International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5679~5689
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5679-5689

Enhancing system integrity with Merkle tree: efficient hybrid cryptography using RSA and AES in hash chain systems

Irza Nur Fauzi[1], Farikhin[2], Ferry Jie[3]
[1]Master of Information Systems, School of Postgraduate, Diponegoro University, Semarang, Indonesia
[2]Department of Mathematics, Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia
[3]School of Business and Law, Edith Cowan University, Joondalup, Australia

Article Info

Article history:

Received May 27, 2025
Revised Jul 17, 2025
Accepted Sep 14, 2025

Keywords:

Advanced encryption standard algorithm
Data integrity
Hash chain
Hybrid cryptography
Merkle tree
Rivest-Shamir-Adleman algorithm

ABSTRACT

An analysis is conducted to address the growing threats of data theft and unauthorized manipulation in digital transactions by integrating \structures within hash chain systems using hybrid cryptography techniques, specifically Rivest-Shamir-Adleman (RSA) and advanced encryption standard (AES) algorithms. This approach leverages AES for efficient symmetric data encryption and RSA for secure key exchanges, while the hash chain framework ensures that each data block is cryptographically linked to its predecessor, reinforcing system integrity. The Merkle tree structure plays a crucial role by allowing precise and rapid detection of unauthorized data changes. Empirical analyses demonstrate notable improvements in both the efficiency of cryptographic processes and the robustness of data validation, underscoring the method's applicability in high data throughput environments such as educational institutions. This research makes a substantive contribution to information security by offering a sophisticated solution that strengthens data protection practices, ensuring greater resilience against increasingly sophisticated data threats.

Corresponding Author:

Irza Nur Fauzi
Master of Information Systems, School of Postgraduate, Diponegoro University
Semarang, Indonesia
Email: irzanurf@gmail.com

1. INTRODUCTION

The rapid advancement of digital technology, particularly in the realms of internet usage and online transactions, has drastically transformed the way data is generated, shared, and stored. This transformation, while providing unparalleled convenience and connectivity, has also introduced significant challenges, notably the escalating risks of data theft, corruption, and unauthorized manipulation [1]. As the volume of sensitive information exchanged and processed online grows, so too does the urgency for developers and data custodians to implement robust security measures that ensure data integrity and confidentiality. This evolving digital landscape necessitates the development of innovative cryptographic techniques to safeguard against increasingly sophisticated threats.

Although various security mechanisms have been proposed in prior research, many systems still rely on manual data recording and lack automated integrity verification, resulting in vulnerabilities such as undetected data tampering and insufficient real-time monitoring [2]. This lack of robust, automated security frameworks poses a critical challenge: how to ensure data fidelity and resilience against unauthorized changes effectively. Previous approaches often failed to integrate scalable cryptographic models that simultaneously address confidentiality and integrity. While Merkle tree structures and hash chain systems have been explored as promising tools for data verification, they are frequently implemented without strong

encryption protocols, limiting their effectiveness. This study builds upon those foundations by integrating Merkle trees with Rivest-Shamir-Adleman (RSA) and advanced encryption standard (AES) algorithms into a hybrid cryptographic framework that ensures secure key management, efficient encryption, and real-time integrity verification—effectively overcoming the limitations of earlier models.

One promising avenue of research involves integrating Merkle tree structures within hash chain systems. Merkle trees provides an efficient method for verifying data integrity by cryptographically linking each data block to its predecessor. This chaining process creates a transparent mechanism for detecting any unauthorized data modification, as even minor alterations in the data set will cause a detectable discrepancy in the hash values [3]. Additionally, combining this structure with established cryptosystems, such as RSA and AES, can create a robust framework that encrypts and verifies data securely, thereby significantly enhancing trust and reliability in digital system [4].

Building on existing literature, which highlights the vulnerabilities associated with conventional data storage and security practices [5], this research focuses on the innovative use of RSA and AES algorithms within hash chain frameworks. By leveraging the strengths of both symmetric (AES) and asymmetric (RSA) encryption methods, the research aims to develop a sophisticated cryptographic solution that offers superior security and privacy. Compared to other combinations such as Twofish and ElGamal, RSA and AES provide a more robust hybrid approach, effectively safeguarding against unauthorized access while maintaining efficient encryption and decryption processes [6].

The proposed solution enhances current practices by ensuring that each data block is cryptographically linked to the previous one, thereby strengthening the system's resistance to unauthorized tampering [7]. This research contributes a novel methodology for data protection, particularly in environments characterized by high data throughput, and lays the groundwork for future innovations in information security systems. As such, it holds significant implications not only for the education sector, where secure data management is crucial but also for any industry reliant on the integrity and confidentiality of digital information.

This study is situated within the domain of computer engineering and information security, addressing the growing need for robust data protection in digital systems. The central research question examines how hybrid cryptographic techniques—specifically, the integration of RSA and AES within a hash chain framework enhanced by Merkle tree structures—can effectively improve system integrity and detect unauthorized data modifications. The main objective is to develop a secure and efficient encryption system that can detect unauthorized data modifications in real time. This approach utilizes the strengths of individual components: AES provides fast encryption [8], RSA ensures secure key management [9], and Merkle trees facilitates integrity verification, collectively resulting in a robust and scalable solution [10].

This work contributes a novel methodology that enhances current cryptographic practices by linking each data block cryptographically and validating its integrity through hierarchical hashing. Compared to existing models, the proposed system demonstrates superior performance in both execution time and anomaly detection, as evidenced by empirical testing.

