Vol. 8, No. 4, August 2026

Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Securing Educational Financial Data Against Insider Threats: A Hybrid Blockchain Approach with Merkle Tree Aggregation

Adi Alfian Hafis[1], Soiful Hadi[2]

[1,2]Department of Information Technology and Communication, Universitas Semarang, Semarang, Indonesia

Received:
February 1, 2026
Revised:
June 18, 2026
Accepted:
July 20, 2026
Published:
August 22, 2026

Corresponding Author:
Author Name*:
Adi Alfian Hafis
Email*:
adialfianhafiz@gmail.com

DOI:
10.63158/journalisi.v8i4.1719

Abstract. Financial information systems in educational institutions face insider threats where privileged administrators can manipulate database records undetected by conventional security. This study proposes a Hybrid Blockchain architecture integrating Merkle Tree Aggregation and a Reversal Entry mechanism to establish a tamper-evident financial audit trail and address these data integrity gaps. Developed via the Design Science Research Methodology (DSRM), the system implements three cryptographic layers: a SHA-256 Recursive Transaction Hash Chain for local integrity, a Keccak-256 Merkle Tree for daily aggregation, and public Ethereum anchoring. Empirical evaluations demonstrate successful cryptographic integrity verification across all five database manipulation attack scenarios with zero false positives only under the five tested normal operational scenarios, relying strictly on deterministic hashing rather than AI-based anomaly detection. Computational latency remains consistent at 3.45 ms per transaction. Based on Sepolia testnet data under mainnet-equivalent projections, the 1,000:1 aggregation compression yields 99.90% cost efficiency compared to pure public blockchains, with sensitivity analysis confirming financial viability across volatile gas prices and exchange rates. These findings indicate the technical and economic feasibility of adopting a Hybrid Blockchain for detecting tampering and preserving data integrity in educational institutional financial data under the evaluated scenario.

Keywords: Hybrid Blockchain, Merkle Tree Aggregation, Insider Threats, Insider Threat Detection, Tamper-Evident Logging, Reversal Entry, Smart Contract.

1. INTRODUCTION

Educational financial systems generally rely on centralized databases, possessing inherent structural vulnerabilities due to excessive single-administrator privileges [1]. Within this context, the insider threat model is specifically defined as a rogue Database Administrator (DBA) or super-administrator possessing full root privileges. Such an actor can bypass application-level controls to directly manipulate data records using raw SQL commands (e.g., executing unauthorized UPDATE or DELETE queries), modify system configurations, or intentionally disable built-in logging mechanisms [2]. This architectural condition opens a critical vector for insider threats, whereby direct database query access enables malicious actors to execute precise financial manipulations such as surgical edits or salami attacks [1], [3] or perform silent deletions without detection. Furthermore, database-level access grants the capability to alter audit trails, directly compromising data integrity and obscuring the validity of forensic investigations into communal fund management [4].

To mitigate database anomalies, conventional cloud and local database systems typically rely on audit logs and perimeter defenses, such as firewalls. However, these traditional audit mechanisms remain insufficient against insider threats because native logs and triggers reside within the same centralized environment, allowing a privileged DBA to easily disable or delete them [5]. Alternatively, enterprise-grade Write-Once-Read-Many (WORM) storage provides robust immutability but introduces prohibitive infrastructure overhead and recurring subscription costs that are not financially viable for medium-scale educational institutions. Furthermore, WORM systems lack the dynamic querying flexibility required for active relational databases [6]. Adopting a pure public blockchain offers architectural transparency without a centralized authority [7]. Nevertheless, implementing public blockchains in high-throughput environments conflicts with scalability constraints and highly volatile, expensive gas fees, making the recording of large-scale datasets financially unviable [8]. Similarly, permissioned consortia networks like Hyperledger Fabric demand highly complex node infrastructures and sophisticated communication designs, creating a substantial barrier to entry [7], [9].

Although blockchain technology has proven effective in securing academic records in regional education [10], a more scalable architecture is required to protect high-volume

school financial data from insider manipulation. MAN 1 Kota Semarang serves as the case-study environment and partial production prototype for this evaluation, selected due to its bidirectional high-volume micro-transactions managed by staff with limited technical literacy and without a dedicated internal auditor, with key stakeholders comprising Financial Staff (school treasurers), School Committee, and Headmaster. This study validates technical feasibility within this specific institutional context without overclaiming broad generalization. Crucially, the system is scoped exclusively to post-entry tamper detection: once a transaction is recorded, it cannot be silently altered by a rogue DBA. Nevertheless, it operates as a post-hoc defense and does not prevent fraudulent data entered originally by authorized users — a recognized GIGO limitation. To ensure transparency, the architecture includes a public verification portal accessible to any stakeholder with the portal URL that allows authorized stakeholders (Super Administrator, Headmaster, School Committee, external auditors) to independently verify cryptographic integrity against the on-chain Merkle Root. Broader public access (e.g., parents) remains future work.

To address this gap, this study proposes a Hybrid Blockchain architecture paired with Keccak-256 Merkle Tree Aggregation. Transaction data is processed locally in a MySQL database (off-chain) for privacy and performance, while only the daily Merkle Root is anchored to the public Ethereum Sepolia network (on-chain). Architectural novelty is further reinforced by replacing destructive CRUD operations with a forensic accounting-compliant Reversal Entry protocol.

This approach targets a 99.9% reduction in blockchain transaction costs while maintaining a robust tamper-evident audit trail. Section 2 details the system architecture and algorithms; Section 3 presents empirical evaluations; and Section 4 concludes with limitations and future directions.

2. METHODS

2.1. Research Design

This study adopts the Design Science Research Methodology (DSRM) framework, widely recognized and validated as an ideal instrument for constructing applied blockchain architectures [11]. To meet contemporary scientific standards, the conceptual design in

this study is cross-validated using the Experimental Evaluation Framework [12]. This dual methodological cycle ensures that the proposed Hybrid Blockchain architecture not only operates functionally in a production environment (MAN 1 Kota Semarang) but is also quantitatively evaluated for resilience. Procedurally, this research follows six systematic DSRM stages [12]. The mapping of these phases is detailed in Table 1.

Table 1. DSRM Phase Mapping to Research Activities

DSRM Phase	Research Activities	Outputs / Deliverables
1. Problem Identification	Analyzing school financial system vulnerabilities to insider threats and database-level manipulation by DBAs.	Problem formulation and research gaps
2. Objective Definition	Defining three primary objectives: (a) tamper-proof integrity, (b) cost-efficient operation, and (c) forensic audit trail capability.	Research objectives and evaluation metrics
3. Design & Development	Designing a 4-layer architecture: Local Hash Chain, Merkle Tree Aggregation, Ethereum Anchoring, and Append-Only Reversal Entry.	Integrated system artifact.
4. Demonstration	Deployment in a production environment at MAN 1 Kota Semarang.	Operational system in a real-world environment.
5. Evaluation	Conducting four quantitative experiments: Tamper Detection, Performance Benchmark, Cost Simulation, and Comparative Analysis.	Measured quantitative data.
6. Communication	Documenting findings and submitting to a peer-reviewed journal.	Research publication

2.2. System Architecture

By default, blockchain networks are isolated and face scalability challenges alongside privacy regulation conflicts when storing large datasets [13]. To overcome these hurdles, this research implements a Hybrid Blockchain architecture to segregate on-chain and off-chain storage [13], [14]. Sensitive financial data is isolated locally in a relational MySQL

database (off-chain) to ensure institutional privacy compliance. Previous research [10] utilized smart contracts on the Ethereum network to record academic achievements on-chain, thereby preventing external modifications. As an architectural advancement of this approach, the proposed system introduces a Hybrid Blockchain model that integrates Keccak-256 Merkle Tree compression to minimize gas fees and optimize on-chain storage specifically for school financial transaction logs.

