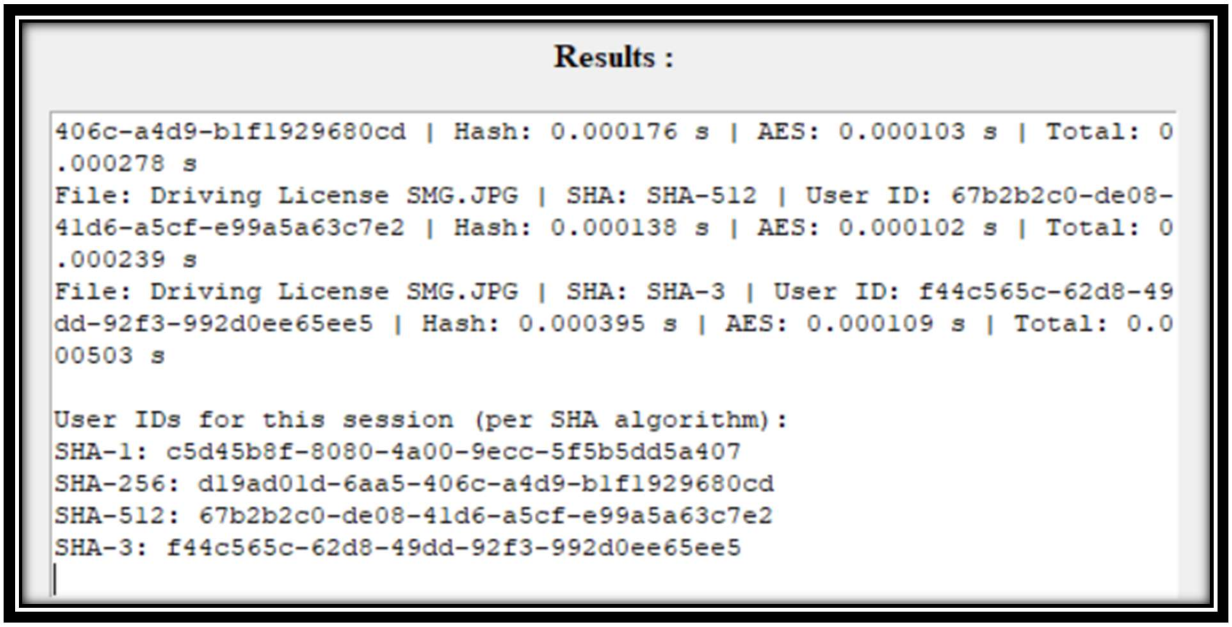
Fig 3.2 User Input and Validation of Documents

Once the user inputs have been validated and submitted, the system begins the core cryptographic processing. Table 1 shows the stage involved in applying multiple hashing algorithms along with AES encryption to generate secure and tamper-proof user IDs.

As shown in the Table 1, each uploaded file in this case is processed through multiple SHA algorithms that is SHA-1, SHA-256, SHA-512 and SHA-3. Each algorithm generates a unique hash which is used as a user-specific ID. These hash values ensure that even if the original document remains the same, each algorithm independently produces a different digital user ID to reinforcing data integrity.

Alongside hashing, Advanced Encryption Standard is used for additional encryption support. The result includes processing times for both hashing and AES, highlighting the system's efficiency.

Table 1: Hashing, AES Integration and User ID Generation



Results :

```
406c-a4d9-b1f1929680cd | Hash: 0.000176 s | AES: 0.000103 s | Total: 0
.000278 s
File: Driving License SMG.JPG | SHA: SHA-512 | User ID: 67b2b2c0-de08-
41d6-a5cf-e99a5a63c7e2 | Hash: 0.000138 s | AES: 0.000102 s | Total: 0
.000239 s
File: Driving License SMG.JPG | SHA: SHA-3 | User ID: f44c565c-62d8-49
dd-92f3-992d0ee65ee5 | Hash: 0.000395 s | AES: 0.000109 s | Total: 0.0
00503 s

User IDs for this session (per SHA algorithm):
SHA-1: c5d45b8f-8080-4a00-9ecc-5f5b5dd5a407
SHA-256: d19ad01d-6aa5-406c-a4d9-b1f1929680cd
SHA-512: 67b2b2c0-de08-41d6-a5cf-e99a5a63c7e2
SHA-3: f44c565c-62d8-49dd-92f3-992d0ee65ee5
```