The findings expand existing knowledge in the field of information security and cryptographic engineering, offering practical insights for implementation in sectors such as education, cloud services, database backend engineering, and IoT. Future research may build upon this model to further optimize cryptographic efficiency and explore its adaptability to emerging technologies and distributed systems. Therefore, this study proposes a hybrid cryptographic model that integrates RSA and AES within a hash chain system enhanced by Merkle tree structures, aiming to ensure real-time data integrity verification and secure encryption in high-throughput digital environments.

2. LITERATURE REVIEW

Integrating advanced cryptographic techniques is critical for improving system integrity and data security. This review examines various studies that focus on using hash chains, Merkle trees, and hybrid cryptography, collectively enhancing data protection in information systems. Nagaraj and Mohanraj investigated one-way hash chain algorithms for selective data encryption, emphasizing their role in maintaining data integrity. While effective, one-way encryption limits decryption capabilities, underscoring the need for two-way hash chains or their combination with complementary strategies, particularly in systems employing Merkle trees for robust verification [11].

Gaur *et al*. [12] conducted a comparative analysis of cryptographic algorithms, identifying AES as particularly efficient regarding memory and execution time. AES's efficiency supports its integration with RSA in hybrid cryptographic models, which are well-suited for hash chain systems leveraging Merkle trees for enhanced integrity checks.

Jintcharadze and Iavich [6] explored hybrid encryption with AES and RSA, highlighting significant security benefits and efficient vital exchanges. The conjunction of these algorithms within a hash chain architecture offers robust protection and speed, with Merkle trees further reinforcing data verification by establishing a secure audit trail.

Sholikhatin *et al.* [13] examined the execution time and avalanche effect of RSA and AES, demonstrating AES's superior performance. Zou *et al.* [14] further noted that the stability of hybrid algorithms for processing large files is crucial for maintaining high integrity within hash chain systems enriched by Merkle tree structures. Both studies highlight that the efficiency and stability of cryptographic algorithms are critical factors in preserving data integrity, especially when applied to complex and large-scale hash chain systems enhanced by Merkle tree structures.

Hash chains combined with Merkle trees provide a strong authentication and data integrity framework. Zhao *et al.* [15] demonstrated the efficacy of hash chain-integrated setups. Han and Jiang [16] proposed a secure communication method based on message hash chains, demonstrating improved message integrity, non-repudiation, and transmission efficiency through chain-based authentication and signature mechanisms. These approaches collectively inform strategies to enhance system integrity through the adept use of hybrid cryptography and Merkle trees.

3. METHOD

Cryptographic algorithms enhance data security by transforming input into secure output through defined steps. This paper examines RSA and AES encryption within a hash chain framework enriched by Merkle trees. RSA, an asymmetric algorithm, secures key exchanges, while AES offers efficient symmetric encryption. Merkle trees reinforce data integrity and verification. Together, these technologies form a robust architecture that strengthens security, ensures data integrity, and delivers efficient cryptographic solutions across applications.

3.1. AES

The advanced encryption standard (AES) is a modern symmetric cryptographic algorithm introduced by NIST in 2001 to replace the data encryption standard (DES) [16]. AES was developed by Joan Daemen and Vincent Rijmen and selected for its resistance to cryptographic attacks, computational efficiency, and design simplicity [17]. AES operates with a block length of 128 bits and key lengths of 128, 192, or 256 bits, with the number of rounds varying based on crucial length [18].

AES employs five data units: bit, byte, word, block, and state. A bit is the smallest data unit, part of the binary system, while a byte consists of 8 bits. A word is 4 bytes (32 bits), and a block is 16 bytes (128 bits). The state is a block arranged as a 4x4 byte matrix [19]. The encryption process involves transforming the state in several rounds, each comprising four transformations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. The number of rounds depends on the key length, and MixColumns is omitted in the final round [19].

Decryption in AES is the reverse of the encryption process, with each transformation having its inverse: InvSubBytes, InvShiftRows, and InvMixColumns. The procedures ensure the secure conversion of encrypted data back to its original form, maintaining data integrity and confidentiality throughout the transformation [20]. This reversal process not only restores the original data but also reinforces the reliability of AES in secure communication systems where data accuracy and protection are paramount.

3.2. RSA

RSA is a widely recognized public-key algorithm, introduced in 1976 by MIT researchers Ron Rivest, Adi Shamir, and Leonard Adleman [21]. Its security is rooted in the complexity of factoring large numbers into prime factors. RSA's computations rely on Euler's theorem, with Euler's totient function, $\Phi(n)$, representing the count of numbers less than n that are coprime to n. For prime numbers, $\Phi(n)$ equals n-1, and if n equals the product of two primes (p and q), then $\Phi(n)$ is (p-1)(q-1) [22].

RSA's encryption and decryption processes hinge on selecting two large random prime numbers, p and q. These primes generate the modulus $n = pq$, which is used in the public key. The RSA key generation involves calculating $\Phi(n)$ and selecting an integer e as the public key, where $1 < e < \Phi(n)$ and e is coprime with $\Phi(n)$. The decryption key, d, is determined by the relationship $ed \equiv 1 \pmod{\Phi(n)}$ [21].

RSA's encryption involves converting the plaintext message into blocks, applying the formula $c \equiv m^e \bmod n$ to transform it into ciphertext. Decryption reverses this using $m \equiv c^d \bmod n$, revealing the plaintext. This process emphasizes secure vital exchanges and ensures data confidentiality, making RSA an integral part of modern cryptographic protocols [23].