This segregation ensures that the public Ethereum network (on-chain) only stores daily cryptographic fingerprints in the form of a Merkle Root, as shown in Figure 1. This strategy reduces the blockchain storage burden by up to 74.8% [13], while accommodating data privacy principles mandated by GDPR and the Indonesian Personal Data Protection Law (UU PDP). The transaction payload contains only pseudonymized user identifiers (UUID), financial amounts, and categories — without direct student PII. If a deletion request is executed, the user PII is scrubbed while the transaction hash chain remains cryptographically unbroken, ensuring regulatory compliance without compromising forensic verification integrity. Unlike conventional hybrid architectures that rely on complex blockchain bridges [13] or separate sidechains, this system eliminates server-side complexity by interacting directly with the public Ethereum Sepolia network via a public RPC endpoint.

To prevent a DBA from redirecting the smart contract address, the production environment implements strict OS-level access controls: configuration files are read-only, and critical variables are encrypted using Laravel's configuration caching and injected via system-level environment variables, preventing unauthorized modification of anchoring parameters. The system is comprehensively structured around four integrated layers:

1) Presentation Layer: A web-based Laravel interface for cashier operations and a public verification portal.
2) Application Layer: The business logic engine that validates income and expenditure records, calculates local integrity, processes the Merkle Tree algorithm, and triggers external anchoring scripts.
3) Data Layer (Off-Chain): A private relational database (MySQL) storing complete transaction data and cryptographic metadata to comply with internal audit privacy constraints.

4) Verification Layer (On-Chain): A smart contract deployed on the public Ethereum Sepolia network, acting as an immutable, non-repudiation security anchor.

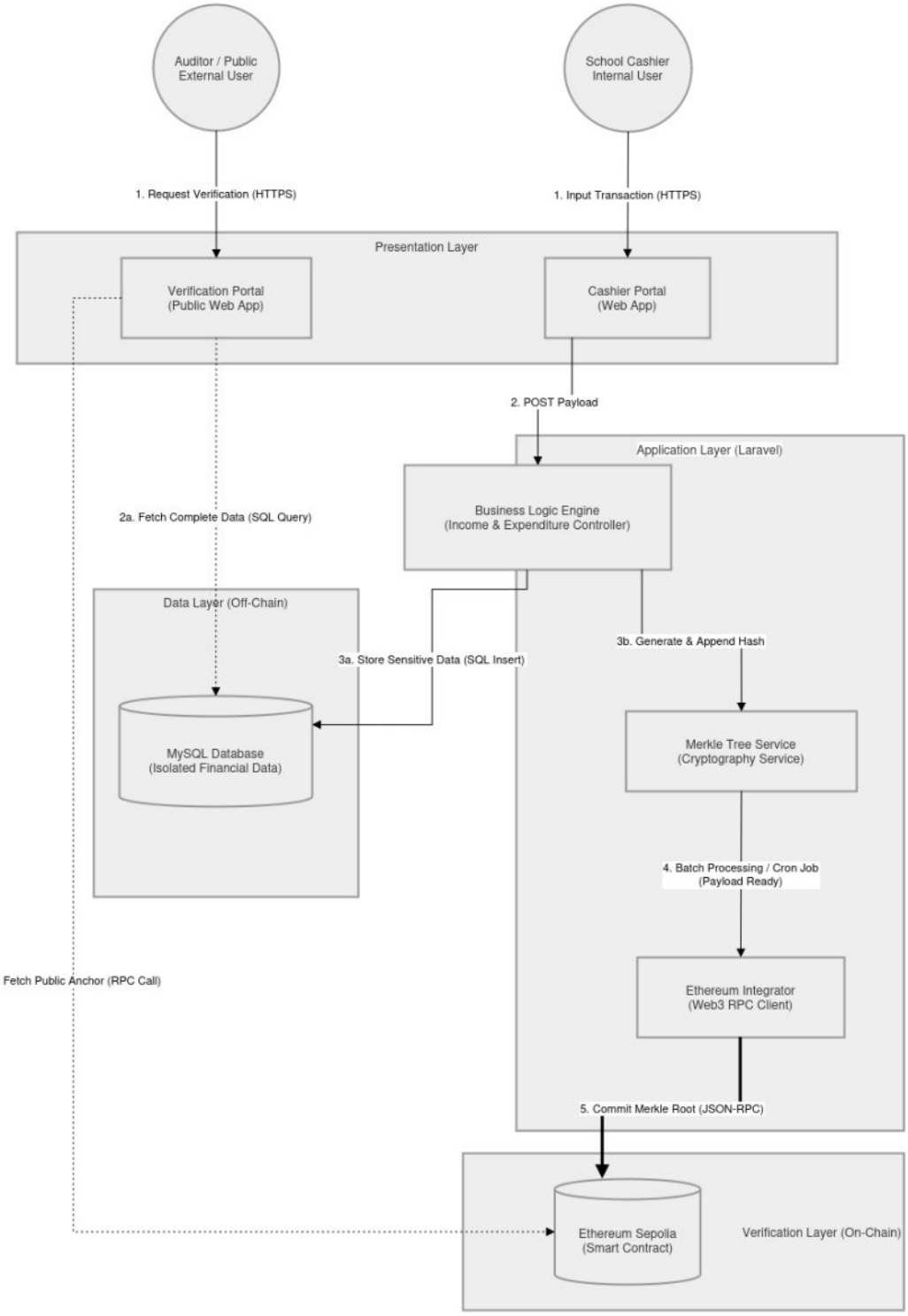

Figure 1. A four-layer Hybrid Blockchain system architecture: Presentation, Application, Data (Off-Chain), and Verification (On-Chain).

2.3. Algorithm Design

2.3.1. Layer 1 Cryptographic Local Hash Chain

The first layer implements a Recursive Transaction Hash Chain (RTHC) to guarantee forward integrity and strict temporal ordering [15]. Each transaction is cryptographically linked using the following hash function as shown in Equation 1.

$$H_i = \text{SHA-256}(N_i \parallel H_{i-1} \parallel payload_i) \quad (1)$$

To eliminate ambiguity, the exact concatenation sequence for the hash input is strictly ordered as follows: the 32-byte CSPRNG nonce (N_i) followed by the previous transaction hash (H_{i-1}), and finally the payload consisting of the financial amount, timestamp, user ID, and category. This deterministic configuration secures the validity of the transaction log. The hash computation process for each new transaction is detailed in Algorithm 1.

