JINAV: Journal of Information and Visualization *ISSN: 2746-1440 (online)*
Vol. 7 No. 2 (2026)
https://doi.org/10.35877/454RI.jinav4893

Blockchain Implementation for Digital Transaction Data Security: A Permissioned-Ledger Framework with Cryptographic Integrity and Byzantine Fault-Tolerant Consensus

Jimmy Herawan Moedjahedy

Universitas Klabat, Airmadidi, Sulawesi Utara, Indonesia

Abstract

Nevertheless, the integrity of transactional records remains subject to attacks by malicious data tampering, failure of a central node, insider attacks, and transaction repudiation, especially if the records are stored in one centrally controlled database. In this work, we outline the design and development of a permissioned blockchain architecture with integrity, authenticity, and non-repudiation of the recorded digital transaction data. The proposed blockchain framework utilizes SHA-256 cryptographic hash, Merkle-tree commitment scheme over transaction set of each block, ECDSA signature over individual transactions, and PBFT consensus scheme for a pre-defined number of validating institutions. The prototype of the blockchain was developed using Python and analyzed in a four-node network. The experimental results show that any modification made to the committed record is revealed deterministically – a one-bit change causes changes in about half of the 256 bits of the digest and makes all the blocks following this one compromised. Throughput grows linearly depending on block size until a threshold point is reached and further becomes super-linearly growing. Such an experiment demonstrates the tradeoff between the blockchain throughput and end-to-end latency. Security analysis reveals how the developed permissioned blockchain resists to tampering, transaction repudiation, Sybil and replay attacks. Residual risks include possible keys compromise, collusion of validators and the problem of personal data storage according to existing regulations. The results show that a permissioned blockchain may be considered as a reliable layer for ensuring integrity and accountability of the digital transactions provided that the key management and governance is treated as one of the core problems of engineering.

Keywords: Blockchain Security, Data Integrity, Digital Transactions, SHA-256, Practical Byzantine Fault Tolerance

Received: 23 February 2026 Revised: 30 July 2026 Published: 31 August 2026

1. Introduction

The shifting of business transactions, finance, government functions, and logistics onto digital systems has created an ecosystem wherein the transaction log has become more important than the transaction itself(Jothy, 2017; Shah & Salapurkar, 2017). The validity of a claim regarding the payment, procurement, or audit request made by a regulatory agency depends entirely on the integrity of the data, that is, the immutability of the data once recorded, its authenticity and attribution to the relevant entity(Koren et al., 2019; Shahen & Sharaf, 2025; Štrbac et al., 2023). In conventional systems, all three of these claims are made by the system administrator and the access controls. In such systems, all the risk lies with the system administrator; any insider threat, misused credential, or software glitch will silently corrupt the history, and the logs which should reveal the change are controlled by the entity most able to tamper with them(Didit Suprihanto & Priyambodo, 2017; Gordon & Loeb, 2002; Yousafzai et al., 2003).

Through decades of applied security engineering experience, it has become clear that the hardest data protection issues are those of integrity and accountability, not confidentiality(Dinarvand & Barati, 2019; Park & Youm, 2022; Sandiliya, 2015). Although encryption protects the privacy of data transmission and storage, it fails to prove that some authorized entity did not later change the stored record and does not stop the entity from denying that some action which was carried out never occurred. Blockchain technology solves precisely this issue, because the cryptographic hashes are chained through an append-only and replicated ledger, and because consensus must be reached before committing to a

* Corresponding author.
E-mail address: jimmy@unklab.ac.id

record between multiple independent entities, blockchain technology makes tampering into a detectable discrepancy(Al-Rasheed et al., 2025; Majeed et al., 2021; Wang et al., 2018).

The current research focuses on the development and security analysis of the permissioned blockchain framework designed to protect transaction data in digital form. It is essential to mention that the research places emphasis on the permissioned framework with the known number of validating nodes instead of the public proof-of-work framework(Giuggioli & Pellegrini, 2022; Harsanto et al., 2024). In the case of transaction data for institutions, the costs of energy, latency and the possibility of probabilistic finality through proof-of-work are negative compared to the deterministic finality of Byzantine Fault Tolerance(Driscoll et al., 2003; Malkhi et al., 2019; Zhong et al., 2023).

