NFSU JOURNAL OF CYBER SECURITY & DIGITAL FORENSICS

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 4, Issue – 1, June 2025
E – ISSN – 2583-7559
International Conference on Role of Blockchain
in Digital Forensics and Cyber Security - 2025

Tamper-Proof Compliance Framework Using Hyperledger Fabric for Secure and Auditable Log Management

Tejas Mane
National Forensic Sciences University,
Gandhinagar, Gujarat, India,
tejasmane006@gmail.com

Dr. Ravirajsinh S. Vaghela
Assistant Professor,
School of Cyber Security & Digital Forensic,
National Forensic Sciences University,
Gandhinagar, Gujarat, India

***Abstract*—Guaranteeing compliance with regulations in enterprise security log management is an important challenge since legacy systems do not have immutability, auditability, and real-time verification. In this paper, a Tamper-Proof Compliance Framework based on Hyperledger Fabric, a permissioned blockchain, is introduced to securely harvest, verify, and store security logs from various SIEM (Security Information and Event Management) systems. The architecture combines JWT authentication, digital signatures, and Role-Based Access Control (RBAC) for guaranteeing log authenticity, integrity, and restricted access. All registered organizations work within a reserved Hyperledger Fabric channel, isolating data and independently validating compliance. Automated verification of compliance is achieved through the application of smart contracts (chaincode) that compare logs with most significant controls from GDPR, PCI DSS, and NIST CSF in order to have a dynamic compliance scoring system evaluating real-time adherence and issuing alerts for failures. Through decentralized architecture, this is improved in terms of security, openness, and regulatory compliance enforcement, ensuring a scalable and tamper-resistant compliance solution for contemporary enterprises.**

Keywords: Blockchain, Hyperledger Fabric, Compliance Monitoring, SIEM Integration, Tamper-Proof Logging

I. Introduction

Security Information and Event Management (SIEM) solutions are critical to enterprise security today by correlating and analyzing security logs to identify threats and deliver regulatory compliance. However, traditional SIEM architectures present severe challenges to ensuring the integrity and authenticity of log data. Privileged administrators can tamper with or remove logs, rendering these records unreliable for compliance audits and forensic analyses. Such vulnerabilities not only undermine effective security operations but also offer threats of non-compliance with regulations such as the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS), both of which require immutable audit trails for accessing sensitive information.

To tackle these challenges, the adoption of blockchain technology, specifically Hyperledger Fabric, presents a viable solution. Hyperledger Fabric is a permissioned blockchain platform tailored for enterprise applications, featuring characteristics like modularity, scalability, and support for confidential transactions. Organizations can attain tamper-resistant secure log storage by using Hyperledger Fabric, thereby ensuring that logged information cannot be altered or removed secretly. Such an immutability property significantly increases log reliability, making them more trustworthy for compliance audit and judicial judgments.

Additionally, Hyperledger Fabric architecture includes private data collections that allow organizations to exchange data confidentially within subsets of the network. Such a feature includes the ability to exchange cryptographic data between the qualified nodes securely without broadcasting to the whole network and thus maintaining confidentiality without compromising integrity. Such a system, if implemented, ensures not only security but also streamlines compliance procedures with less likelihood of regulatory penalties in addition to improving the general security posture of the organization.

This work proposes a new integration of Hyperledger Fabric with current SIEM systems for the enhancement of log management security and compliance. The major contributions of this work are:

1. **Immutable Log Storage:** By leveraging Hyperledger Fabric's blockchain technology, the system ensures that logs cannot be altered or

removed once they are placed on record, thereby keeping them intact and authentic.

2. **Improved Compliance Verification:** The system supports compliance with regulatory standards, including GDPR and PCI DSS, by providing verifiable and tamper-proof audit trails, thereby improving the verification process of compliance.

3. **Enhanced Legal Admissibility**: The unalterable nature of the blockchain-kept logs makes them more credible and admissible in court, and the evidence is guaranteed to be uncompromised.

4. **Seamless Integration with SIEM**: The proposed framework is built to integrate seamlessly with existing SIEM solutions, enabling organizations to expand their log management functions without having to redesign their current infrastructure.

II. Problem Statement

Traditional SIEM systems, as much as they are a security watch necessity, possess certain essential limitations that detract from their value in enforcing compliance and forensic analysis [8]:

1. **Susceptibility to Log Tampering:** High-level administrators have the ability to change or erase log records, either accidentally or on purpose, which contradicts the integrity of such logs [16]. As SIEMs do not natively provide tamper-proof storage, this is a serious impediment to being able to accurately identify security incidents or carry out effective investigations [2], [17].

2. **Compliance Verification Challenges:** The conventional SIEM solutions do not support immutable storage, and therefore it is challenging to prove compliance with regulatory standards such as GDPR and PCI DSS [4], [14]. Due to this, organisations will be forced to struggle to prove verifiable audit trails [1], [12], leaving them vulnerable to increased risks of non-compliance and ensuing legal or financial penalties [19].