Algorithm 1: Recursive Transaction Hash Chain Construction

Input: Transaction data D = {amount, date, user_id, category}, Previous transaction hash H_{i-1}
Output: Current transaction hash H_i

```
N_i ← CSPRNG(32 bytes)
payload_i ← UTF8_ENCODE(
      D.amount + ":" + D.date + ":" + D.user_id + ":" + D.category
)
raw_input ← N_i ‖ H_{i-1} ‖ payload_i
H_i ← SHA-256(raw_input)
STORE(record, H_i, N_i)
RETURN H_i
```

Prior to hashing, all transaction fields are serialized as normalized UTF-8 strings using a colon-delimited canonical format: `"amount:date:user_id:category"` (e.g., `"500000.00:2025-10-15T08:30:00Z:usr_abc123:income"`). This canonical representation ensures that identical transaction data always produces the same hash digest, regardless of database encoding or application-layer formatting differences [16], [17]. For the genesis transaction (i = 1), the previous hash reference H_0 (the Genesis Hash) is initialized as a deterministic constant: $\text{H}_0 = \text{SHA-256}(\text{"GENESIS\_HASH"})$. This hardcoded initialization string ensures reproducibility across system restarts and eliminates dependency on external randomness for the chain origin.

2.3.2. Layer 2 Merkle Tree Aggregation

To mitigate computational and financial bottlenecks on-chain, thousands of daily local transaction hashes are compressed into a single Merkle Tree [18]. The construction of this tree is defined in Equation 2.

$$H_{\text{parent},i} = \text{Keccak-256}(H_a \parallel H_b) \tag{2}$$

where $(H_a, H_b) = \text{sort}(H_{\text{left}}, H_{\text{right}})$, and $\parallel$ denotes the concatenation of the lexicographically sorted child node hashes.

A fundamental architectural choice at this layer is the transition to the Keccak-256 algorithm [19] to comply natively with the Ethereum Virtual Machine (EVM). Computational complexity correlates directly with gas usage; employing the EVM's native hash is crucial to minimizing operational transaction fees [20]. Furthermore, Keccak-256 provides strong collision resistance and immunity to length extension attacks common in older SHA families [21]. This implementation utilizes lexicographical sorting of child nodes to ensure deterministic branch hierarchies. When the number of leaf nodes is odd, the last unpaired node is duplicated to form a complete binary pair before hashing [22]. Child nodes are sorted lexicographically by their lowercase hexadecimal string representation before concatenation, ensuring deterministic and reproducible tree construction regardless of insertion order.

2.3.3. Layer 3 Ethereum Anchoring

As a final security layer, the Merkle Root is anchored to a decentralized smart contract to guarantee non-repudiation [23]. The smart contract provides a `storeRoot(bytes32)` function that records the Merkle Root permanently with block timestamps, consuming approximately 48,000 gas units (single SSTORE operation). The public verification portal verifies integrity by recalculating the local Merkle Root and comparing it with the on-chain record via the Etherscan API [24]. The automated anchoring process utilizes a dedicated Ethereum wallet with restricted gas quotas; the private key is injected as an encrypted environment variable, and the smart contract enforces an ACL via the `onlyOwner` modifier, restricting execution to the authorized wallet. Mainnet deployments should adopt encrypted key management services (KMS) for enhanced security.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Crucially, this architecture inherently neutralizes advanced insider threats where a privileged DBA might attempt to recalculate and rewrite the entire local hash chain after modifying past records. Even if a DBA successfully alters a record and recalculates all subsequent local hashes to produce a new Merkle Root, this newly generated root will permanently mismatch the original Merkle Root already anchored on the public Ethereum network. Because the on-chain data is immutable and transparent, any local recalculation is instantly exposed during the verification process.

To operationalize this detection, the system employs a specific verification protocol, detailed in Algorithm 2. This process detects unauthorized deletions, injections, or alterations by comparing the locally reconstructed Merkle Root against the on-chain counterpart.

Algorithm 2: Cryptographic Verification and Tamper Detection
Input : *local_records* for Day D, On-chain Merkle Root ($Root_{chain}$) for Day D
Output : Boolean (True if intact, False if tampered)

```
local_records ← FETCH_ALL_TRANSACTIONS(Day D)
local_hashes ← EXTRACT_HASHES(local_records)
// Phase 1: Verify per-record hash chain integrity (Layer 1)
FOR i FROM 1 TO LENGTH(local_records):
    payload ← UTF8_ENCODE(record[i].amount + ":"
        + record[i].date + ":" + record[i].user_id
        + ":" + record[i].category)
    calculated ← SHA-256(record[i].nonce
        || record[i].prev_hash || payload)
    IF calculated ≠ local_hashes[i] THEN
        RETURN False  // Hash mismatch: tampering detected
    END IF
END FOR
// Phase 2: Reconstruct Merkle Root and compare (Layer 2-3)
Root_local ← BUILD_MERKLE_TREE(local_hashes)
IF Root_local = Root_chain THEN
    RETURN True   // Data intact
ELSE
    RETURN False  // Recalculation or deletion detected
END IF
```

The verification architecture operates across three tiers: Layer 1 hash-chain validation executes synchronously upon each transaction insertion, providing real-time tamper detection; Layers 2 and 3 Merkle Root reconstruction and on-chain comparison execute on a scheduled daily basis following automated anchoring. The complete end-to-end verification protocol may additionally be invoked on-demand by authorized stakeholders

including the Super Administrator, Headmaster, and external auditors through the public verification portal for immediate forensic audit purposes.

2.3.4. Append-Only Void Mechanism (Reversal Entry)

Digital ledger optimization has been shown empirically to improve transaction throughput and reporting accuracy [25]. Based on this principle, system operations eliminate destructive CRUD actions (specifically DELETE and UPDATE) and replace them with the fundamental accounting concept of a Reversal Entry. Input errors are offset using a negative balancing entry without breaking the sequential continuity of the Hash Chain, thereby minimizing reconciliation discrepancies [25].

Mechanically, when an operator identifies an entry error (e.g., an input of Rp 500,000 instead of Rp 50,000), the system automatically generates a negative balancing entry (-Rp 500,000) and computes a new hash linking it to the previous hash chain. Subsequently, a correction entry with the correct amount (Rp 50,000) is recorded as a separate transaction. This mechanism ensures that the complete error-and-correction chronology is transparently recorded within the hash chain, enabling auditors to reconstruct the transaction history without losing the audit trail.