Table 1: Hashing, AES Integration and User ID Generation

At the end of the process, the system neatly displays the User IDs generated by each SHA algorithm. These IDs act as anonymous digital representations of the uploaded document, which means the actual data is no longer needed to verify identity only the hash matters. This greatly enhances both privacy and security.

Overall, it reflects the heart of the system by converting real-world identity documents into secure, irreversible and unique identifiers using a combination of trusted cryptographic standards. It's not just about securing data, but about building digital trust.

4. RESULTS AND DISCUSSION

Comparative Analysis and Evaluation of SHA-1, SHA-256, SHA-512, and SHA-3 for Secure Identity Hashing:

The system recorded the execution time for each major operation that is hashing time, AES encryption time, and the total time taken for every uploaded file and each applied SHA algorithm. These time metrics provide a clear picture of how efficiently each algorithm performs in real-world conditions.

As shown in the Figure 4.1 Distribution of Hash time by SHA algorithm, once a document is processed, the system logs the exact time it took to generate the hash then apply AES encryption and complete the full cycle. This breakdown helps in evaluating the speed and responsiveness of each SHA method by making it easier to compare their performance not just in terms of security, but also in terms of processing efficiency.

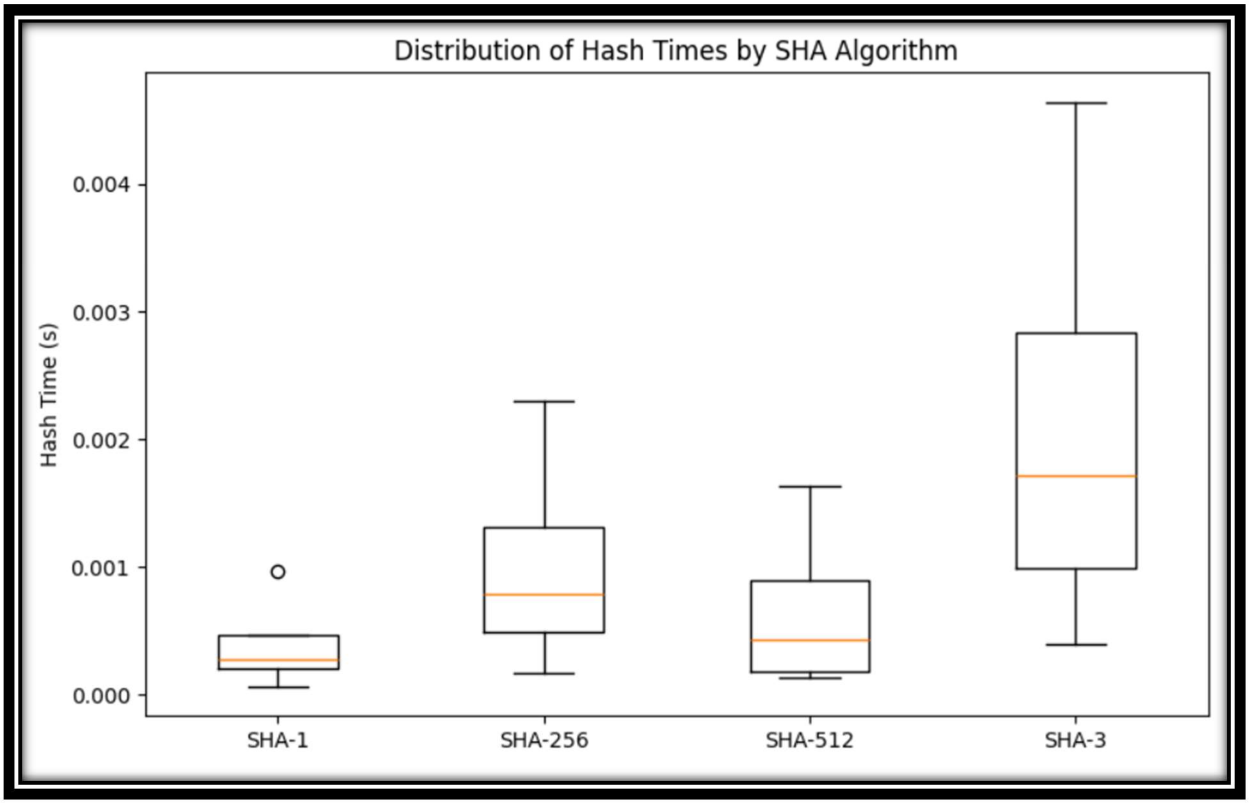


Fig 4.1: Distribution of Hash Times by SHA Algorithm

Table 2: Performance Analysis of Hash Times by SHA Algorithm it captures the minimum, average and maximum time taken for hashing purpose, AES encryption time taken with its values.

SHA-1 consistently performs the fastest in terms of hashing with the lowest average and maximum hash times by making it lightweight but less secure by modern standards.

SHA-256 takes a bit more time than SHA-1 by showing moderate performance and better security.

SHA-512 offers a good balance although slightly slower than SHA-1 it also provides a much stronger hash output.

SHA-3 shows the highest times in all categories, indicating that while it is the most advanced and secure among the listed algorithms, it requires more computational effort and time. But SHA-3 real world implementation is still in process.

Overall, this performance table helps in selecting the right SHA algorithm based on the trade-off between speed and security requirements for a particular implementation.