3.3. Hash chain

A hash chain is a cryptographic technique for securing data by successfully generating a sequence of related hash values. It protects data transmitted over communication networks by encrypting it using hash values as encryption keys [24]. The hash function H is an algorithm that accepts an input message and a fixed-length key, producing an output message of a fixed length referred to as the digest [4].

3.4. Merkle tree

The Merkle Tree, introduced by Merkle in 1979, is a method for organizing data into blocks, computing hash values for each block, and inserting them into leaf nodes [25]. This process is repeated in a binary tree structure until a single hash value, known as the hash root, is created. The hash root serves as a signature for verifying the integrity of the data blocks. In blockchain technology, the Merkle tree is commonly used to verify the integrity of block data [26].

A Merkle tree is particularly effective at handling large datasets by dividing them into blocks of a specific size and placing them in corresponding leaf nodes. When synchronizing data to another node, the Merkle root is extracted, allowing for a thorough integrity check. Even a single-byte error during data transmission can be quickly identified by comparing Merkle roots, as every data block's hash value is compared with others, ensuring data integrity in blockchain environments [26].

As an essential component of blockchain technology, the Merkle tree enables secure data stability and content verification. It is integral to blockchain architectures, facilitating the secure data exchange between users by ensuring that all data remains consistent and error-free across the network [27]. This mechanism ensures that any tampering or inconsistency within the data blocks can be swiftly identified, making Merkle trees a cornerstone of trust and reliability in decentralized systems.

3.5. Methodological foundation and procedural design

The proposed cryptographic model is designed to enhance data integrity and security by combining symmetric and asymmetric encryption techniques within a hash-based framework. This section outlines the theoretical basis underlying the method selection and explains the procedural steps taken during the research. By leveraging the complementary strengths of AES, RSA, and Merkle tree structures, the model establishes a resilient cryptographic foundation capable of detecting tampering and ensuring secure data transmission across distributed environments.

The model utilizes two foundational cryptographic algorithms: AES and RSA. AES is a symmetric encryption algorithm widely recognized for its high computational efficiency, scalability, and suitability for securing large volumes of data in real-time applications. It operates on fixed-size blocks and supports multiple key lengths, making it suitable for high-speed data encryption [28]. RSA, on the other hand, is an asymmetric encryption algorithm that provides secure key exchange and digital signature capabilities. Its security is based on the mathematical difficulty of prime factorization, which ensures robustness in key management [29].

To further strengthen the integrity verification process, the model incorporates Merkle tree structures into a hash chain system. A Merkle tree is a binary tree of hash values that enables efficient and scalable verification of data blocks. Each leaf node represents the hash of a data block, and internal nodes are derived from the hashes of their child nodes. The root of the tree, known as the Merkle root, serves as a cryptographic summary of the entire dataset [10]. Any modification to a single data block results in a change in the corresponding hash, which propagates through the tree and alters the Merkle root, thereby enabling immediate detection of tampering.

The methodological rationale for this integration lies in the complementary strengths of the components. AES ensures rapid encryption and decryption, RSA secures the transmission of encryption keys, and Merkle trees provide a reliable mechanism for verifying data integrity across distributed systems. The hash chain structure links each data block to its predecessor, creating a sequential dependency that further reinforces the system's resistance to unauthorized alterations.

The research procedure is implemented as follows.

a. Initialization of empty arrays to store encrypted data blocks and Merkle tree leaf hashes.
b. Iterative processing of each data block:
   - Compute the hash of the block.
   - Encrypt the block using AES with a generated symmetric key.
   - Encrypt the AES key using RSA with the public key.
   - Append the encrypted block and encrypted key to the data chain.
   - Generate a new AES key for the next block using the current block's hash.
   - Store the block hash in the Merkle tree leaf array.

c. Construction of the Merkle tree from the collected leaf hashes.
d. Computation of the Merkle root to represent the integrity of the dataset.
e. Generation of verification proofs for each block hash and validation against the Merkle root.
f. Final output includes the Merkle root and the encrypted data chain.

This procedure ensures that the cryptographic operations are both secure and reproducible. By combining AES, RSA, and Merkle tree structures within a hash chain framework, the model leverages the strengths of each component. As a result, it offers a comprehensive solution for maintaining data confidentiality and integrity in digital environments.

4. PROPOSED MODEL

The proposed model employs hybrid cryptography, combining symmetric and asymmetric encryption techniques to bolster data security within a hash chain system, further reinforced by Merkle tree structures. This strategy harnesses the speed of symmetric key encryption alongside the robust security provided by asymmetric key cryptography and Merkle trees. While hybrid cryptography is typically secure, it may encounter challenges like slower processing speeds with large datasets.

As shown in Figure 1, this model combines fast data encryption using AES with secure key exchange through RSA, all within a hash chain structure. Each data block includes a "Next key", which is generated from the hash of the current block and used to encrypt the next one—creating a secure link between blocks. The AES key used to encrypt each data block is securely wrapped using RSA encryption—referred to in the model as "RSA Enc"—which ensures access and decryption are restricted to authorized users. To further enhance the system's reliability, each block is verified through Merkle tree validation, enabling quick and accurate detection of any unauthorized changes to the data. This layered approach yields a cryptographic framework that is not only secure but also resilient and scalable in environments where data integrity is crucial.