3. **Admissibility in Courts of Law:** In courts of law, authenticity and integrity of logs are paramount. Logs that are tamper-prone or fail to adhere to documented chain-of-custody protocols could render them inadmissible in courts of law as evidence [5], [6]. This is a reality that nullifies forensic investigations and increases the likelihood that criminal activity will go unpunished.

III. Literature Review

Blockchain for Secure Log Management

The increasing regulatory complexity and the demand for tamper-evident logging have accelerated research into blockchain-based log management systems. Traditional Security Information and Event Management (SIEM) solutions often suffer from data tampering vulnerabilities, insufficient auditability, and delayed compliance verification processes [1]. Blockchain technology, with its immutable and transparent architecture, offers a robust alternative for enhancing log integrity, traceability, and regulatory alignment.

Shekhtman and Waisbard [2] presented EngraveChain, a Hyperledger Fabric platform with tamper-proof log storage and decentralized validation, for use in enterprise environments where transparency and privacy are needed. Wang et al. [3] presented a distributed security system using smart contracts to enforce data confidentiality and integrity rules in real time. These advancements complement ideas from the PCI Security Standards Council [4], which aim at continuous, auditable log monitoring for high-security settings.

Blockchain in Forensics and Compliance Auditing

Blockchain has found traction in digital forensics, especially in Chain-of-Custody (CoC) preservation. Al-Khateeb et al. [5] demonstrated how blockchain technology can be leveraged to secure IoT log trails, maintaining evidentiary integrity and admissibility in court. Building on this idea further, Ahmad et al. [6] devised a private Ethereum-based architecture to hold evidence metadata, with access control and end-to-end traceability. These models form the conceptual foundation for the implementation of auditable, multi-tenant compliance models in enterprise SIEMs.

Following more recent subsequent related blockchain integrations with Hyperledger Fabric, forensic readiness support is also available. Salzano and Pareschi [7] deployed a Fabric-based monitoring system with NLP for anomaly detection in logs through REST APIs. Meanwhile, González-Granadillo et al. [8] investigated the development of SIEM architectures based on big data analytics, which fit blockchain's scalable, distributed architecture.

Performance Considerations and Implementation Readiness

Latency and scalability are the major concerns in the deployment of blockchain-based SIEMs. Gorenflo et al. [9] compared the throughput of Hyperledger Fabric, stating that it can process thousands of transactions per second in some configurations. Thakkar et al. [10] built on this by analyzing bottlenecks in Fabric's consensus protocols and suggesting optimizations appropriate for enterprise deployment. Wang and Chu [11] built on this by analyzing how endorsement policies directly impact system performance, stating security granularity vs. transaction latency trade-offs.

Smart contracts have special strengths for regulatory automation. Amato et al. [12] had proposed models for verification of legal compliance in IoT settings, employing contract logic to enforce policy conditions. Muneeb et al. [13] had proposed SmartCon, an administration system for smart contracts, and Finck [14] had analyzed GDPR implications of automated processing and suggested guidelines for ensuring the balance between automation and compliance with privacy obligations. These works form the basis of our system's compliance-oriented design.

Access Control and Data Privacy

Strong access control continues to be needed for safe, multi-tenant logs. Pathak et al. [15] had a trust-aware Attribute-Based Access Control (ABAC) solution in edge computing ecosystems based on blockchain. Wang et al. [16] later investigated private sets of data within Hyperledger Fabric to support disciplined sharing across the network actors. Li et al. [17] generalized ABAC to supply chains on blockchains, even though security for implementation and misconfigured policy issues were identified [16].

Cryptographic enhancement also provides log integrity. Hu et al. [18] introduced Merkle 2, a low-latency transparency log, intended to reduce computational overhead while still being secure. Sun et al. [19] demonstrated the feasibility of blockchain in storing medical records in distributed environments, further establishing its viability in many areas of critical data management.

To enable high scalability and availability, our design utilizes Kubernetes for container orchestration [20]. Chacko's recent work [21] offers insights regarding global performance optimization techniques in decentralized systems, with a practical basis for production deployments.

IV. Proposed Framework – Tamper-Proof Compliance Model

A. *System Architecture*

The system architecture is based on a modular structure, offering secure, scalable, and efficient Hyperledger Fabric-based compliance validation.

It consists of three fundamental elements:

The SIEM Log Integration module seamlessly integrates major Security Information and Event Management (SIEM) products such as Splunk, ELK Stack, and IBM QRadar. Organizations can securely configure their SIEM instances for log forwarding to an API endpoint with JSON Web Token (JWT)-based authentication and digital signatures to validate the integrity and authenticity of the logs. A JWT token is assigned to each organization, which holds a company-specific identifier, thus preventing unauthorized usage and proper log attribution. Digital signatures are utilized to validate the integrity of logs prior to processing, thus deterring any spoofing or tampering attempts. This end-to-end security measure provides a trust-based and verifiable compliance process.