2.3.5. Experimental Design

To ensure methodological rigor while protecting institutional operations, the deployment boundaries were strictly delineated. The frontend application and local database logging mechanisms were deployed in a production environment at MAN 1 Semarang to validate practical feasibility. Conversely, to avoid risking real operational data, all destructive evaluations, specifically the Tamper Detection (E1) and Performance Benchmarks (E2), were executed in an isolated laboratory environment orchestrated via Docker containers (Laravel Sail) on a local machine (Intel Core i5, 8GB RAM). The development stack utilizes Ubuntu 25.10, Docker 29.x (Laravel Sail), Laravel 8.x (PHP 8.0), MySQL 8.0, Solidity 0.8.x, and web3.php 0.1.x. Ethereum Sepolia anchoring was performed as a controlled deployment simulation within the laboratory environment. During Sepolia testing, private keys for the wallet were stored in the application's encrypted .env file, which is acceptable for testnet environments with no real monetary value. For production Mainnet or Layer-2 deployments, private keys must be managed through a dedicated Key Management Service (KMS) such as AWS KMS or HashiCorp Vault, with hardware security module (HSM)

integration to prevent key extraction. The DSRM evaluation phase is executed through four empirical tests mapped in Table 2.

Table 2. Experimental Design and Evaluation Metrics

Experiment	Objective	Test Variable	Evaluation Metrics
E1: Tamper Detection	Measure internal attack detection capabilities.	5 attack scenarios + 5 false-positive scenarios.	Detection Rate (%) & Latency (ms).
E2: Performance Benchmark	Measure server computational overhead.	Batch sizes: 1K, 5K, 10K (10 iterations).	Mean ± SD (s), Overhead (%)
E3: Cost Simulation	Compare 30-day operational costs.	Pure Blockchain vs. Hybrid Merkle.	Cost Efficiency (%).
E4: Comparative Analysis	Evaluate system against alternatives.	10 criteria × 4 technologies.	Total Score (Max. 50).

1) **Tamper Detection Test:** The E1 evaluation is structured into two complementary sub-evaluations. The first sub-evaluation (True-Positive Detection) executes direct SQL query injections bypassing the ORM across five critical attack scenarios: financial amount manipulation (surgical edit), timestamp manipulation (date tampering), cryptographic metadata insertion (chain injection), deletion of records (silent deletion), and transaction category manipulation (category swap). The second sub-evaluation (False-Positive Resistance) assesses whether the system correctly classifies five diverse legitimate operational scenarios as intact, without triggering spurious alerts: (a) a standard income entry, (b) a standard expenditure entry, (c) a legitimate Reversal Entry representing a negative balancing correction, (d) a correction entry immediately following a Reversal Entry, and (e) a high-volume sequential batch of 10 consecutive legitimate transactions. This two-part design validates both the system's resilience against insider threat vectors [26], [27] and its precision in avoiding false alarms under heterogeneous normal operations.

2) **Performance Benchmark:** Latency overhead and throughput (Transactions Per Second) are measured during simulated batch insertions of 1,000, 5,000, and 10,000 transactions to evaluate throughput and response times [6], [27].
3) **Cost Simulation:** Public vs. hybrid operational costs are evaluated using empirical gas logs (TX0x78c7..a02c) to validate hybrid on-chain/off-chain performance [28], [29].
4) **Comparative Analysis:** The proposed architecture (Hybrid Merkle) is evaluated against traditional logs, pure public blockchains, and permissioned blockchains using a 10-dimension matrix [27].

For the quantitative evaluations conducted in the Docker laboratory, the experiments utilized a hybrid dataset comprising 10,000 transactions. To simulate realistic financial patterns while maintaining high-volume stress conditions, this dataset consisted of 10% (1,000) actual historical production transactions and 90% (9,000) synthetically generated records. Formal ethical permission to conduct this research was granted by the Headmaster of MAN 1 Kota Semarang through an official institutional research authorization letter. To ensure stringent data protection compliance, all historical data utilized in the laboratory environment was fully anonymized; Personally Identifiable Information (PII) such as student names and specific activity descriptions were completely scrubbed prior to experimentation. This protocol guarantees that no sensitive institutional data was exposed to experimental manipulation or public blockchain anchoring.

Quantitative data analysis for all experiments is conducted using descriptive statistics. The sample mean ($\bar{x}$) is calculated using Equation (3), and variance is reported using the sample standard deviation (s) with Bessel's correction in Equation (4).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{4}$$

Here n represents the number of test iterations ($n = 10$ iterations per scenario). This follows methodological precedents in blockchain performance evaluation [27], which recommend a minimum of 10 runs to obtain representative descriptive statistics in controlled environments.

3. RESULTS AND DISCUSSION

3.1. Tamper Detection Integrity

The first experiment (Tamper Detection Test) was orchestrated to validate the resilience of the local security layer (Layer 1) against root-level database manipulation (insider threats). The E1 evaluation was structured into two complementary sub-evaluations: True-Positive Detection (S1–S5) and False-Positive Resistance (S0a–S0e), each executed across $n = 10$ iterations per scenario. The detection latency represents the time elapsed from query execution to system response (hash mismatch confirmation or chain gap detection).

Table 3. Tamper Detection Test Results ($n = 10$ iterations per scenario)

ID	Scenario	Category	Status	Mean Latency (ms)	Std Dev (ms)	95% CI (ms)	Anomaly Mapping
S0a	Standard Income Entry	Normal Operation	Normal	1.1119	± 0.3087	[0.891, 1.333]	None (chain intact)
S0b	Standard Expenditure Entry	Normal Operation	Normal	1.3876	± 0.3751	[1.119, 1.656]	None (chain intact)
S0c	Legitimate Reversal Entry (negative correction)	Normal Operation	Normal	1.2120	± 0.4814	[0.868, 1.556]	None (chain intact)
S0d	Correction Entry post-Reversal	Normal Operation	Normal	1.2819	± 0.2236	[1.122, 1.442]	None (chain intact)
S0e	High-volume sequential batch (10 consecutive tx)	Normal Operation	Normal	1.3423	± 0.4475	[1.022, 1.662]	None (chain intact)

ID	Scenario	Category	Status	Mean Latency (ms)	Std Dev (ms)	95% CI (ms)	Anomaly Mapping
S1	Surgical Edit (Amount Manipulation)	Attack	Detected	1.1503	± 0.2236	[0.990, 1.310]	Hash mismatch
S2	Date Tampering (Temporal Anomaly)	Attack	Detected	1.2000	± 0.0851	[1.139, 1.261]	Hash mismatch
S3	Chain Injection (Cryptographic Linkage Break)	Attack	Detected	2.2649	± 0.3088	[2.044, 2.486]	Linkage broken
S4	Silent Deletion (Database Entropy)	Attack	Detected	2.2980	± 0.2690	[2.106, 2.491]	Chain gap
S5	Category Swap (Classification Manipulation)	Attack	Detected	1.2882	± 0.2000	[1.145, 1.431]	Hash mismatch

The false-positive sub-evaluation (S0a–S0e) demonstrates that the system correctly classified all five heterogeneous legitimate operational scenarios as intact, yielding a 0.0% false-positive rate across diverse normal transaction types including standard income and expenditure entries, legitimate Reversal Entries, and high-volume sequential batches. Critically, the system accurately distinguished a legitimate Reversal Entry (S0c), which appends a negative-value hash into the chain, from a malicious silent deletion (S4), which creates a gap in chain linkage. This distinction validates that the append-only Reversal Entry mechanism does not interfere with cryptographic integrity verification. These results provide stronger empirical evidence for false-positive resistance compared to a single sanity-check baseline, directly addressing the need for heterogeneous operational validation.