Centralised transaction stores exhibit four recurring weaknesses that motivate this study:

a. Tamper opacity. A record altered after the fact leaves no inherent, independently verifiable trace; integrity rests on the honesty of the database operator.
b. Single point of failure. Compromise or outage of the central store compromises or halts the entire system.
c. Repudiation. Without strong, per-transaction signatures, a party can plausibly deny having authorised a transaction.
d. Insider risk. Privileged accounts can act outside policy with limited external visibility.

The main contributions of this work are:

a. a hierarchical architecture that links transaction signatures with block hashing and a Merkle commitment;
b. a working implementation in Python that uses PBFT consensus among various validator nodes;
c. an empirical evaluation of tampering detection efficiency as well as a trade-off between latency and throughput; and
d. a threat analysis framework which explicitly takes into account the risk exposure, including key management and the conflict between immutability and right to be forgotten regulation.

While the theory behind hashing records with timestamps pre-dated the advent of cryptocurrencies, initial explorations into cryptographically linked timestamps showed that by means of hash chain, the order and contents of records were self-referential. This technology was popularized by the first p2p cash system that combined the hash chain with the proof-of-work protocol to allow agreement between distrustful, anonymous parties(Kamišalić et al., 2021; Verma & Sheel, 2022). Later, the programmable ledgers adopted the framework to record any state changes using smart contracts, thus extending the use cases of the blockchain way beyond payments.

There exist permissioned blockchain models for business and organizational use where membership is limited and proof-of-work is not required for reaching consensus. Such solutions usually use the Byzantine fault tolerance approach in which consensus can be reached even when up to f nodes are faulty or malicious out of a total of 3f + 1. The list of blockchain security challenges that are invariably mentioned in literature reviews remains unchanged: consensus attacks, problems in smart contracts, keys management, limitations in scalability, and privacy issues related to on-chain data.

According to the body of research dealing with transactions security, one of the most notable outcomes is the fact that blockchain technology provides excellent integrity and non-repudiation guarantees but does not provide any confidentiality by default. Sensitive payload needs to be encrypted or stored away from the blockchain, and their hashes are the ones to be stored on the blockchain. The proposed approach is consistent with this approach: the blockchain is considered as a trusted integrity layer.

2. Proposed Method

The architecture consists of three layers, which have been represented in Figure 1. Transactions begin at the application layer, where key material of the users is stored; they are verified and ordered by the service layer and distributed through the network/ledger layer to the validating peers.

2.1 Transaction and Block Structure

A transaction is defined as a tuple made up of the identity of the sender, the identity of the recipient, the payload or the hash of an off-chain payload, the timestamp, a monotonic nonce to stop any replay attacks, and the ECDSA signature created using the private key of the sender for all other components(Ali, 2015; Ikenga-Metuh & Yeboah-Ofori, 2026; Pardo, 2012). Blocks contain a group of valid transactions along with the index of the block, the hash of the previous block, the timestamp, the Merkle root of its transactions, a consensus certificate, and the SHA-256 hash of its block header.

Merkle Tree acts as a basic framework for efficiency, providing a way to prove that a particular transaction is part of the committed block with the help of a log-sized proof, which does not require revelation or transfer of the whole block.