For tamper-proof compliance verification, permissioned blockchain technology Hyperledger Fabric is being used in the system. Organizations in the system will have their own private channel for isolated logging and there is a dedicated peer node assigned from peer pools made for secure compliance information handling. Chaincode (smart contract) provides compliance validation through validation against pre-defined controls of regulatory standards such as GDPR, PCI DSS, and NIST CSF. The system is modular and future extension is possible, accommodating new compliance standards as and when required. Each organization gets their own peer node to facilitate decentralized and autonomous validation. The system is highly multi-tenant, and there can be more than one organization with data isolation. architecture design is scalable, and adding new compliance frameworks and security controls is straightforward by adding new ones as the regulatory requirements evolve.

The Compliance Scoring and Alerting tool identifies log data against pre-set compliance controls. The logs are split into two groups: Compliant and Non-Compliant. Each company is given a compliance score, representing the degree of compliance of the company with regulatory compliance. If the compliance score falls below a defined level, automatic alerts notify responsible stakeholders such that it becomes easy to trigger remediation actions. The system also applies Role-Based Access Control (RBAC) in the compliance dashboard. Different roles such as Audit Officers, Compliance Officers, and Administrators are given selective access rights to audit reports, compliance controls, and system reports. Additionally, reports are downloadable for the purpose of regulatory audit and internal analysis.

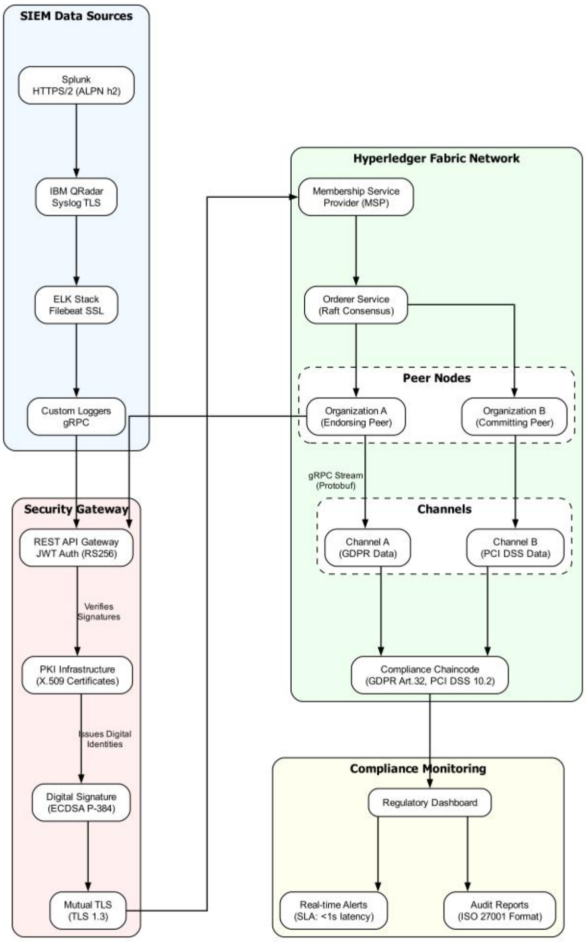

Fig. 1. System Architecture for Auditable Log Management

Fig. 1 shows an end-to-end perspective of how the logs flow from SIEM solutions to the blockchain network, receive compliance validation via smart contracts, and receive visualization via the compliance dashboard. Each layer, from data ingestion to alerting, is designed to preserve data authenticity, enable decentralized validation, and support regulatory readiness.

B. *Hyperledger Fabric Implementation*

The compliance validation framework is applied on Hyperledger Fabric using its permissioned, modular, and scalable design to offer privacy, security, and data integrity across organizational boundaries. The system uses a multi-channel, multi-peer topology where each organization is given a separate communication channel and an isolated peer node to offer data confidentiality.

As shown in Figure 2, the network structure includes a Raft-based ordering service and a centralized organizational domain (Org1MSP), which oversees a pre-defined pool of peers and a specific Certificate Authority (Org1-CA). Upon the onboarding of a new company, a peer from the pre-defined pool is allocated and subsequently registered via the CA, thereby obtaining cryptographic credentials that enable secure participation in the network.