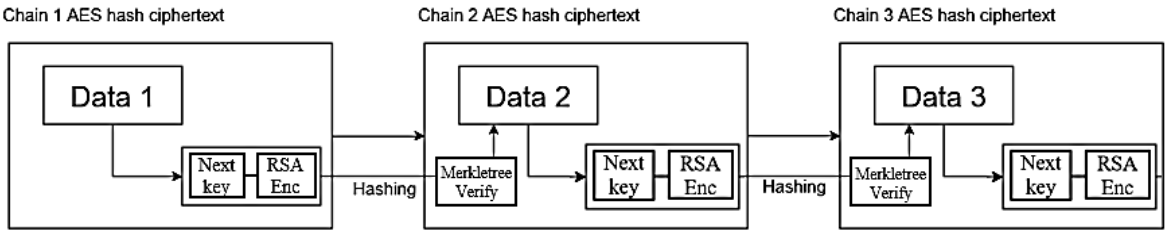


Figure 1. Proposed model of hash chain block

Within this framework, data is encrypted using a symmetric key, which is then secured with the sender's private asymmetric key. The receiver decrypts the symmetric key to access the data. Merkle tree validation ensures integrity by verifying each block and detecting unauthorized changes. This process is detailed in the accompanying pseudocode below.

```
BEGIN
       1. Initialize:
           - EncryptedDataChain[] ← []
           - MerkleLeaves[] ← []
       2. FOR each block in Data[], DO
           3. Compute the hash of the data block:
               BlockHash ← Hash(block)
           4. Encrypt data block with AES:
               EncryptedBlock ← AES_Encrypt(AES_Key, block)
           5. Encrypt AES key with RSA:
               EncryptedAESKey ← RSA_Encrypt(RSA_PublicKey, AES_Key)
           6. Append the encrypted block and encrypted AES key to the chain:
               EncryptedDataChain.append({EncryptedBlock, EncryptedAESKey})
           7. Generate a new AES key for the next block:
               AES_Key ← Hash(BlockHash)
           8. Append BlockHash to MerkleLeaves:
               MerkleLeaves.append(BlockHash)
END FOR
       9. Construct Merkle Tree from MerkleLeaves:
           MerkleRoot ← ConstructMerkleTree(MerkleLeaves)
```

```
    10. Verify Merkle Tree:
        FOR each BlockHash in MerkleLeaves, DO
            VerificationProof ← GenerateProof(MerkleLeaves, BlockHash)
            ASSERT MerkleTree_Verify(MerkleRoot, VerificationProof, BlockHash) == True
        END FOR
    11. Return MerkleRoot and EncryptedDataChain
END
```

The provided pseudocode outlines a secure data encryption and verification process using hybrid cryptography, integrating AES, RSA, and Merkle trees. Initially, it sets up two empty arrays, `EncryptedDataChain` for storing encrypted data and keys and `MerkleLeaves` for storing hash values of each data block. For each data block, the process begins by computing its hash. The data block is then encrypted using AES with a specified key, ensuring data confidentiality. The AES key is further encrypted with RSA, securing the key itself. The encrypted block and the AES key are appended to the `EncryptedDataChain`. A new AES key for the next block is generated by hashing the current block's hash, creating a chain of linked encryption keys.

Following this, the hash of each data block is added to `MerkleLeaves`, which are used to construct a Merkle Tree. The root of this tree, `MerkleRoot`, represents the cryptographic summary of all the blocks, ensuring integrity verification of the entire data set. To verify the integrity of the data, the pseudocode generates a verification proof for each block hash and confirms it against the Merkle root. This confirmation ensures that no unauthorized changes have occurred to the data blocks. Finally, the process returns the Merkle root and the `EncryptedDataChain`, providing data security through encryption and data integrity through the Merkle tree structure.

5. RESULTS AND DISCUSSION

This section presents the outcomes of the comprehensive evaluation of the proposed hybrid cryptography model integrated with Merkle trees. The evaluation is structured across several sub-sections, including preliminary tests, complexity assessments, and execution time analyses. The preliminary tests focus on validating the model's robustness and ability to maintain data integrity under various conditions. Following this, the complexity analysis of the cryptographic algorithms is explored, offering insights into their computational efficiency. Finally, execution times are assessed to determine the model's performance in real-time applications, ensuring both security and efficiency are upheld.

5.1. Integrity testing

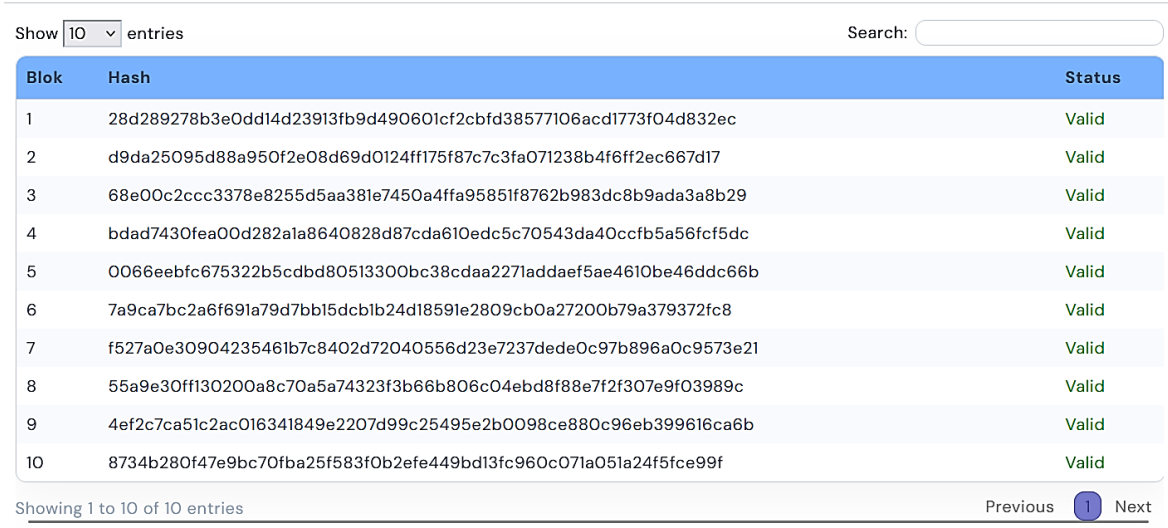
The Merkle tree supports data consistency testing through its hierarchical structure. Each node contains the hash of its child nodes, while the Merkle root serves as the overall integrity marker. This structure is essential for maintaining data integrity in complex and distributed systems, ensuring that information remains intact from origin to destination [3].