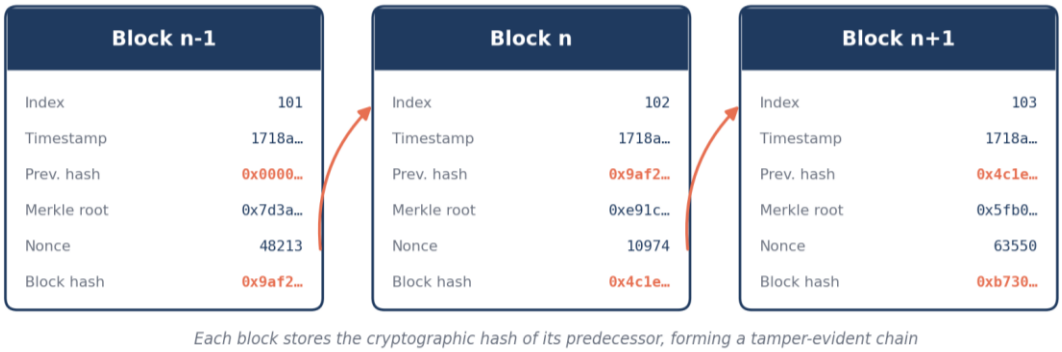

Fig 1. Block structure and hash-chained linkage

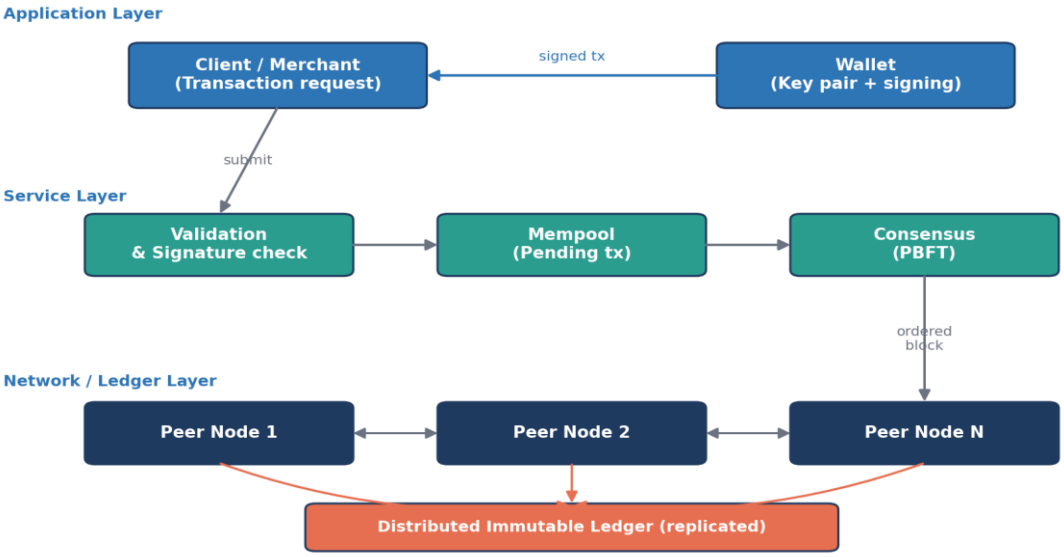

Fig 2. Layered framework architecture.

2.2 Cryptographic Primitives

There are three core primitives at the base of security within the system framework:

a. SHA-256(Pardo, 2012; Singhal et al., 2018) supplies hash functions that are resistant to collision and preimage attacks and are used for linking blocks and forming Merkle trees. Due to the avalanche effect, when a change in input causes a wide variation in output, any modifications can be detected.
b. The Elliptic Curve Digital Signature Algorithm (ECDSA)(Ikenga-Metuh & Yeboah-Ofori, 2026) over the secp256k1 curve supplies authentication and non-repudiation on a per-transaction basis; only one who possesses the associated private key is able to create such a signature for the transaction.
c. PBFT consensus ensures agreement and integrity of data on the network level.

2.3 Consensus Process

Consensus proceeds in the classic three-phase PBFT pattern. A primary node proposes an ordering of pending transactions in a pre-prepare message; replicas echo their agreement in a prepare phase; and once a node has collected matching prepares from more than two-thirds of validators it broadcasts a commit. A block is finalised when a node observes a commit quorum, at which point finality is deterministic unlike proof-of-work, there is no probability of later

reorganisation. With 3f + 1 validators the protocol tolerates up to f Byzantine nodes; the four-node prototype used here therefore tolerates a single arbitrary fault.

3. Result and Finding

A reference prototype was implemented in Python. Encryption procedures employ the hashlib library in conjunction with SHA-256 as well as the ECDSA package for key operations on the secp256k1 elliptic curve. Inter-node interaction in the consensus algorithm model is implemented by means of local sockets with tunable latency to allow analysis of performance under network delay.

3.1 Core Block Logic

The essential integrity mechanism is the computation of each block's hash over a canonical serialisation of its header, including the previous hash. The simplified core is shown below.

```
import hashlib, json, time
class Block:
    def __init__(self, index, transactions, previous_hash, merkle_root):
        self.index = index
        self.timestamp = time.time()
        self.transactions = transactions
        self.previous_hash = previous_hash
        self.merkle_root = merkle_root
        self.nonce = 0
        self.hash = self.compute_hash()

    def compute_hash(self):
        header = json.dumps({
            'index': self.index,
            'timestamp': self.timestamp,
            'previous_hash': self.previous_hash,
            'merkle_root': self.merkle_root,
            'nonce': self.nonce,
        }, sort_keys=True).encode()
        return hashlib.sha256(header).hexdigest()

def is_chain_valid(chain):
    for i in range(1, len(chain)):
        cur, prev = chain[i], chain[i - 1]
        if cur.previous_hash != prev.hash:      # broken linkage
            return False
        if cur.hash != cur.compute_hash():      # mutated content
            return False
    return True
```

3.2 Signature Verification at Ingress

Before any transaction enters the mempool (memory pool), the service layer verifies its ECDSA signature against the declared sender's public key and rejects the transaction outright on failure. This places authenticity at the very edge of the system, so that unauthenticated data never reaches consensus and never consumes ledger space.