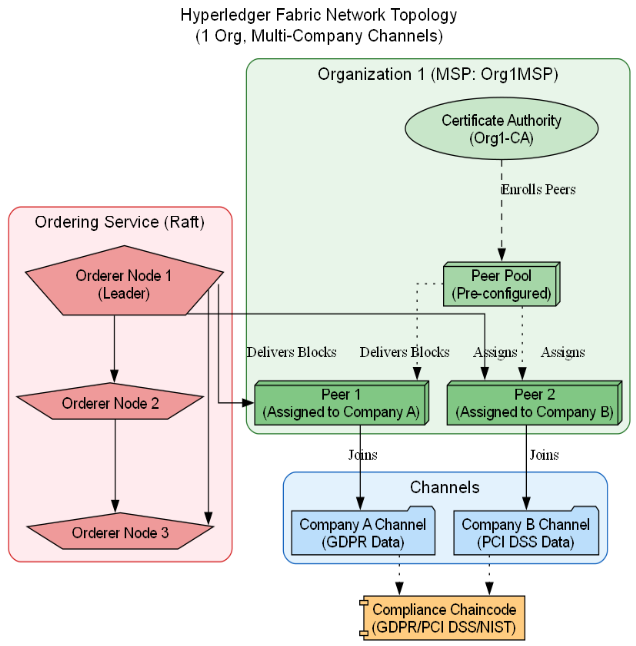

Fig. 2. Hyperledger Fabric Network Topology for Tamper-Proof Compliance

During onboarding, the system provisions a separate channel for each company (Company A Channel for GDPR data, Company B Channel for PCI DSS data). This isolates logs and maintains privacy because peers only have access to their respective channels. These access permissions are managed by Membership Service Provider (MSP) policies that uphold cryptographic identity authentication.

To add stronger security controls, Fabric uses watchful access mechanisms via Attribute-Based Access Control (ABAC) and endorsement schemes. These prevent requests placed by one company from being disclosed and revealed within another company's logs even within a shared network.

Each company has its own dedicated Certificate Authority (CA) to issue cryptographic identities for all entities, including peer nodes and administrators. These certificates play a pivotal role in offering secure transaction authentication and communication in the network.

The Raft-based ordering service delivers accurate, consistent ordering of transactions across all channels and securely broadcasts verified blocks to the target peers. While each company has its own peer, all peers together provide a decentralized ledger where verifiable compliance records are stored. The chaincode (smart contract) instantiated on each peer applies regulatory controls in line with pre-determined standards like GDPR, PCI DSS, and NIST CSF. These controls are hard-coded in the smart contract logic and analyze logs in real-time, marking them as Compliant or Non-Compliant.

The smart contract life cycle packaging, installing, approving, and committing is invoked as necessary when compliance logic changes. This ensures that the platform is responsive and agile to changing regulatory needs while also improving system scalability, security, and audit readiness.

In brief, the Hyperledger Fabric-based implementation guarantees:

- Isolation of per-organization data and compliance logic
- Smart contract-based real-time enforcement of compliance
- Cryptographic authentication and role-based access
- Cross-organizational log non-disclosure and attribute-based policy control
- Multi-standard auditing support and scalable onboarding

This design strategy gives a decentralized structure that ensures compliance with regulations by the principles of transparency, traceability, and tamper-evidence.

C. Compliance Verification & Scoring Process

The compliance verification process is initiated by secure transmission of SIEM logs to the system's API. Log records are authenticated using Hyperledger Fabric's chaincode, which enforces compliance with GDPR, PCI DSS, and NIST CSF controls. Logs are stamped as Compliant or Non-Compliant according to pre-defined rules of compliance.

The existing architecture supports the validation of compliance by using five fundamental controls adopted from other regulatory models. The controls are adopted and included in the chaincode to enable real-time validation and the security log classification, as per the requirements set by GDPR, PCI DSS, and NIST CSF. The following Table 1 explains the adopted control specifics for each of the regulatory models:

TABLE I. REGULATORY FRAMEWORK COMPLIANCE CONTROLS AND VALIDATION MECHANISMS

Frame work	***Control Number***	***Control Name***	***SIEM Log Example***	***Chaincode Validation Logic***
GDPR	Art. 32(1)(a)	Encryption of Personal Data	encryption_method=null for sensitive data	Checks if encryption_method is TLS 1.2+ or AES-256
	Art. 33(1)	Breach Notification Within 72 Hours	No breach report logged within 72 hours of exfiltration	Validates incident_reported=true within 72h
	Art. 25(1)	Integrity & Confidentiality	log_altered=true (tampering detected)	Validates log immutability in Hyperledger
	Art. 17(1)	Right to Erasure ("Right to Be Forgotten")	Data accessed post-deletion request	Cross-checks data_deletion_request timestamps
PCI DSS	Req. 3.4	Render PAN Unreadable	PAN stored in plaintext (card_data_masked=false)	Verifies data_masking=true or tokenization
	Req. 7.1	Restrict Access to Cardholder Data	Unauthorized access to card_transactions.log	Validates user_role against PCI DSS access matrix
	Req. 4.1	Encrypt Cardholder Data in Transit	Payment data sent via HTTP (protocol=HTTP)	Ensures protocol=TLS 1.2+
	Req. 8.1.6	Limit Failed Authenticati	10+ failed logins in 5	Triggers alert if