This approach ensures immediate detection of changes and offers high verification efficiency, even with large datasets. Initially, a test using entirely valid data confirmed the system's baseline integrity checks. Subsequently, a second test involved altering one of the data blocks to assess the system's ability to detect discrepancies. Based on the results, the system successfully identified inconsistencies, providing outputs that match those illustrated in the figures. This test evaluates the system's reliability in anomaly detection, ensuring consistent and accurate system functions. Analyzing the test results provides insights into the system's effectiveness in preserving data integrity.

Figure 2 demonstrates the cryptographic strength of the Merkle tree, which secures data blocks through hierarchical hashing. This structure allows for precise detection of even minor data modifications. As a result, the integrity of the entire data chain is preserved, protecting it from unauthorized alterations.

Figure 2(a) reflects the validation results of ten data blocks, all deemed "Valid" by the Merkle tree verification process. This consistent status indicates that each block's hash value aligns precisely with the expected values, confirming that the data remains unaltered and secure. The uniformity across all entries demonstrates the system's effectiveness in maintaining data integrity through encryption and verification mechanisms. The Merkle tree plays a critical role here, ensuring that even the slightest data modification would trigger a discrepancy, thereby highlighting the robustness of this cryptographic technique in safeguarding data.

In Figure 2(b), a modification in block 6 triggers a ripple effect, resulting in an "Invalid" status for block six and all subsequent blocks. This illustrates the Merkle Tree's effectiveness in maintaining data integrity, as even a single alteration disrupts the hash verification for all dependent blocks. The hash inconsistency begins with the altered block and propagates through the chain, emphasizing the robustness of this cryptographic method in detecting and preventing unauthorized changes.

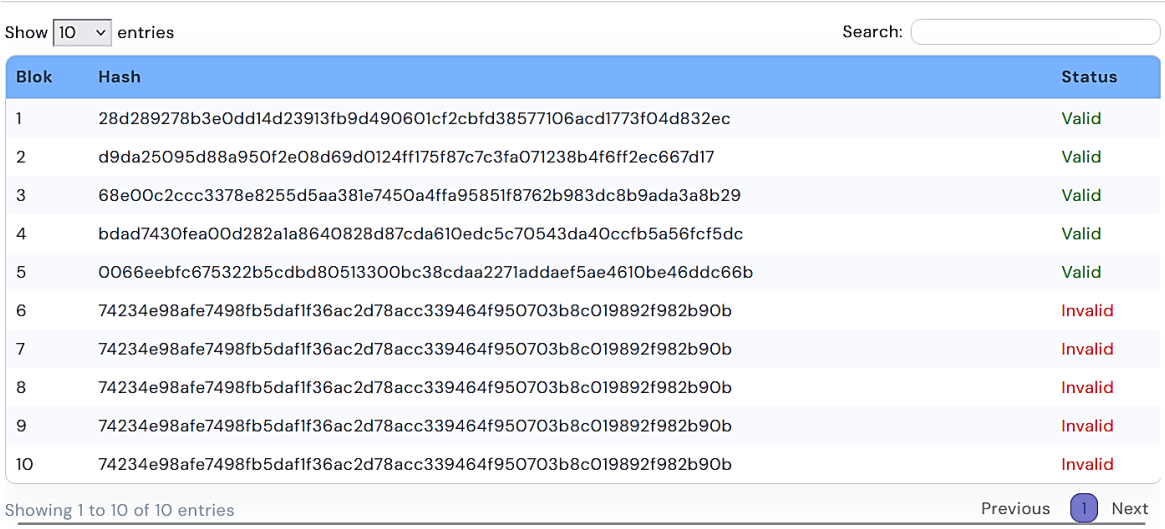


Show 10 entries Search:

Blok	Hash	Status
1	28d289278b3e0dd14d23913fb9d490601cf2cbfd38577106acd1773f04d832ec	Valid
2	d9da25095d88a950f2e08d69d0124ff175f87c7c3fa071238b4f6ff2ec667d17	Valid
3	68e00c2ccc3378e8255d5aa381e7450a4ffa95851f8762b983dc8b9ada3a8b29	Valid
4	bdad7430fea00d282a1a8640828d87cda610edc5c70543da40ccfb5a56fcf5dc	Valid
5	0066eebfc675322b5cdbd80513300bc38cdaa2271addaef5ae4610be46ddc66b	Valid
6	7a9ca7bc2a6f691a79d7bb15dcb1b24d18591e2809cb0a27200b79a379372fc8	Valid
7	f527a0e30904235461b7c8402d72040556d23e7237dede0c97b896a0c9573e21	Valid
8	55a9e30ff130200a8c70a5a74323f3b66b806c04ebd8f88e7f2f307e9f03989c	Valid
9	4ef2c7ca51c2ac016341849e2207d99c25495e2b0098ce880c96eb399616ca6b	Valid
10	8734b280f47e9bc70fba25f583f0b2efe449bd13fc960c071a051a24f5fce99f	Valid