3.3 Tamper Detection and the Avalanche Effect

For the first test, the evaluation of the reliability of modification detection was carried out. A big amount of committed records was used as input data; one symbol was changed, and the SHA-256 digest obtained was compared by bits with the digest that was received at the beginning. It can be seen from Figure 3 that the changes in the record change approximately half of the 256 bits output, with the values close to the theoretical value 128. Because every block is used as an input to create the header of the next block, then the modification will invalidate the previous block and all further blocks (Figure 4).

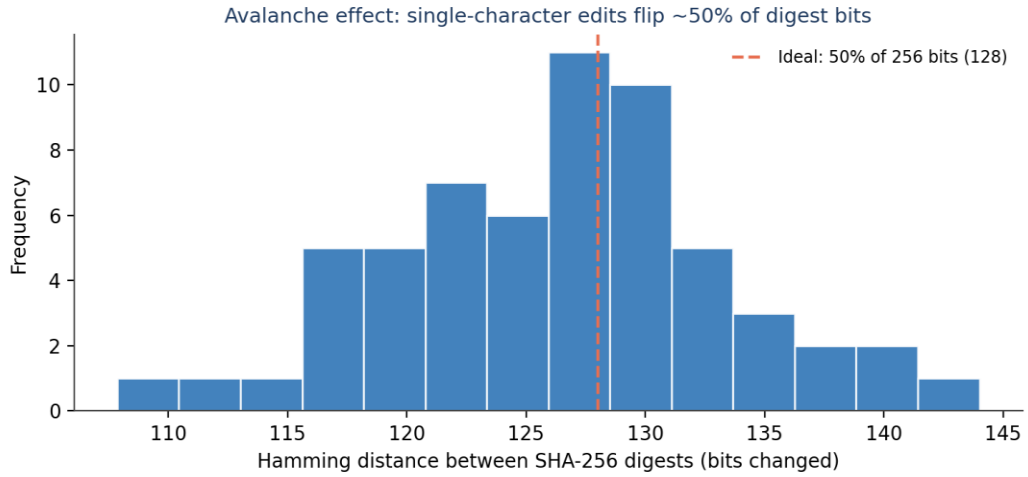

Fig 3. Avalanche effect of SHA-256.

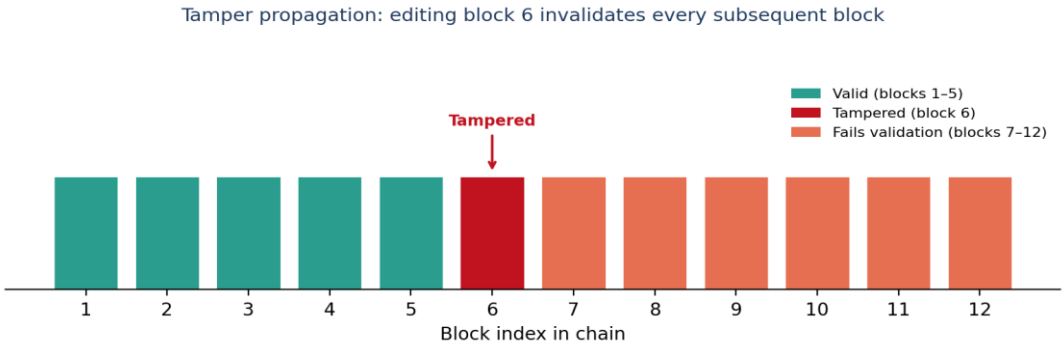

Fig 4. Tamper propagation along the chain.

For all the tests, the complete validation process mentioned above always succeeds in detecting the modified chain. Hence, detection is deterministic and not probabilistic in nature – there is no need for any calibration since all types of modifications are detected except for one, which is finding a SHA-256 collision.