Frame work	***Control Number***	***Control Name***	***SIEM Log Example***	***Chaincode Validation Logic***
		on Attempts	minutes from same IP	authentication_failures > threshold
	Req. 10.5	Secure Audit Logs	No transaction.log entries for 24h	Monitors heartbeat_logs for gaps
NIST CSF	PR.IP-12	Vulnerability Management	Last scan 8 months ago (last_scan_date outdated)	Flags if current_date - last_scan_date > 90 days
	ID.AM-1	Inventory of Authorized Devices	Unregistered device_id connected	Checks device_id against CMDB
	PR.AC-5	Network Segmentation	Login from high-risk geo_location	Compares IP against trusted_countries list
	DE.CM-1	Continuous Monitoring	Zero security events in 48h (possible logging failure)	Validates log_receipt_timestamp is recent
	RS.AN-5	Incident Response Reporting	Incident detected but no incident_reported=true	Ensures incident logs include response actions

The compliance scoring system evaluates an organization's compliance against regulations based on metadata-driven logs, patterns, and keywords. Unlike enumerating all the violations as equal, the system applies weighted scoring in which every instance of non-compliance is given a different score depending on the severity levels.

The compliance score is calculated using the following formula:

$$S = \left(\frac{W_C C}{W_C C + \sum W_{NC_i} NC_i}\right) \times 100$$

where:

- S = Compliance Score (%)
- C = Number of Compliant Logs
- NC_i= Number of Non-Compliant Logs for violation iii
- W_C = Weight for compliant logs (default = 1)
- W_{NC_i}= Weight assigned to each non-compliant log type (higher values for critical violations)

Example Calculation

Suppose an organization handles 1,000 logs as follows:

- Compliant Logs (C): 700

Non-Compliant Logs (NC):

- Authentication Failures (Low Impact, W=1W = 1W=1): 150
- Unencrypted Data Transfer (High Impact, W=3W = 3W=3): 100
- Unauthorized Admin Access (Critical, W=5W = 5W=5): 50

Applying the formula:

$$S = \left(\frac{1 \times 700}{(1 \times 700) + (1 \times 150) + (3 \times 100) + (5 \times 50)}\right) \times 100$$

$$S = \left(\frac{700}{700 + 150 + 300 + 250}\right) \times 100$$

$$S = \left(\frac{700}{1400}\right) \times 100 = 50\%$$

This approach ensures that severe violations are assigned higher weight in the compliance score compared to less significant violations, hence giving a balanced and realistic estimate of compliance.

Weight Preservation in Chaincode

To ensure integrity, weight values for all controls are stored in the Hyperledger Fabric chaincode, thereby guaranteeing:

- Regulation-compliant severity levels according to regulatory requirements (GDPR, PCI DSS, NIST CSF).
- Automated updates based on regulatory changes.
- Continuous compliance scoring across organizations.

V. IMPLEMENTATION & RESULTS

Tamper-Proof Compliance Model was developed using Hyperledger Fabric in an effort to provide secure, immutable, and verifiable compliance logging. The application was deployed in a controlled environment using Kubernetes for Hyperledger Fabric component management and SIEM log ingestion APIs. The following described key features of the implementation have been covered:

- **Log Collection & Processing**: Logs were gathered from various SIEM systems (Splunk, QRadar, ELK Stack) and securely transported using JWT authentication and digital signatures. logs get authenticated and store in hyperledger fabric channel for each company.
- **Chaincode Execution and Compliance Verification**: Chaincode deployed guaranteed compliance checks were carried out according to the prescribed protocols as stated in the GDPR, PCI DSS, and NIST CSF frameworks. The compliance score was updated continuously depending on the compliance with the protocols.

- **Compliance Alert Notification and Monitoring:** There existed an automated system that scanned logs and triggered alarms on cases of violation of compliance thresholds. This allowed real-time monitoring and corrective action at the earliest.

  The system used Role-Based Access Control (RBAC), where three distinct roles Admin, Compliance Officer, and Auditor were defined, each having specific access rights to logs, compliance reports, and analytics dashboards.

- **Dashboard and Reporting**: A high-end dashboard was created to graphically display compliance scores, historical trends, and log audit trails. Improved filtering and searching capabilities allowed users to effectively derive meaningful insights related to compliance.