The true-positive sub-evaluation (S1–S5) confirms a 100% attack detection rate across all five insider threat scenarios. Notably, Chain Injection (S3) and Silent Deletion (S4) exhibit higher mean latencies (2.26–2.30 ms) compared to field-modification attacks (S1, S2, S5:

1.15–1.29 ms), attributable to the additional chain linkage traversal required for gap detection. All detection latencies remain well within sub-3-millisecond thresholds. The system logged a mean response latency of under 2.30 ms across all attack scenarios, indicating high computational stability. This 100% detection rate confirms that the RTHC implementation successfully utilizes SHA-256 collision resistance in a production-like environment, complementing existing literature that rarely validates resistance against direct raw SQL execution [27].

More importantly, the detection accuracy in S3 and S4 demonstrates that the RTHC algorithm secures not only individual record contents but also the chronological integrity of the entire ledger. Two latency metrics are reported in this study: tamper-detection latency (Table 3, mean < 2.30 ms) measures hash recalculation and comparison time for existing records, while insertion latency (Table 4, mean 3.45 ms/tx) measures full RTHC hash computation, nonce generation, and chain linkage for new records. Both confirm low-millisecond performance well within acceptable interactive thresholds. The 95% confidence intervals (t-distribution, df = 9) for all Table 3 scenarios fall within ±0.06–0.34 ms of the reported means, confirming low measurement variance despite the modest sample size. While 10 runs represent an acceptable minimum, future studies should use larger sample sizes to strengthen external validity.

3.2. Computational Performance

The second experiment (Performance Benchmark) quantified the computational overhead imposed on the server by the recursive cryptographic hashing layer. The testing methodology compared conventional data insertion latencies (Baseline) against RTHC-augmented insertions across three transaction volumes: 1,000, 5,000, and 10,000 entries. The compiled data from 10 test runs per volume is summarized in Table 4.

Table 4. Latency Comparison: Baseline vs. Hash Chain ($n = 10$ runs per volume)

Volume (n)	Baseline (s)	Hash Chain (s)	Overhead (%)	Latency/Tx (ms)	95% CI (ms)	Throughput (TPS)
1,000	1.1255 ± 0.1031	3.4543 ± 0.0653	+206.92%	3.4543	[3.4076, 3.5010]	289
5,000	5.3219 ± 0.0877	17.0676 ± 0.1135	+220.71%	3.4135	[3.3973, 3.4297]	293

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Volume (n)	Baseline (s)	Hash Chain (s)	Overhead (%)	Latency/Tx (ms)	95% CI (ms)	Throughput (TPS)
10,000	10.5547 ± 0.1193	34.4994 ± 1.3548	+226.86%	3.4499	[3.3530, 3.5468]	293

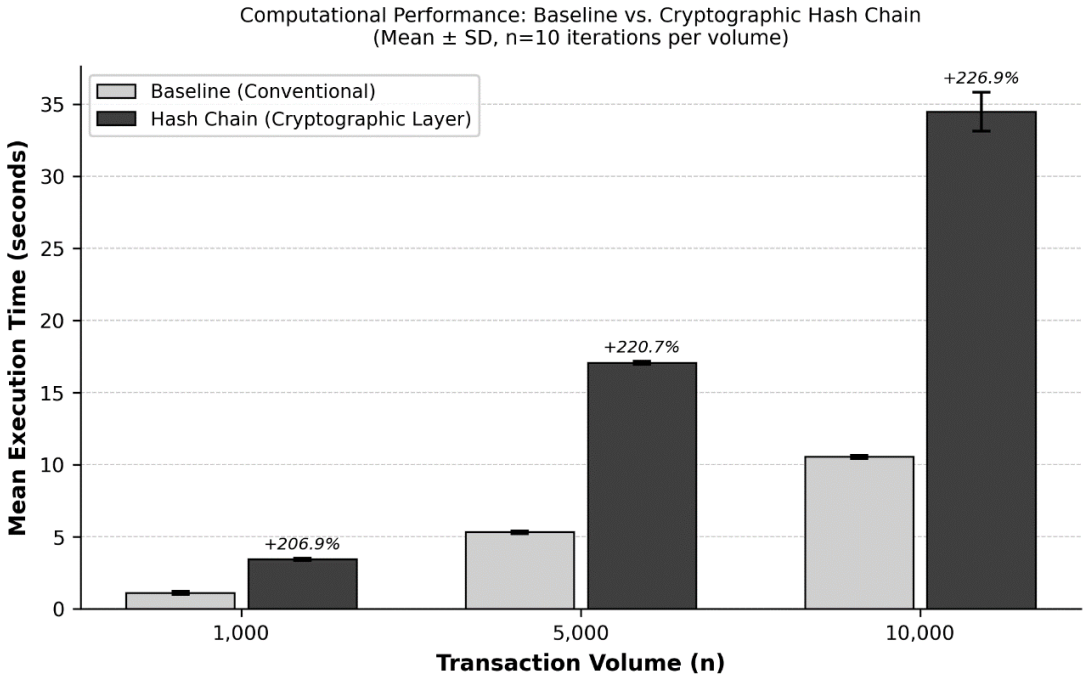

Figure 3. Comparison of computational performance: Baseline vs. Cryptographic Hash Chain (Mean ± SD, n=10 iterations per volume). The error bars represent ± 1 SD; the baseline SD value (≤ 0.12 s) is barely visible on this scale.

Integrating the cryptographic layer introduces a relative overhead of 206.92%–226.86%; however, absolute latency remained consistent at 3.41–3.45 ms per transaction across all scales, yielding a stable throughput of 289–293 TPS. The 95% confidence intervals for per-transaction latency remain narrow across all volumes (±0.02–0.10 ms), with the widest interval occurring at the 10,000-entry scale — consistent with the higher variance attributable to virtualized environmental factors discussed below. The 10,000 entry volume represents a simulated batch insertion to stress-test the algorithm, not typical daily traffic at MAN 1 Kota Semarang. Since MAN 1 Kota Semarang processes only 20–50 transactions daily, the system's throughput of 289–293 TPS surpasses this requirement by a factor of 5,000, eliminating cryptographic overhead as a performance bottleneck. The 289–293 TPS throughput reflects full local processing (database insertion, SHA-256 hashing, nonce generation, and chaining). Ethereum anchoring latency is excluded, as it

operates as a separate daily batch process. With a mean latency of 3.45 ms, the architecture operates well below the 75 ms interactive usability threshold [30], supporting a responsive administrative experience [31]. The higher deviance at 10,000 entries (σ = 1.35 s) is attributable to virtualized environmental factors such as Docker garbage collection or disk I/O contention.