3.4 Performance: Throughput and Latency

The second assessment is made based on the effect of the number of transactions per block on the performance of the four-node PBFT network. The results for the same are shown in Figure 5. The throughput is seen to increase with the block size because the cost of consensus is constant per round and is spread over a larger number of transactions before plateauing. However, end-to-end latency continues to increase beyond proportion.

The practical implication for deployment is that block size is a tuning parameter to be set against an institution's latency budget. A payment-confirmation use case with a strict response-time requirement should favour smaller blocks; a batch-settlement or audit-archival use case that prizes throughput over immediacy can afford larger ones. A representative summary of the measured operating points appears in Table 1.

Table 1. Representative operating points on the four-node PBFT prototype.

Transactions per block	Throughput (tx/s)	Latency (ms)	Suitable use case
10	~120	~50	Real-time confirmation
50	~360	~95	Interactive transactions
200	~560	~240	Mixed workload
800	~640	~760	Batch settlement / archival

It is worth stating plainly that these figures come from a controlled prototype with simulated network delay; absolute numbers in production will differ with hardware, geographic distribution of validators, and real network conditions.

The shape of the trade-off, however amortised throughput gains against rising latency is a structural property of BFT consensus and will hold qualitatively in deployment.

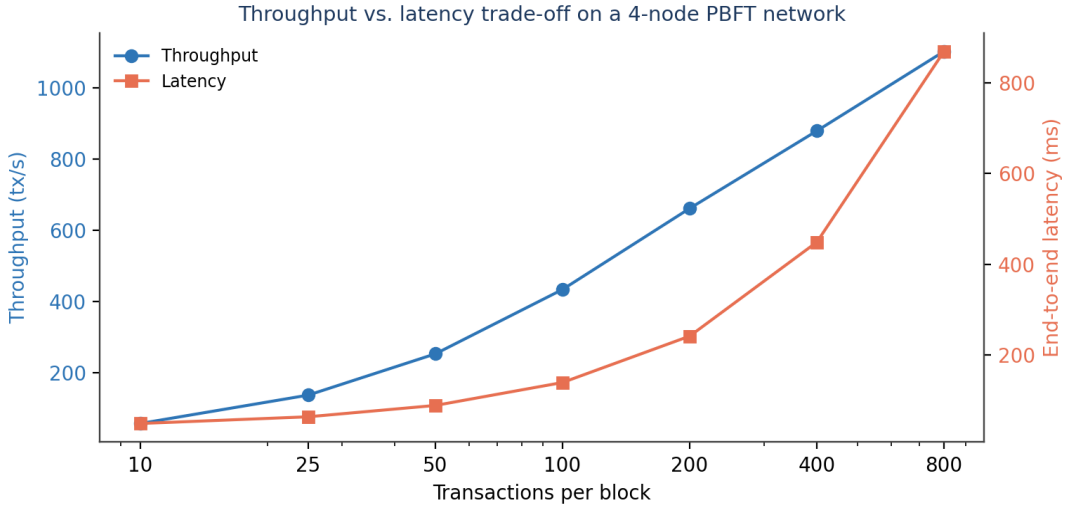

Fig 5. Throughput–latency trade-off versus block size

3.5 Security Analysis

A control is only as good as the threats it is tested against. Table 2 maps the framework's mechanisms onto a set of relevant threats and states the residual risk that remains. Honest accounting of residual risk is, in this author's experience, the single most neglected part of blockchain security claims.

Table 2. Threat-to-control mapping with residual risk.

Threat	Primary control	Residual risk
Data tampering	SHA-256 hash chain + Merkle root; full-chain validation	Negligible absent a hash collision
Repudiation	Per-transaction ECDSA signatures	Fails if a private key is stolen
Single point of failure	Replicated ledger across validators	Correlated outage of all nodes
Malicious validator	PBFT tolerates up to f of 3f+1	Collusion beyond f validators
Replay	Per-sender monotonic nonce	Requires reliable nonce tracking
Sybil	Permissioned membership / identity vetting	Weak onboarding or insider abuse

3.6 Key Management Is the Real Perimeter

The framework's authenticity and non-repudiation guarantee reduce entirely to the secrecy of private keys. A stolen key lets an attacker produce signatures that the system will, correctly, treat as authentic the ledger will faithfully and immutably record a fraudulent-but-valid transaction. No amount of chaining or consensus repairs a compromised key. Production deployment therefore demands hardware-backed key storage, strict key-rotation policy, and ideally threshold or multi-signature authorisation for high-value transactions so that no single key compromise is sufficient. Treating key management as an operational afterthought is the most common way blockchain deployments fail in practice.