Showing 1 to 10 of 10 entries Previous 1 Next

(a)



Show 10 entries Search:

Blok	Hash	Status
1	28d289278b3e0dd14d23913fb9d490601cf2cbfd38577106acd1773f04d832ec	Valid
2	d9da25095d88a950f2e08d69d0124ff175f87c7c3fa071238b4f6ff2ec667d17	Valid
3	68e00c2ccc3378e8255d5aa381e7450a4ffa95851f8762b983dc8b9ada3a8b29	Valid
4	bdad7430fea00d282a1a8640828d87cda610edc5c70543da40ccfb5a56fcf5dc	Valid
5	0066eebfc675322b5cdbd80513300bc38cdaa2271addaef5ae4610be46ddc66b	Valid
6	74234e98afe7498fb5daf1f36ac2d78acc339464f950703b8c019892f982b90b	Invalid
7	74234e98afe7498fb5daf1f36ac2d78acc339464f950703b8c019892f982b90b	Invalid
8	74234e98afe7498fb5daf1f36ac2d78acc339464f950703b8c019892f982b90b	Invalid
9	74234e98afe7498fb5daf1f36ac2d78acc339464f950703b8c019892f982b90b	Invalid
10	74234e98afe7498fb5daf1f36ac2d78acc339464f950703b8c019892f982b90b	Invalid

Showing 1 to 10 of 10 entries Previous 1 Next

(b)

Figure 2. Results of system testing: (a) all valid data scenario and (b) altered data block scenario

The integrity testing results underscore the system's ability to detect even minimal data alterations, which is essential for maintaining trust in distributed environments. This behavior reflects the Merkle Tree's strength in anomaly detection, where a single block change invalidates the entire chain. Critically, this implies that the proposed model is not only effective in preserving data integrity but also serves as a proactive mechanism for identifying tampering in real-time. Interpretation of these results suggests that hierarchical hashing is a reliable strategy for ensuring data consistency. The implications are significant for applications in blockchain-based education systems and secure cloud platforms, where data authenticity is paramount.

5.2. Efficient testing

This section presents a comparative analysis of the efficiency of different cryptographic algorithm combinations, explicitly focusing on AES+RSA, Twofish+El-Gamal, AES+El-Gamal, and Twofish+RSA. By employing Big O Notation alongside execution time metrics [30]. The analysis aims to evaluate each algorithm pair's computational efficiency and performance speed, as shown in Tables 1 and 2 and Figure 3. This approach allows for a comprehensive assessment of their suitability for secure data processing and real-time applications.

Table 1. Comparative complexity on a different algorithm

Algorithm	Key Preparation	Encryption	Decryption
AES+RSA	$O(k^3)$ (RSA) $O(r \cdot n^2)$ (AES)	$O(r \cdot n^2) + O(k^3)$	$O(r \cdot n^2) + O(k^3)$
Twofish+ElGamal	$O(k^3)$ (ElGamal) $O(n^2)$ (Twofish)	$O(n^2) + O(k^3)$	$O(n^2) + O(k^3)$

Table 2. Comparative time execution on different algorithms

Block	Time Execution (ms)	
	AES+RSA	Twofish+El-Gamal
1	0,266	7,120
2	0,644	6,725
3	0,981	7,758
4	1,340	11,209
5	1,758	12,614
6	2,205	12,721
7	2,671	16,390
8	3,016	15,181
9	3,668	16,544
10	3,995	20,309

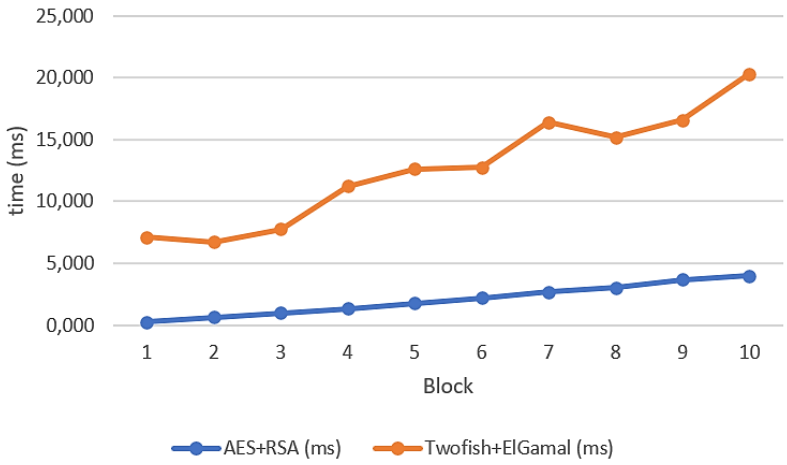


Figure 3. Comparison of execution time between AES+RSA and Twofish+ElGamal algorithms

Based on Tables 1 and 2 and Figure 3, the analysis evaluates both algorithmic complexity and execution time across different cryptographic combinations. This comparison highlights the relative effectiveness and efficiency of each approach. Each algorithm exhibits distinct characteristics, making them suitable for varying security requirements and application contexts.

AES+RSA demonstrates a well-balanced approach with efficient execution times ranging from 0.266 to 3.995 ms across different blocks. The complexity of $O(k^3)$ for RSA ensures secure vital exchanges, while $O(r \cdot n^2)$ for AES facilitates fast encryption and decryption. This combination ensures rapid data processing, making it highly suitable for applications requiring both speed and security.