3.3. Gas Cost Scaling & Economic Feasibility

The third experiment analyzed the economic feasibility of the Hybrid Blockchain design compared to a pure public blockchain approach. The extrapolative calculations are based on empirical daily anchoring transactions on the public Ethereum Sepolia testnet, which recorded a gas consumption of 48,295 units. The valuation is calibrated using a standard Sepolia gas price of 30 Gwei and the ETH/IDR exchange rate recorded via the CoinGecko API at the time of evaluation, which stood at Rp 40,079,419 per ETH. The pure blockchain cost assumes individual, unbatched on-chain calls (48,295 gas for full hash calldata). As highlighted in recent studies, relying on unbatched transactions on pure public blockchains is increasingly prohibitive due to escalating gas costs and network congestion [32]. This worst-case baseline ensures a deliberately conservative comparison favoring the hybrid architecture. The 30-day operational cost projection is detailed in Table 5.

Table 5. 30-Day Operational Cost Projection (at 1,000 tx/day)

Architecture	Total Call	Total Gas Units	Cost (ETH)	Cost (IDR)
Pure Blockchain (A)	30,000	1,448,850,000	43.46550000	Rp 1,742,072,465
Hybrid Merkle Chain (B)	30	1,448,850	0.04346550	Rp 1,742,072

3.3.1. Sensitivity Analysis

To assess the financial robustness of the proposed architecture under real-world market volatility, a sensitivity analysis was conducted across three key parameters: Ethereum gas price, ETH/IDR exchange rate, and daily transaction volume. All projections apply the empirically measured gas consumption of 48,295 units per anchoring transaction, maintaining one anchoring per day (30 per month) for the hybrid model.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 5a. Hybrid Monthly Cost Sensitivity: Gas Price × ETH/IDR Exchange Rate

Gas Price (Gwei)	ETH = Rp 20,000,000	ETH = Rp 40,000,000	ETH = Rp 60,000,000	ETH = Rp 100,000,000
10	Rp 289,770	Rp 579,540	Rp 869,310	Rp 1,448,850
30 *(baseline)*	Rp 869,310	Rp 1,738,620	Rp 2,607,930	Rp 4,346,550
50	Rp 1,448,850	Rp 2,897,700	Rp 4,346,550	Rp 7,244,250
100	Rp 2,897,700	Rp 5,795,400	Rp 8,693,100	Rp 14,488,500

Table 5b. Cost Efficiency by Daily Transaction Volume (Gas Price = 30 Gwei; ETH = Rp 40,000,000)

Tx/Day	Hybrid Monthly Cost	Pure Blockchain Monthly Cost	Cost Efficiency
100	Rp 1,738,620	Rp 173,862,000	99.00%
500	Rp 1,738,620	Rp 869,310,000	99.80%
1,000 *(baseline)*	Rp 1,738,620	Rp 1,738,620,000	99.90%
2,000	Rp 1,738,620	Rp 3,477,240,000	99.95%
5,000	Rp 1,738,620	Rp 8,693,100,000	99.98%

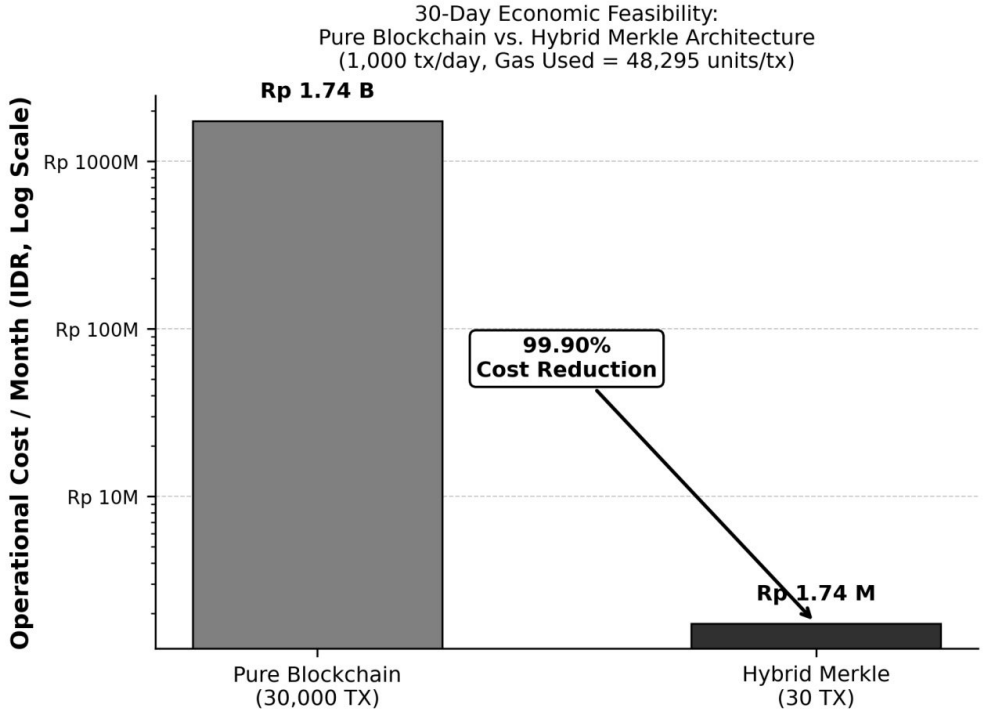

Figure 4. Comparison of 30-day operational costs between Pure Blockchain and Hybrid Merkle architectures (1,000 transactions per day, Gas Used = 48,295 units per transaction, logarithmic scale). Cost reduction: 99.90%.

The sensitivity results across Table 5a and 5b reveal three key findings. First, even under extreme market conditions — a gas price of 100 Gwei combined with an ETH/IDR rate of Rp 100,000,000 — the monthly operational cost remains Rp 14,488,500, which constitutes less than 0.84% of the equivalent pure blockchain cost at the same parameters. Second, cost efficiency scales positively with transaction volume: even at a low daily volume of 100 transactions, the hybrid architecture still yields a 99.00% cost reduction over a pure public blockchain. Third, the hybrid monthly cost is invariant to transaction volume — it remains fixed at approximately Rp 1.74 million per month regardless of whether the institution processes 100 or 5,000 transactions daily, since only one Merkle Root is anchored per day. These findings collectively suggest that the hybrid architecture remains financially viable for medium-scale educational institutions across a wide range of cryptocurrency market conditions and transaction volumes. This 99.90% cost reduction is a direct architectural benefit of the hybrid design. By anchoring only the daily cryptographic Merkle Root, the system achieves asymmetric blockchain security at a fraction of the cost of pure on-chain storage [28].

3.4. Comparative Architectural Analysis

The final experiment evaluated the structural positioning of the Hybrid Merkle design against three dominant data security paradigms: Traditional Audit Logs, Pure Public Blockchain (Ethereum), and Permissioned Blockchain (Hyperledger Fabric). The evaluation uses a multi-criteria 10-dimension matrix rated on an ordinal scale of 1–5. The criteria are derived from three sources (a) security dimensions in blockchain integrity literature [21], [33], (b) operational barriers identified in past studies [5], [8], [28], and (c) empirical findings from E1–E3. The score matrix is summarized in Table 6.