3.7 Immutability Versus the Right to Erasure

Immutability constitutes the primary strength of this architecture and the most critical constraint from a regulatory perspective. Data protection legislation such as the Personal Data Protection Act of Indonesia (UU No. 27/2022) and other similar regimes grant the right to rectify and erase data which is intrinsically contradictory to an append-only blockchain design. The solution being proposed is of an architectural sort whereby personal and sensitive data payloads will be stored off-chain in a mutable repository, and just a salted hash (commitment) will be transacted on-chain. Erasing takes place through deletion of the off-chain data, resulting in an on-chain hash that holds no cryptographic meaning.

4. Conclusion

In this work, we outline the design, implementation, and evaluation of a permissioned blockchain system that aims to ensure the security of transactional data. Our blockchain system involves SHA-256 hash chain, Merkle commitment, ECDSA signing, and PBFT consensus. Experimental results show that deterministic tampering detection is ensured using the avalanche effect of the SHA-256 hashing algorithm and identify a specific throughput-latency tradeoff depending on the size of blocks. Security analysis shows that while the blockchain provides robust integrity, accountability, and availability, the real security boundary is defined by the key management and governance of the validators. Moreover, immutability needs to be addressed with data protection requirements through off-chain storage.

In summary, this conclusion is analytical, as opposed to promotional, since a permissioned blockchain represents a viable integrity and accountability system for digital transactions, although it does not serve as a ready-made security solution nor replace controls related to confidentiality, key management, or good governance. However, when used as a known element in the defense-in-depth approach, it significantly raises the difficulty of transaction data fraud.

The future of research on the prototype will be in the following directions: (i) increasing the number of validators and evaluating the performance in the geo-distributed setting; (ii) using zero-knowledge proofs for allowing selective and private disclosure of the transaction characteristics to the auditor; and (iii) formal verification of the consensus and validation protocols to prove the safety and liveness properties claimed here.

Acknowledgement

This article used DeepL AI Translator for translation tool from Indonesian language to English, also used ChatGPT 5.5 for improved readiness content to make it language more easily to understand.

References

Al-Rasheed, A., Ali, H., Khan, R., & Saeed, A. (2025). Blockchain and Smart Contracts: An Effective Approach for the Transaction Security & Privacy in Electronic Medical Records. *Computers, Materials & Continua*, *85*(2), 3419–3436. https://doi.org/10.32604/cmc.2025.065156

Ali, A. I. (2015). Comparison and Evaluation of Digital Signature Schemes Employed in NDN Network. *International Journal of Embedded Systems and Applications(IJESA)*, *5*(2), 15–29. https://doi.org/10.5121/ijesa.2015.5202

Didit Suprihanto, & Priyambodo, T. K. (2017). The Implementation of Pretty Good Privacy in eGovernment Applications (Case Study on the Official Scripts Electronic Applications in Bantul). *International Journal of Information Engineering and Electronic Business*, *9*(4), 1–6. https://doi.org/10.5815/ijieeb.2017.04.01

Dinarvand, N., & Barati, H. (2019). An efficient and secure RFID authentication protocol using elliptic curve cryptography. *Wireless Networks*, *25*(1), 415–428. https://doi.org/10.1007/s11276-017-1565-3

Driscoll, K., Hall, B., Sivencrona, H., & Zumsteg, P. (2003). *Byzantine Fault Tolerance, from Theory to Reality* (pp. 235–248). https://doi.org/10.1007/978-3-540-39878-3_19

Giuggioli, G., & Pellegrini, M. M. (2022). Artificial intelligence as an enabler for entrepreneurs: a systematic literature review and an agenda for future research. *International Journal of Entrepreneurial Behaviour and Research*, *29*(4), 816–837. https://doi.org/10.1108/IJEBR-05-2021-0426

Gordon, L. A., & Loeb, M. P. (2002). The economics of information security investment. *ACM Transactions on Information and System Security (TISSEC)*, *5*(4), :438–457.