Twofish+ElGamal shows high execution times, notably peaking at 20.309 ms, which reflects its significant computational overhead due to $O(k^3)$ complexity associated with ElGamal's key management. Although secure, this combination may not be optimal for environments demanding quick data processing. The efficiency analysis reveals that AES+RSA achieves optimal performance compared to other algorithm pairs, with significantly lower execution times. From a critical standpoint, the results validate the feasibility of hybrid cryptography in real-time systems.

5.3. Discussion of findings and contribution to the field

The findings of this study confirm that integrating AES and RSA within a hash chain system, supported by Merkle tree structures, significantly enhances both data integrity and encryption efficiency. This aligns with prior research that highlights the individual strengths of these cryptographic components [6], [12], but this study advances the field by demonstrating their combined effectiveness in a unified framework.

The ripple effect observed in the Merkle tree validation—where a single data block alteration invalidates subsequent blocks—illustrates the model's robustness in anomaly detection [3], [16]. This behavior is particularly valuable in distributed systems where real-time integrity verification is critical.

Compared to other algorithm pairs, such as Twofish+ElGamal, the AES+RSA combination consistently outperforms in execution time and complexity. These results not only support earlier findings but also extend their implications to high-throughput environments such as educational platforms, IoT networks, and secure cloud infrastructures.

Collectively, these insights underscore the practical relevance and technical soundness of the proposed model. It achieves a balanced trade-off between speed, security, and integrity, thereby laying a strong foundation for future research aimed at optimizing hybrid encryption systems and adapting them to emerging technologies and distributed architectures. Moreover, the findings of this study have the potential to contribute meaningfully to the scope of Information Technology and Information Systems, particularly in the domains of security and cryptograph. Its emphasis on algorithmic performance, data integrity, and system scalability directly addresses key concerns in modern digital infrastructure, making it highly relevant to both academic researchers and industry professionals seeking robust solutions in cybersecurity and information systems.

6. CONCLUSION

This study explored hybrid cryptography with RSA and AES to enhance system integrity within hash chain frameworks augmented by Merkle trees. AES+RSA was identified as the most efficient combination, providing swift processing and robust security ideal for data-intensive environments. The integration of hash chains and Merkle trees significantly bolstered data integrity, ensuring rapid detection of inconsistencies or unauthorized changes. While other combinations like Twofish+ElGamal remain robust, their longer processing times highlight the superior efficiency of AES+RSA. Future work should further optimize cryptographic strategies and explore IoT applications, leveraging hash chain and Merkle tree frameworks to enhance data protection and system reliability in evolving technological landscapes.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Irza Nur Fauzi	✓	✓	✓		✓	✓	✓	✓	✓		✓		✓	✓
Farikhin		✓		✓	✓	✓	✓			✓		✓		✓
Ferry Jie		✓		✓	✓	✓	✓			✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

ETHICAL APPROVAL

The conducted research is not related to either human or animal use.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author INF on request.

REFERENCES

[1] S. Joseph and W. Fred, "Network security in the old age: protecting information and structures." Nov. 2023.

[2] Y. Zongkai, L. Weimin, and T. Yunmeng, "A new fair micropayment system based on hash chain," in *Proceedings - 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, EEE 2004*, 2004, pp. 139–145, doi: 10.1109/eee.2004.1287300.

[3] O. Kuznetsov, A. Rusnak, A. Yezhov, K. Kuznetsova, D. Kanonik, and O. Domin, "Merkle trees in blockchain: A Study of collision probability and security implications," *Internet of Things (Netherlands)*, vol. 26, p. 101193, Jul. 2024, doi: 10.1016/j.iot.2024.101193.

[4] D. Tiwari, A. Singh, and A. Prabhakar, "Performance analysis of AES, RSA and hashing algorithm using web technology," in *Springer*, 2020, pp. 413–418.

[5] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018, doi: 10.1504/IJWGS.2018.095647.

[6] E. Jintcharadze and M. Iavich, "Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems," in *2020 IEEE East-West Design and Test Symposium, EWDTS 2020 - Proceedings*, 2020, pp. 1–5, doi: 10.1109/EWDTS50664.2020.9224901.

[7] Y. MEI, "Using the HashChain to improve the security of the Hadoop," 2017, doi: 10.2991/eeeis-17.2017.82.

[8] E. G. Abdallah, Y. R. Kuang, and C. Huang, "Advanced encryption standard new instructions (AES-NI) analysis: security, performance, and power consumption," in *ACM International Conference Proceeding Series*, 2020, pp. 167–172, doi: 10.1145/3384613.3384648.

[9] I. C. Sari, M. Zarlis, and T. Tulus, "Optimization of data encryption modeling using RSA cryptography algorithm as security e-mail data," *Journal of Physics: Conference Series*, vol. 1471, no. 1, p. 12068, 2020, doi: 10.1088/1742-6596/1471/1/012068.

[10] S. Jing, X. Zheng, and Z. Chen, "Review and investigation of Merkle Tree's technical principles and related application fields," in *Proceedings - 2021 International Conference on Artificial Intelligence, Big Data and Algorithms, CAIBDA 2021*, 2021, pp. 86–90, doi: 10.1109/CAIBDA53561.2021.00026.

[11] S. Nagaraj and E. Mohanraj, "Enhanced selective encryption method for bigdata sensing stream using one way Hash Chain algorithm," in *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, 2020, pp. 297–302, doi: 10.1109/ICACCCN51052.2020.9362915.