Table 6. Multi-Criteria Comparative Matrix (Scale 1–5)

Evaluation Criteria	Traditional Log	Pure Blockchain	Fabric	Hybrid
Tamper-Proof	2	5	5	**5**
Operational Cost	5	1	3	**5**
Transaction Latency	5	1	3	**5**
Data Privacy (PII)	5	1	5	**5**

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Evaluation Criteria	Traditional Log	Pure Blockchain	Fabric	Hybrid
Permanent Audit Trail	1	5	5	**5**
Reversal Entry	5	1	2	**5**
Minimal Infrastructure	5	2	1	**5**
Public Verification	1	5	1	**5**
Time-Travel Prevention	1	5	5	**5**
Implementation Complexity	5	2	1	**4**
Total Score (/50)	**35**	**28**	**31**	**49**

While Table 6 suggests the superiority of the proposed architecture (49/50), these researcher-assigned comparative scores are inherently subjective, based solely on the MAN 1 case study constraints, and do not represent an absolute universal benchmark.

3.5. Discussion

The empirical findings indicate that the proposed Hybrid Blockchain architecture bridges the gap between expensive public blockchains and highly complex permissioned consortia. Unlike earlier academic ledger systems focused on general data manipulation, the Reversal Entry protocol specifically neutralizes database-level insider attacks while maintaining a forensically complete audit trail. While alternatives like append-only tables, immutable triggers, or Temporal Tables provide basic auditing, they remain fundamentally vulnerable to a privileged DBA who can disable constraints or truncate tables at the database engine level. The proposed Hybrid Blockchain architecture addresses this gap by externalizing the cryptographic verification anchor to a public, decentralized Ethereum ledger, ensuring that even full database-level control cannot falsify evidence of historical tampering without producing a detectable mismatch against the on-chain Merkle Root. Existing methods like WORM storage lack dynamic querying and affordability, while triggers and Temporal Tables suffer from single-administrator vulnerabilities.

Despite these contributions, three limitations must be acknowledged. First, the system operates as a post-hoc defense: if fraudulent data is submitted through the official frontend at the point of entry, it will be hashed and secured as-is (GIGO vulnerability). Second, daily anchoring creates a 24-hour window reliant solely on local Layer 1 verification. This deliberate cost-security trade-off avoids the 24× gas overhead of hourly anchoring, ensuring sufficient forensic assurance for MAN 1 Kota Semarang's low transaction volume (20–50 daily). Third, production deployments necessitate robust operational protocols: (a) continuous wallet balance monitoring to prevent anchoring failure due to insufficient gas funds, (b) automated gas price monitoring, (c) documented key rotation, (d) multi-signature contract administration with a defined smart contract ownership transfer procedure in case of administrator turnover, (e) redundant RPC endpoints (e.g., Infura/Alchemy), and (f) comprehensive disaster recovery plans including cold-wallet backups.

4. CONCLUSION

This study validates a Hybrid Blockchain architecture utilizing Merkle Tree Aggregation to secure financial data integrity within a controlled case-study environment at MAN 1 Kota Semarang. The system achieved 100% tamper detection, 289–293 TPS (3.45 ms/tx) throughput, and a 99.90% cost reduction via 1,000:1 batch compression compared to pure public blockchains. Mainnet-equivalent projections confirm its continuous financial viability (Rp 289,770–14,488,500 monthly) across volatile gas prices (10–100 Gwei) and ETH/IDR exchange rates. While effectively resolving local security-cost-complexity trade-offs, this architecture operates as a post-hoc defense susceptible to initial data-entry (GIGO) vulnerabilities. Future research must address these operational boundaries by targeting: (1) Mainnet or Layer-2 deployments, (2) Zero-Knowledge Proof (ZKP) integration for privacy, (3) Delphi-method expert validation to eliminate researcher bias, and (4) formal User Acceptance Testing (UAT) with institutional stakeholders.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to MAN 1 Kota Semarang for granting permission to utilize their system and data for this research. Their support and

cooperation were instrumental in the successful implementation and empirical evaluation of the proposed Hybrid Blockchain architecture.

REFERENCES

[1] M. A. Almaiah, L. M. Saqr, L. A. Al-rawwash, L. A. Altellawi, R. Al-ali, and O. Almomani, "Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems," *Comput. Mater. Contin.*, vol. 81, no. 2, pp. 3189–3220, 2024, doi: 10.32604/cmc.2024.057673.

[2] A. Al-harrasi, A. K. Shaikh, and A. Al-badi, "Towards protecting organisations' data by preventing data theft by malicious insiders," *Int. J. Organ. Anal.*, vol. 31, no. 3, pp. 875–888, 2023, doi: 10.1108/IJOA-01-2021-2598.

[3] A. R. Marbut and P. D. Harms, "Fiends and Fools: A Narrative Review and Neo-socioanalytic Perspective on Personality and Insider Threats," *J. Bus. Psychol.*, vol. 39, no. 3, pp. 679–696, 2024, doi: 10.1007/s10869-023-09885-9.

[4] R. Zhao, M. Shoaib, V. T. Hoang, and W. U. Hassan, "Rethinking tamper-evident logging: A high-performance, co-designed auditing system," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, Taipei: Association for Computing Machinery, 2025, pp. 2624–2638, doi: 10.1145/3719027.3765024.

[5] R. Biswas, S. Jana, M. Pal, D. Chatterjee, K. Pal, and P. Dhar, "Data Integrity and Security Mechanisms in Cloud-Based Relational Databases," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 6, pp. 399–405, 2024, doi: 10.48175/IJARSCT-22561.

[6] M. M. Khan, F. S. Khan, M. Nadeem, T. H. Khan, S. Haider, and D. Daas, "Scalability and efficiency analysis of hyperledger fabric and private Ethereum in smart contract execution," *Computers*, vol. 14, no. 4, p. 132, 2025, doi: 10.3390/computers14040132.

[7] P. Shylaja and J. S. Jayasudha, "A Comprehensive Review of Blockchain and Smart Contracts: Foundations, Applications, and Technical Challenges," *Prem. J. Sci.*, vol. 15, p. 100257, 2026, doi: 10.70389/PJS.100257.

[8] E. Alaka, K. Abiodun, S. O. Jinadu, E. Igba, and V. N. Ezeh, "Data Integrity in Decentralized Financial Systems: A Model for Auditable, Automated Reconciliation Using Blockchain and AI," *Int. J. Manag. Commer. Innov.*, vol. 13, no. 1, pp. 136–158, 2025, doi: 10.5281/zenodo.15753099.

[9] J. Westphall and J. E. Martina, "Blockchain Privacy and Scalability in a Decentralized Validated Energy Trading Context with Hyperledger Fabric," *Sensors*, vol. 22, no. 12, p. 4585, 2022, doi: 10.3390/s22124585.