A. *Practical Compliance Logging & Violation Detection*

In order to validate the mechanism of enforcing compliance, logs were gathered in JSON format from various SIEM sources. A structured format was utilized for easy processing, such as metadata to be used for validation and compliance checking.
A sample log entry is as follows:

```
{
  "@timestamp": "2025-03-30T12:45:00Z",
  "event": {
    "provider": "ELK",
    "dataset": "authentication",
    "category": ["authentication"],
    "type": ["failure"]
  },
  "organization": {
    "company_id": "COMP12345",
    "department": "finance"
  },
  "user": {
    "name": "test_user",
    "role": "contractor"
  },
  "source": {
    "ip": "192.168.1.15",
    "geo": {
      "country_iso_code": "RU"
    }
  },
  "destination": {
    "application": "admin_portal",
    "endpoint": "/api/cardholder_data"
  },
  "log": {
    "level": "warning",
    "message": "Failed authentication attempt for admin_portal",
    "original": "User 'test_user' failed login due to invalid credentials."
  },
  "action": {
    "type": "login",
    "status": "failed",
    "reason": "invalid_credentials",
    "attempts": 12
  },
  "security": {
    "authentication": {
      "protocol": "TLS_1.2",
      "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI..."
    },
    "integrity": {
      "hash": {
        "algorithm": "SHA-256",
        "value": "5f4dcc3b5aa765d61d8327deb882cf99"
      }
    },
    "signature": {
      "algorithm": "RSA-2048",
      "value": "b3V6...dGh9A=="
    }
  }
}
```

When a log is received, the system follows these procedures:

1. JWT Verification: The API subsequently verifies the JWT in the log using the public key of the registered firm. This ensures that the log has come from an authenticated and trusted source.
3.
4. Metadata Extraction: It verifies the company_id field to determine the related Hyperledger Fabric channel and route the log to the relevant partition of a ledger.
5. Chaincode Execution: Upon the log is routed, the system runs a pre-defined smart contract (chaincode) to verify compliance metrics. For instance, according to GDPR Art. 32(1)(a), it verifies whether encryption_method is AES-256 or TLS 1.2+. If encryption_method=null, log validation fails:

```
func (s *SmartContract) ValidateEncryption(ctx
contractapi.TransactionContextInterface,
encryptionMethod string) (bool, error) {
   if encryptionMethod != "TLS1.2" &&
encryptionMethod != "AES-256" {
       return false, fmt.Errorf("Encryption compliance
failed: method = %s", encryptionMethod)
   }
   return true, nil
}
```

6. Threshold-Based Alerting: On the occurrence of compliance breaches, alerts are initiated based on the level of severity.

A threshold-based compliance alerting mechanism was implemented, and alerts were triggered when compliance fell below predefined values, as detailed in Table II:

TABLE II. COMPLIANCE ALERT LEVELS AND CORRESPONDING ACTIONS

Compliance Score (%)	***Alert Level***	***Action Protocol***
90 - 100	Normal	Routine monitoring
70 - 89	Advisory	Review with team leads
50 - 69	Warning	Formal notification to Compliance Officer
Below 50	Critical	Immediate remediation + root cause analysis

2. Public Key Signature Verification: The digital signature is checked using RSA-2048 to ensure log integrity. Tampering makes the log unusable, demonstrating adherence to GDPR Art. 32(1)(a) on encryption and protection of data.

B. *Secure Communication and Authentication*

In order to ensure a tamper-resistant system for the management of compliance logs, some security mechanisms were incorporated into the design. Initially, a public-private key encryption mechanism was applied, and to each enrolled company, a unique pair of keys was assigned, which was maintained by the Hyperledger Fabric's Membership Service Provider (MSP). This cryptographic mechanism guaranteed that only authorized parties could send compliance logs. Secondly, a JWT-based authentication mechanism was implemented, where companies were compelled to authenticate their log submissions through JWT tokens cross-checked against their registered credentials.

To further ensure the integrity of the logs, digital signatures were included. Each submitted log was cryptographically hashed using the private key of the submitting party prior to transmission, thereby guaranteeing non-repudiation and authenticity of the data. The MSP system also played a crucial role in imposing access control by restricting unauthorized parties from accessing log submission and retrieval activities. Through the incorporation of these multi-faceted security features, the system was able to effectively reduce the risks of log tampering, forgery, and unauthorized data injection, thereby offering a secure mechanism for compliance assurance.