[12] S. Gautam, S. Singh, and H. Singh, "A comparative study and analysis of cryptographic algorithms: RSA, DES, AES, BLOWFISH, 3-DES, and TWOFISH," *International Journal of Research in Electronics and Computer Engineering*, vol. 7, no. 1, 2019.

[13] S. A. Sholikhatin, A. P. Kuncoro, A. L. Munawaroh, and G. A. Setiawan, "Comparative study of RSA asymmetric algorithm and AES algorithm for data security," *Edu Komputika Journal*, vol. 9, no. 1, pp. 60–67, Jun. 2022, doi: 10.15294/edukomputika.v9i1.57389.

[14] L. Zou, M. Ni, Y. Huang, W. Shi, and X. Li, "Hybrid encryption algorithm based on AES and RSA in file encryption," in *International conference on frontier computing*, 2020, pp. 541–551.

[15] C. Zhao, M. Shi, M. Huang, and X. Du, "Authentication scheme based on Hashchain for space-air-ground integrated network," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761821.

[16] M. Han and W. Jiang, "A secure communication method based on message Hash Chain," *Applied Sciences*, vol. 12, no. 9, p. 4505, Apr. 2022, doi: 10.3390/app12094505.

[17] S. K. Waybhase and P. Adakane, "Data security using advanced encryption standard (AES)," *International Journal of Engineering Research & Technology (IJERT)*, vol. 11, no. 06, pp. 630–632, 2022, doi: 10.17577/IJERTV11IS060338.

[18] A. Muhammad Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16.1, p. 11, 2017, [Online]. Available: https://www.researchgate.net/publication/317615794.

[19] S. Zabell, "Cryptology, mathematics, and technology," in *Technology and Mathematics: Philosophical and Historical Investigations*, 2018, pp. 137–161.

[20] J. A. Buchmann, *Introduction to cryptography*. New York, NY: Springer New York, 2004.

[21] Y. Liu, W. Gong, and W. Fan, "Application of AES and RSA hybrid algorithm in e-mail," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Jun. 2018, pp. 701–703, doi: 10.1109/ICIS.2018.8466380.

[22] R. M. Avanzi and T. Lange, "Introduction to public-key cryptography," *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, pp. 1–15, 2005, doi: 10.1007/978-3-662-69007-9_6.

[23] J. Ikbal, "An introduction to cryptography," *Information Security Management Handbook, Sixth Edition*, pp. 1121–1139, 2007, doi: 10.55621/idpro.102.

[24] F. H. M. S. Al-Kadei, H. A. Mardan, and N. A. Minas, "Speed up image encryption by using RSA algorithm," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2020, pp. 1302–1307, doi: 10.1109/ICACCS48705.2020.9074430.

[25] Qiuyuan Huang, Renzhi Cao, Bobin Deng, and Xingfu Wang, "Hash-chain based public key management algorithm of mobile ad hoc network," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, Jun. 2011, pp. 247–251, doi: 10.1109/CSAE.2011.5953214.

[26] S. Gangadharaiah and P. Shrinivasacharya, "Secure and efficient public auditing system of user data using hybrid AES-ECC crypto system with Merkle hash tree in blockchain," *Multimedia Tools and Applications*, vol. 83, no. 29, pp. 72301–72320, Feb. 2024, doi: 10.1007/s11042-024-18363-0.

[27] D. Lee and N. Park, "Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree," *Multimedia Tools and Applications*, vol. 80, no. 26–27, pp. 34517–34534, Nov. 2021, doi: 10.1007/s11042-020-08776-y.

[28] T. Rathee and P. Singh, "Secure data sharing using Merkle hash digest based blockchain identity management," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3851–3864, Nov. 2021, doi: 10.1007/s12083-021-01212-4.

[29] K. S, K. A. S, I. F. A, H. P. S, and L. M, "Advanced encryption standard to prevent intruders in email through cloud environment," in *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, May 2023, pp. 1183–1189, doi: 10.1109/ICAAIC56838.2023.10140373.

[30] W. N. A. Ruzai, M. R. K. Ariffin, M. A. Asbullah, and A. H. A. Ghafar, "New simultaneous Diophantine attacks on generalized RSA key equations," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, p. 102074, Jun. 2024, doi: 10.1016/j.jksuci.2024.102074.

BIOGRAPHIES OF AUTHORS

Irza Nur Fauzi is a student in the Master of Information Systems program at the School of Postgraduate, Diponegoro University, Semarang, Indonesia. He holds a bachelor's degree in electrical engineering with a concentration in Information Technology from the Faculty of Engineering, Diponegoro University. His research interests focus on information systems and information technology, particularly in the areas of data integrity, system security, and digital infrastructure. He can be contacted at email: irzanurf@gmail.com.

Farikhin is a lecturer and researcher at the Department of Mathematics, Faculty of Science and Mathematics, Diponegoro University, Indonesia. He earned his bachelor's and master's degrees in mathematics from Universitas Gadjah Mada (UGM), and completed his Ph.D. in Computational Analysis at Universiti Malaysia Terengganu. His research interests include mathematical analysis, information system and computational. He can be contacted at email: farikhin.math.undip@gmail.com.

Ferry Jie is an associate professor and Course Coordinator for Supply Chain and Logistics Management at the School of Business and Law, Edith Cowan University, Australia. He obtained his Ph.D. in Supply Chain Analysis from The University of Sydney in 2008. His research focuses on supply chain and logistics management, operations management, quantitative methods, and strategic management. He can be contacted at email: jieferry@yahoo.com.