[10] S. Hadi, A. N. Putri, and P. A. Buana, "Selecting Achievement-Based Students Using Blockchain and AHP: Semarang University Case Study," *J. Inf. Syst. Informatics*, vol. 6, no. 4, pp. 2192–2206, 2024, doi: 10.51519/journalisi.v6i4.901.

[11] E. M. Alotaibi, H. Issa, and M. Codesso, "Blockchain-based conceptual model for enhanced transparency in government records: a design science research approach," *Int. J. Inf. Manag. Data Insights*, vol. 5, no. 1, p. 100304, 2025, doi: 10.1016/j.jjimei.2024.100304.

[12] F. Wen, "A Design Science Study of a Mobile Human Resource System for Internal Staffing and User Acceptance," *J. Cases Inf. Technol.*, vol. 27, no. 1, pp. 1–29, 2025, doi: 10.4018/JCIT.389148.

[13] K. Yeh, G. Yang, C. Butpheng, L. Lee, and Y. Liu, "A Secure Interoperability Management Scheme for Cross-Blockchain Transactions," *Symmetry (Basel)*, vol. 14, no. 12, p. 2473, 2022, doi: 10.3390/sym14122473.

[14] G. Mandinyenya and V. Malele, "A Hybrid Framework for Enhancing Privacy in Blockchain-Based Personal Data Sharing using Off-Chain Storage and Zero-Knowledge Proofs," *J. Inf. Syst. Informatics*, vol. 7, no. 2, pp. 1977–2005, 2025, doi: 10.51519/journalisi.v7i2.1119.

[15] S. V. K. Gummadi, "Recursive Transaction Hash Chains for Immutable Audit Trails in Mortgage Platforms," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 6, no. 4, pp. 49–54, 2025, doi: 10.63282/3050-9246.IJETCSIT-V6I4P107.

[16] L. Wei, F. Al-rashidi, and A. Krishnamurthy, "ChainGuard: A Blockchain- and IoT-Augmented Framework for Real-Time Database Integrity Assurance in Distributed Healthcare Information Systems," *DATAMIND*, vol. 4, no. 1, pp. 6–24, 2026, doi: 10.63646/datamind.2026.040102.

[17] N. R. Reddy, S. Suryadevara, and K. G. R. Reddy, "Quantum secured blockchain framework for enhancing post quantum data security," *Sci. Rep.*, vol. 15, no. 1, p. 31048, 2025, doi: 10.1038/s41598-025-16315-8.

[18] P. Dhiman, S. K. Henge, S. Singh, A. Kaur, P. Singh, and M. Hadabou, "Blockchain Merkle-Tree Ethereum Approach in Enterprise Multitenant Cloud Environment," *Comput. Mater. Contin.*, vol. 74, no. 2, pp. 3297–3313, 2023, doi: 10.32604/cmc.2023.030558.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[19] J. P. Lemayian, G. Gagnon, K. Zhang, and P. Giard, "HardVault: A Hybrid FPGA-Based Ethereum-Bitcoin Cold Wallet," *IEEE Trans. Very Large Scale Integr. Syst.*, 2026, doi: 10.1109/TVLSI.2026.3696577.

[20] A. S. Paramita and M. Tarigan, "Analysis of Gas Fee Patterns in Blockchain Transactions - A Case Study on Ethereum Smart Contracts," *J. Curr. Res. Blockchain*, vol. 2, no. 3, pp. 180–189, 2025, doi: 10.47738/jcrb.v2i3.41.

[21] C. F. Ikenga-Metuh and A. Yeboah-Ofori, "Blockchain security using confidentiality, integrity, and availability for secure communication," *Blockchains*, vol. 4, no. 1, p. 3, 2026, doi: 10.3390/blockchains4010003.

[22] 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*, vol. 26, p. 101193, 2024, doi: 10.1016/j.iot.2024.101193.

[23] C. Regueiro, I. Seco, B. Urquizu, and J. Mansell, "A Blockchain-Based Audit Trail Mechanism: Design and Implementation," *Algorithms*, vol. 14, no. 12, p. 341, 2021, doi: 10.3390/a14120341.

[24] W. Liu, J. Li, and N. Chen, "CBAATM: A Blockchain-AI Integrated Framework for Real-Time Anomaly Detection and Compliance Verification in Smart Accounting Information Systems," *Informatica*, vol. 49, no. 20, pp. 253–272, 2025, doi: 10.31449/inf.v49i20.10028.

[25] F. Khan and S. Chadni, "Digital Ledger Optimization Techniques for Enhancing Transaction Speed and Reporting Accuracy in Accounting," *Am. J. Sch. Res. Innov.*, vol. 1, no. 02, pp. 171–222, 2022, doi: 10.63125/33t06k57.

[26] U. Rauf, F. Mohsen, and Z. Wei, "A Taxonomic Classification of Insider Threats: Existing Techniques, Future Directions & Recommendations," *J. Cyber Secur. Mobil.*, vol. 12, no. 2, pp. 221–252, 2023, doi: 10.13052/jcsm2245-1439.1225.

[27] I. U. Akpara and O. V. Bamigwojo, "Secure Database Trigger and Stored-Procedure Design for Automated Compliance Logging in Multi-User Administrative Systems," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, pp. 974–999, 2023, doi: 10.32628/CSEIT25112793.

[28] B. Wang, R. Jiang, X. Pu, and H. Zhang, "An on-chain and off-chain collaborative data sharing and access control model for electronic medical records," *J. Supercomput.*, vol. 81, no. 2, p. 396, 2025, doi: 10.1007/s11227-024-06884-2.

[29] H. Su, S. Dong, and T. Zhang, "A Hybrid Blockchain-Based Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 11, pp. 17059–17072, 2024, doi: 10.1109/TVT.2024.3424786.

[30] S. Van Damme *et al.*, "Impact of Latency on QoE, Performance, and Collaboration in Interactive Multi-User Virtual Reality," *Appl. Sci.*, vol. 14, no. 6, p. 2290, 2024, doi: 10.3390/app14062290.

[31] K. Somei, K. Oshima, and T. Tsumugiwa, "Effects of Display Response Latency on Brain Activity During Device Operation," *IEEE Access*, vol. 11, pp. 34860–34869, 2023, doi: 10.1109/ACCESS.2023.3262658.

[32] S. Porkodi and D. Kesavaraja, "Escalating Gas Cost Optimization in Smart Contract," *Wirel. Pers. Commun.*, vol. 136, no. 1, pp. 35–59, 2024, doi: 10.1007/s11277-024-11066-7.

[33] J. Seol, J. Deuja, I. N. Park, C. Pu, and N. Park, "A Quantitative Study across CIA (Confidentiality, Integrity, Availability) Triad and Performance in Blockchain-Based Crypto-Space," in *2025 7th International Conference on Blockchain Computing and Applications (BCCA)*, Dubrovnik, Croatia, 2025, pp. 161–168, doi: 10.1109/BCCA66705.2025.11229817.