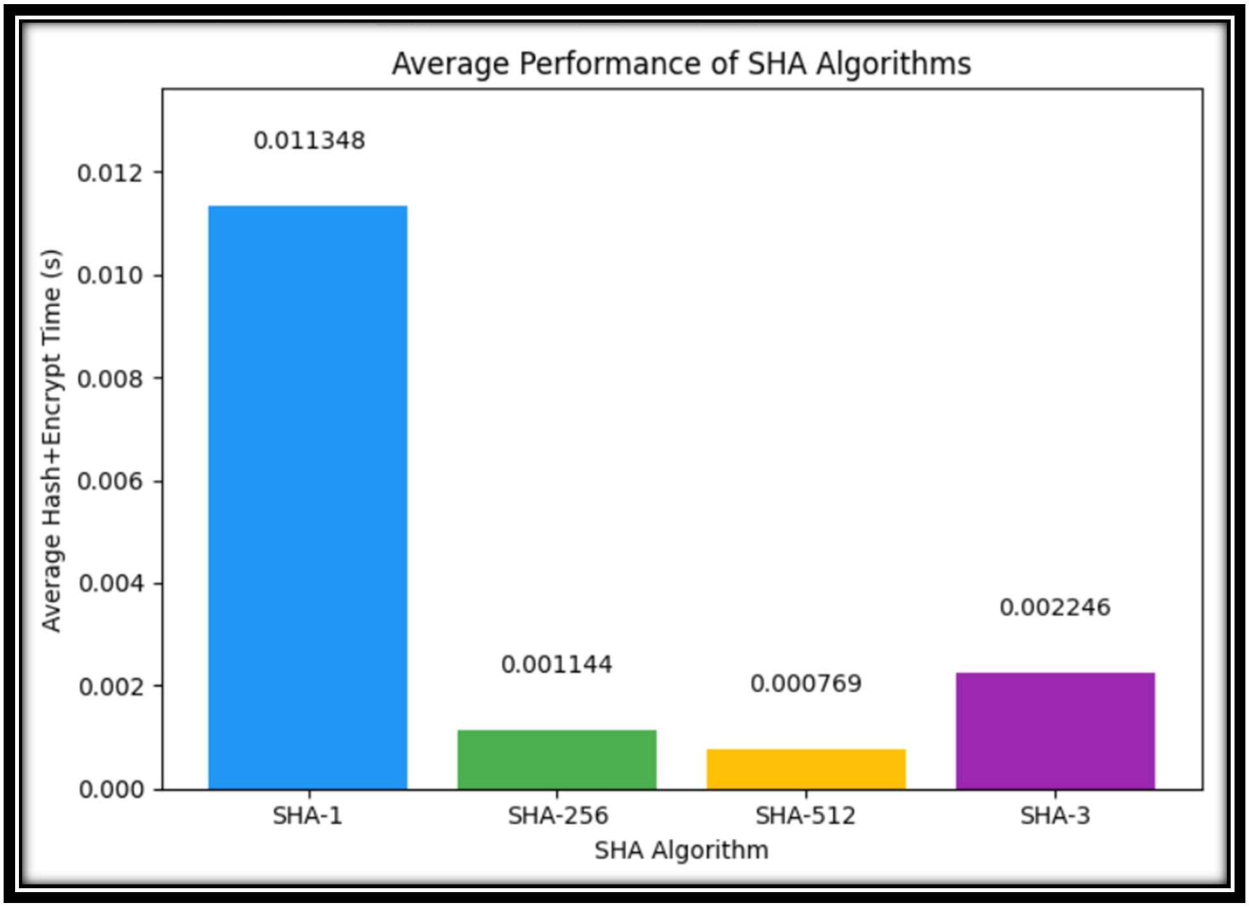
SHA ALGORITHM	HASH MIN	HASH AVG	HASH MAX	AES MIN	AES AVG	AES MAX	TOTAL
SHA-1	0.000064	0.000397	0.000967	0.000104	0.000951	0.043429	0.011348
SHA-256	0.000176	0.001015	0.002296	0.000103	0.000129	0.000161	0.001144
SHA-512	0.000138	0.000656	0.001633	0.000102	0.000113	0.000128	0.000769
SHA-3	0.000395	0.002116	0.004646	0.000109	0.000130	0.000157	0.002246

Table 2: Performance Analysis of Hash Times by SHA Algorithm

Finally, we plot the bar graph, Graph 4.1 Average Performance Analysis of SHA Algorithms with AES Encryption provides a visual comparison of the average total time of hashing + AES encryption taken by four SHA algorithms that is SHA-1, SHA-256, SHA-512 and SHA-3. Each bar represents how long the algorithm takes on average to complete both hashing and encryption tasks.

Here we obtain the result as below:

- SHA-1 stands out to have highest total time (~0.0113 seconds), indicating that despite its reputation for being fast in this case, it consumes more time due to its AES encryption overhead.
- SHA-512 stands the best performance with the lowest combined time (~0.000769 seconds) by suggesting it is both efficient and secure as it is a good balance between speed and strength.
- SHA-256 performs moderately and is most widely used in real world, clocking (~0.001144 seconds), which is reasonable for most practical applications needing better security than SHA-1.
- SHA-3, while more modern and secure, takes slightly more time than SHA-256 and SHA-512 (~0.002246 seconds), which reflects the additional complexity of its hashing mechanism.