C. *Performance Evaluation*

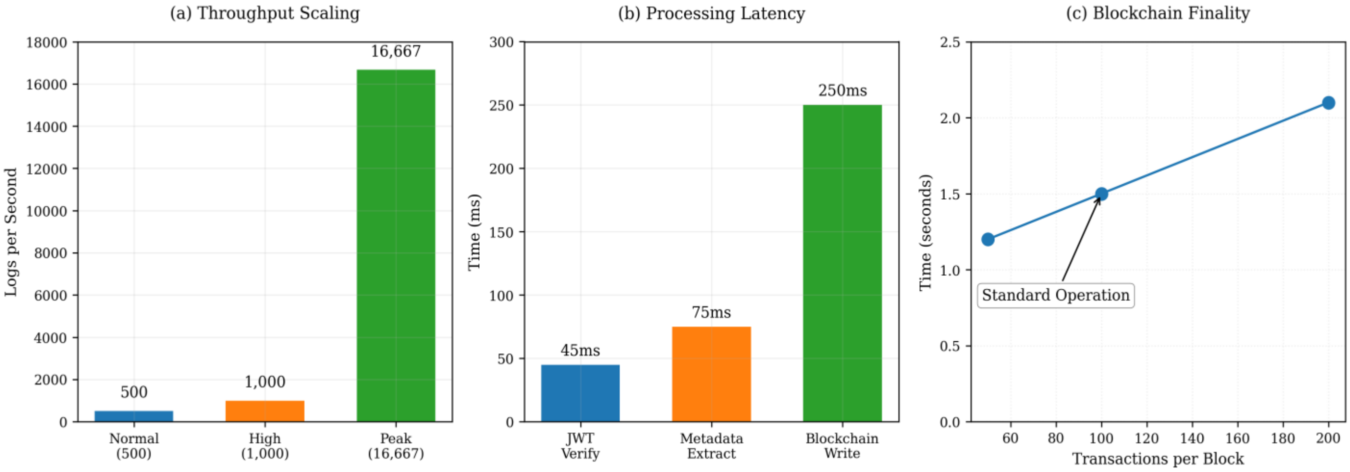

Fig. 3. System performance metrics: (a) Throughput scaling across operational loads, (b) Latency breakdown for log processing components, and (c) Blockchain finality times as a function of block size.

The system performance was validated by ingesting large quantities of security logs gathered from various heterogeneous SIEM feeds under varied load scenarios. The system of interest demonstrated a high-performance, scalable log processing pipeline with readings of throughput continually hitting 500 logs per second under normal operational loads and a maximum of 1,000 logs per second under heavy usage scenarios. The system processed as much as 16,667 logs per second under maximum loads, thus depicting the scalability of the Kubernetes environment (see Fig. 3(a)).

A breakdown of latency by large segments of the pipeline revealed that JWT verification and metadata extraction operations combined contributed an average of 120 milliseconds per log record, and blockchain write latency contributed a further 250 milliseconds per transaction. These results indicate that the system can support compliance logging near-real-time with minimal delay (see Fig. 3(b)).

For ensuring tamper-evidence and auditability, blockchain commit finality was achieved around 1.5 seconds per transaction with Hyperledger Fabric. The finality time, which is validated across the spectrum of block sizes, indicates a stable and effective commit rate, which is of utmost importance for forensic soundness in tracking compliance. As shown in Fig. 3(c), the system ensures low and predictable finality even at scaled transaction loads, with 100 transactions per block as the standard operating point.

To counter performance degradation risks under traffic surges, an autoscaling mechanism using Kubernetes was implemented. This enabled dynamic provisioning of additional API pods and Fabric peer nodes in response to increasing ingestion rates. Under stress testing, the design scaled without fail to process up to 1 million logs per minute, with steady performance and minimal latency fluctuations [20][21][10][11].

D. *Multi-Organization Support and Identity Management*

One of the most important aspects of the framework was that it could support multiple organizations and manage identities, thereby offering data isolation and protection between entities. Each of the participating organizations functioned on a separate Hyperledger Fabric channel, thus successfully preventing cross-contamination of compliance logs. During the onboarding process, a new channel was dynamically established for each organization to allow data segregation and independent compliance analysis. Chaincode was also instantiated within each channel, allowing compliance logic specific to individual organizations to be executed.

To allow for proper identity resolution, each organization listed was assigned a unique identifier specific to each company (company_id). This identifier allowed the system API to direct dynamic log submissions to their corresponding channels, thereby facilitating organizational-level compliance analysis. With the assignment of a unique channel to each organization, the platform allowed compliance logs to be retained in a separate, secure, and auditable manner, which ensured the protection of the integrity and confidentiality of sensitive enterprise security information.

E. Comparative Analysis

Table III presents a comparative analysis which outlines the advantages of the proposed system over classic compliance validation frameworks and standard blockchain frameworks, especially security, privacy, scalability, and auditability. The table summarizes how the major features are handled in the proposed framework as opposed to current frameworks.

TABLE III. COMPARATIVE ANALYSIS OF TRADITIONAL LOGGING VS. HYPERLEDGER-BASED COMPLIANCE MODEL

Feature	***Traditional Logging***	***Hyperledger-Based Model***
Data Integrity	Logs can be altered post-creation with admin access	Cryptographic hashing and blockchain immutability prevent all modifications
Access Control	Basic role-based access (static permissions)	Dynamic MSP-based RBAC with attribute-based policies
Compliance Proofs	Manual validation required for audits	Automated smart contract validation with immutable evidence
Tamper Evidence	No inherent detection of modifications	Every change creates new blockchain transaction with cryptographic proof
Monitoring Capabilities	Batch processing with delayed alerts	Real-time policy evaluation and instant notifications

VI. CONCLUSION

This paper proposes a Tamper-Proof Compliance Framework on top of Hyperledger Fabric to enable secure, immutable, and verifiable log management in corporate environments. The proposed framework leverages JSON Web Token (JWT) authentication, digital signatures, and Role-Based Access Control (RBAC) to guarantee log integrity and authenticity, as well as access limitation, thus addressing significant security issues in compliance enforcement. By leveraging designated blockchain channels, organizations can perform data segregation and independent verification, thus ensuring regulatory compliance such as GDPR, PCI DSS, and NIST CSF. The smart contract-based compliance mechanism enables real-time verification and scoring of logs, real-time policy violation alerts, and reduced dependence on manual auditing procedures.