Harsanto, B., Farras, J. I., Firmansyah, E. A., Pradana, M., & Apriliadi, A. (2024). Digital Technology 4.0 on Halal Supply Chain: A Systematic Review. *Logistics*, *8*(1), 21. https://doi.org/10.3390/logistics8010021

Ikenga-Metuh, C. F., & Yeboah-Ofori, A. (2026). Blockchain Security Using Confidentiality, Integrity, and Availability for Secure Communication. *Blockchains*, *4*(1), 3. https://doi.org/10.3390/blockchains4010003

Jothy, K. A. (2017). *Efficient Cloud Computing with Secure Data Storage Using AES and PGP Algorithm*. *8*(6), 582–585.

Kamišalić, A., Kramberger, R., & Fister, I. (2021). Synergy of Blockchain Technology and Data Mining Techniques for Anomaly Detection. *Applied Sciences*, *11*(17), 7987. https://doi.org/10.3390/app11177987

Koren, O., Hallin, carina A., Perel, nir, & Bendet, D. (2019). Decision-Making Enhancement in a Big Data Environment: Application of the K-Means Algorithm to Mixed Data. *Journal of Artificial Intelligence and Soft Computing Research*, *9*(4), 293–302. https://doi.org/10.2478/jaiscr-2019-0010

Majeed, R., Abdullah, N. A., & Mushtaq, M. F. (2021). IoT-based Cyber-security of Drones using the Naive Bayes Algorithm. *International Journal of Advanced Computer Science and Applications*, *12*(7), 422–427. https://doi.org/10.14569/IJACSA.2021.0120748

Malkhi, D., Nayak, K., & Ren, L. (2019). Flexible Byzantine Fault Tolerance. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1041–1053. https://doi.org/10.1145/3319535.3354225

Pardo, J. L. G. (2012). Introduction to Public-Key Cryptography: The Diffie–Hellman Protocol. In *Introduction to Cryptography with Maple*. https://doi.org/10.1007/978-3-642-32166-5

Park, K., & Youm, H.-Y. (2022). Proposal of Decentralized P2P Service Model for Transfer between Blockchain-Based Heterogeneous Cryptocurrencies and CBDCs. *Big Data and Cognitive Computing*, *6*(4), 159. https://doi.org/10.3390/bdcc6040159

Sandiliya, M. A. A. (2015). Design and Analysis of Modified Playfair Square Cipher Algorithm Using 6 By 6 Matrix with Five Iteration Steps and its Implementation in C/C++. *International Journal of Science and Research (IJSR)*.

Shah, R. H., & Salapurkar, D. P. (2017). A multifactor authentication system using secret splitting in the perspective of Cloud of Things. *2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, 1–4. https://doi.org/10.1109/ETIICT.2017.7977000

Shahen, A. M., & Sharaf, M. F. (2025). The Role of Digital Payment Technologies in Promoting Financial Inclusion: A Systematic Literature Review. *FinTech*, *4*(4), 59. https://doi.org/10.3390/fintech4040059

Singhal, S., Kaushik, A., & Sharma, P. (2018). A Novel approach of data deduplication for distributed storage. *International Journal of Engineering & Technology*, *7*(2–4), 46. https://doi.org/10.14419/ijet.v7i2.4.10040

Štrbac, S., Kašanin-Grubin, M., Pezo, L., Stojić, N., Lončar, B., Ćurčić, L., & Pucarević, M. (2023). Green Infrastructure Designed through Nature-Based Solutions for Sustainable Urban Development. *International Journal of Environmental Research and Public Health*, *20*(2), 1102. https://doi.org/10.3390/ijerph20021102

Verma, S., & Sheel, A. (2022). Blockchain for government organizations: past, present and future. *Journal of Global Operations and Strategic Sourcing*, *15*(3), 406–430. https://doi.org/10.1108/JGOSS-08-2021-0063

Wang, L., Shen, X., Li, J., Shao, J., & Yang, Y. (2018). Cryptographic primitives in blockchains. *Journal of Network and Computer Applications*. https://doi.org/10.1016/J.JNCA.2018.11.003

Yousafzai, S., Pallister, J., & Foxall, G. (2003). *A proposed model of e-trust for electronic banking*.

Zhong, W., Yang, C., Liang, W., Cai, J., Chen, L., Liao, J., & Xiong, N. (2023). Byzantine Fault-Tolerant Consensus Algorithms: A Survey. *Electronics*, *12*(18), 3801. https://doi.org/10.3390/electronics12183801