Graph 4.1 Average Performance Analysis of SHA Algorithms with AES Encryption

5. CONCLUSION:

This research paper presents a comprehensive framework that integrates AES encryption with multiple SHA hashing algorithms to ensure the secure processing and storage of sensitive identity documents. Each hash value generated from the input files is further encrypted enhancing the security of stored data. These encrypted hashes are systematically logged into a CSV file, creating a reliable and organized method for maintaining secure digital records. Performance monitoring is a key feature of the system where each hashing and encryption operation is timed and recorded by allowing precise measurement of performance per file and per SHA variant. These metrics also includes the hash computation time, AES encryption time and the overall time taken to clearly display the results. To support further insights, the framework includes a dedicated analysis panel which provides a tabular summary of minimum, average and maximum times for all hashing and encryption operations across the SHA algorithms. A plotting visually illustrates the variation in hashing time, while a bar chart offers a direct comparison of the average total processing time for each algorithm. Altogether, this system combines cryptographic strength with practical analysis tools, delivering a secure, transparent and efficient way to manage and evaluate encrypted hash values. It not only ensures the confidentiality and integrity of identity documents but also provides a meaningful assessment of cryptographic performance across different SHA algorithms.

REFERENCES

1. Standard Journal Articles:

1. Khan, M. A., & Salah, A. (2021). Comparative analysis of secured hash algorithms for blockchain technology and Internet of Things. International Journal of Advanced Computer Science and Applications, 12(3), 566–571.
2. Verma, J., Shahrukh, M., Krishna, M., & Goel, R. (2021, December). A critical review on cryptography and hashing algorithm SHA-512. International Research Journal of Modernization in Engineering, Technology and Science, 3(12).
3. Das, A. K. (2012). A secure and efficient user anonymity-preserving scheme for roaming users in global mobility networks. IEEE Transactions on Wireless Communications, 11(1), 58–66.
4. Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. Journal of Computer and System Sciences, 28(2), 270–299.
5. Lamport, L. (1981). Password authentication with insecure communication. Communications of the ACM, 24(11), 770–772.
6. Huynh, H.-T., Tran, T.-K., & Dang, T.-P. (2025, March). Implementing a very high-speed secure hash algorithm 3 accelerator based on PCI Express. International Journal of Research in Engineering and Science, 14(1), 1–11.
7. Politecnico di Torino DET Team. (2023, November). Comparative study of Keccak SHA-3 implementations. Cryptography, 7(4), Article 60.

2. More than Six Authors:

[8] Bhargavan, K., et al. (2017). Implementing TLS with verified cryptographic security. In Proceedings of the IEEE Symposium on Security and Privacy (SP) (pp. 445–462).

3. In Press:

Nil

4. Books and Other Monographs:

[9] Menezes, A., van Oorschot, P., & Vanstone, S. (1996). Handbook of applied cryptography. CRC Press.

5. Conference Papers:

[10] Juels, A., & Brainard, J. (1999, February). Client puzzles: A cryptographic countermeasure against connection depletion attacks. In Proceedings of the Network and Distributed System Security Symposium (NDSS).

[11] Zheng, Y., & Seberry, J. (2001). Practical approaches to attaining security against chosen ciphertext attacks. In Proceedings of the 8th ACM Conference on Computer and Communications Security (pp. 56–67).

[12] Vasuki, M., Karunamurthy, A., & Pothyeswari, G. (2023, July). Enhancing GPU performance for SHA-3 algorithm: Optimizing hashing operations in a parallel computing environment. In Proceedings of IJSREM.

[13] Soni, A., Sahay, S. K., & Mehta, P. (2025). AESHA3: Efficient and secure sub key generation for AES using SHA-3. [Conference details not available].

6. Dissertation:

(None of the sources appear to be a dissertation.)

7. Online Documents / Archives:

[14] Rivest, R. L. (1992). The MD5 message-digest algorithm (RFC 1321). MIT Laboratory for Computer Science. Retrieved from https://www.rfc-editor.org/rfc/rfc1321

[15] Chen, S., Guo, J., List, E., Shi, D., & Zhang, T. (2025). Scrutinizing the security of AES based hashing and one-way functions. Cryptology ePrint Archive, Paper 2025/792. Retrieved from https://eprint.iacr.org/2025/792