In addition, the architecture is also designed to ensure scalability, where multiple organizations and compliance models are integrated smoothly without sacrificing high throughput and low latency in log processing. The immutability and cryptographic verification aspects provided by Hyperledger Fabric enhance trust and transparency, and the solution provides a tamper-proof and verifiable compliance model.

REFERENCES

[1] Satyanarayana, N. (2024, February). A Blockchain-Based Security Assessment Framework. In *2024 26th International Conference on Advanced Communications Technology (ICACT)* (pp. 1483-1493). IEEE.

[2] Shekhtman, L., & Waisbard, E. (2021). Engravechain: A blockchain-based tamper-proof distributed log system. *Future Internet*, *13*(6), 143.

[3] Wang, R., Liu, H., Wang, H., Yang, Q., & Wu, D. (2019). Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches. *IEEE Wireless Communications*, *26*(6), 30-36.

[4] Hancock, S. (2024). PCI DSS Version 4.0: A guide to the payment card industry data security standard.

[5] Al-Khateeb, H., Epiphaniou, G., & Daly, H. (2019). Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger. *Blockchain and Clinical Trial: Securing Patient Data*, 149-168.

[6] Ahmad, L., Khanji, S., Iqbal, F., & Kamoun, F. (2020, August). Blockchain-based chain of custody: Towards real-time tamper-proof evidence management. In *Proceedings of the 15th international conference on availability, reliability and security* (pp. 1-8).

[7] Salzano, F., & Pareschi, R. (2023). Enhancing blockchain security through natural language processing and real-time monitoring. *International Journal of Parallel, Emergent and Distributed Systems*, 1-16.

[8] González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures. *Sensors*, *21*(14), 4759.Finck, M. (2019). Smart contracts as a form of solely automated processing under the GDPR. *International Data Privacy Law*, *9*(2), 78-94.

[9] Gorenflo, C., Golab, L., & Keshav, S. (2020, May). XOX Fabric: A hybrid approach to blockchain transaction execution. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 1-9). IEEE.

[10] Thakkar, P., Nathan, S., & Viswanathan, B. (2018, September). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 IEEE 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS)* (pp. 264-276). IEEE.

[11] Baliga, A., Solanki, N., Verekar, S., Pednekar, A., Kamat, P., & Chatterjee, S. (2018, June). Performance characterization of hyperledger fabric. In *2018 Crypto Valley conference on blockchain technology (CVCBT)* (pp. 65-74). IEEE.

[12] Amato, F., Cozzolino, G., Moscato, F., Moscato, V., & Xhafa, F. (2021). A model for verification and validation of law compliance of smart contracts in IoT environment. *IEEE Transactions on Industrial Informatics*, *17*(11), 7752-7759.

[13] Muneeb, M., Raza, Z., Haq, I. U., & Shafiq, O. (2021). Smartcon: A blockchain-based framework for smart contracts and transaction management. *IEEE Access*, *10*, 23687-23699.

[14] Finck, M. (2019). Smart contracts as a form of solely automated processing under the GDPR. *International Data Privacy Law*, *9*(2), 78-94.

[17] *IEEE 41st International Conference on Distributed Computing Systems (ICDCS)* (pp. 819-829). IEEE.

[18] Li, S., Zhou, T., Yang, H., & Wang, P. (2023). Blockchain-based secure storage and access control scheme for supply chain ecological business data: A case study of the automotive industry. *Sensors*, *23*(16), 7036.

[19] Hu, Y., Hooshmand, K., Kalidhindi, H., Yang, S. J., & Popa, R. A. (2021, May). Merkle 2: A low-latency transparency log system. In *2021 IEEE Symposium on Security and Privacy (SP)* (pp. 285-303). IEEE.

[15] Pathak, A., Al-Anbagi, I., & Hamilton, H. J. (2023). TABI: Trust-based ABAC mechanism for edge-IoT using blockchain technology. *IEEE Access*, *11*, 36379-36398.

[16] Wang, S., Yang, M., Zhang, Y., Luo, Y., Ge, T., Fu, X., & Zhao, W. (2021, July). On private data collection of hyperledger fabric. In *2021*

[20] Sun, J., Yao, X., Wang, S., & Wu, Y. (2020). Blockchain-based secure storage and access scheme for electronic medical records in IPFS. *IEEE access*, *8*, 59389-59401.

[21] Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade. *Queue*, *14*(1), 70-93.

[22] Chacko, J. A. (2024). Holistic Approaches to Performance Optimization in Decentralized Systems: A Study of Hyperledger Fabric (Doctoral dissertation, Technische Universität München).