International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(17s) 2026 pp. 877-887
https://doi.org/10.70917/ijcisim-2026-4806
Article

Security and Privacy Controls in Ingestion Pipelines (PII masking, Encryption, Access Governance)

Shreyansh Sharma

Independent researcher, India, Email: pritikacsc.rsh@gndu.ac.in

Abstract: The use of high-throughput data ingestion pipelines that can constantly amass and process data collected by heterogeneous sources into a centralized or distributed data storage system is increasingly becoming the cornerstone of modern enterprise data ecosystems. Since these pipelines handle delicate personally identifiable information (PII) and financial data, healthcare data, and proprietary telemetry, it has become a critical concern of organizations that have to work within high regulatory standards, including GDPR, HIPAA, and CCPA, to guarantee high-quality security and privacy controls through all ingestion phases. Formal comparison with the Apache Kafka and Apache Spark-based pipeline deployments shows that the suggested framework yields an overhead of the throughput no more than 7.3% and offers extensive coverage of PII protection and role-based access control. These findings suggest that appropriately designed security controls can be integrated into production ingestion pipelines without degrading the performance of those pipelines materially, disproving the long-standing belief that there is an inherent trade-off between the security of data and the operation of pipelines.

Keywords: Data Ingestion Pipelines, PII masking, Data Encryption, Access Governance, RBAC, ABAC, Stream Processing, Data Privacy, GDPR compliance, Cybersecurity

1. INTRODUCTION

Due to the growth of digital services, Internet of Things (IoT) gadgets and cloud-based applications, the amount of data created and moved across enterprise boundaries has never been higher. Organizations in the industries such as healthcare, financial, retail and the government constantly consume terabytes of data daily through pipelines that cut across various network segments, cloud providers and processing frameworks. The pipelines form a key constituent of the contemporary data architecture and allow the transfer of raw and frequently unstructured data sources into the analytics systems, data warehouses, and machine learning systems [1].

Nonetheless, the same attributes that cause data ingestion pipelines to be operationally powerful namely, high velocity, distributed architecture and real-time processing also make them vulnerable to a wide assortment of security threats. IBM Cost of a Data Breach Report 2023 approximated the mean cost of data breach to USD 4.45 million, with the pipeline-level vulnerability becoming a more significant form of attack [2].

Although the security threats of pipelines have been widely recognized, the scholarly field has so far looked at PII protection, encryption, and access governance as separate and different issues that have been approached separately. PII masking studies have been studying the techniques that can be applied to static datasets, and relatively little has been devoted to the special considerations of streaming architectures in which the data is fast-moving and has changing schema [3]. In the same way, when studying encryption in the pipeline setting, network-level transport security has been traditionally given significant emphasis at the expense of field-level and application-level encryption, which is a necessity when the data is passing through intermediary message brokers, transformation layers, and staging areas.

The issue of access governance, including the policies and technical structures that regulate access by whom or whom not to read or write, transform or change, or delete pipeline data has also been treated rather sporadically. Models Role-based access control (RBAC) developed in the context of traditional database settings are not easily mapped to the dynamic, events-driven pipeline architecture where the roles of data ownership, sensitivity classification and processing can vary within the context of a single pipeline execution [4].

This fragmentation is resolved in this paper by formulating an Integrated Security Control Framework (ISCF) that brings all three facets of PII masking, encryption, and access governance to a coherent reference model of both batch and stream-based ingestion pipelines. The main contributions of this work are the following:

1. A taxonomy of architectural data ingestion vulnerabilities-based security vulnerabilities.

2. Comparative study of PII masking, encryption and access governance methods were conducted in the light of pipeline-relevant metrics such as latency overhead, coverage completeness, and regulatory alignment.

3. ISCF reference architecture with all the three security control layers, interfaces, and points of enforcement.

4. ISCF performance assessment of Apache Kafka and Apache Spark environments using empirical performance evaluation.

The rest of this paper is arranged as follows: Section II is the review of related literature. Part III gives a taxonomy of ingestion pipeline security threats. Part IV, V, and VI explain the PII masking, encryption, and access governance mechanism respectively. Section VII outlines the proposed ISCF. In section VIII, experimental evaluation is given. Findings are discussed in Section IX and the conclusion is made in Section X.

2. RELATED WORK

Since the widespread use of distributed stream processing frameworks, the security of data pipelines has drawn an increasing scholarly interest. Initial background studies by Vidyasankar [5] defined the concept of continuous query processing, but did not consider security concerns of streaming architecture. Later innovations in Apache Kafka and Apache Flink brought about new data movement paradigms, which posed stemming security risks that were not envisaged by the previous database security paradigms.

A. PII Protection in Streaming Applications

The concept of k-anonymity formulated by Karagiannis et al. [6] as a formal model of PII protection in relational data sets provided the principles under which other researchers built on that concept to protect PII in streaming contexts. Yoo et al. [7] added the l-diversity on this framework to deal with the attribute disclosure risks. Nevertheless, these models were introduced as having been developed using static data and the extrapolation of them to streaming ingestion must undergo extensive adaptation to consider schema evolution and temporal correlations across event windows, which are considered to be an important progress in pipeline-relevant PII protection. A more current study by Majeed and Lee [8] has discussed the conflict between the utility of data and the strength of privacy in streaming anonymization and that in the aggressive methods of masking, downstream analytics becomes unreliable.

B. Data Pipeline Architectures Encryption

In studying the authentication and integrity checks of streaming data, Garg et al. [10] developed the foundations of the end-to-end encrypted segments of the pipeline. Cloud-native data platforms introduced the need to reconsider envelope encryption and key management; examples of key management service architectures (that have since become de facto when deploying a cloud pipeline) provided by Amazon Web Services [11] and Google Cloud Platform. A comparative study of encryption overhead of distributed data systems using Spark and Hadoop by Khamis [12] identified that field-level encryption analysis had 12-18% throughput overheads relative to 46% transport-level encryption overheads, an important factor in performance-sensitive pipeline systems.

C. Data Pipe Access Governance

The formal RBAC model introduced by Sandhu et al. [13] has prevailed in the enterprise access control design in the last 30 years. More recently, policy-as-code systems like Open Policy Agent (OPA) have gained a lot of research attention as tools to express and enforce governance policies programmatically in pipeline components [14]. In the context of the access control in a cloud data environment, Ameer et al. [15] performed an empirical study on the access control models and stated that the hybrid RBAC/ABAC implementation models outperformed both models in terms of policy expressiveness and acceptable administrative overhead. The data mesh governance concept, developed by Wider et al. [16], suggested models of decentralised data ownership, placing governance responsibility into the hands of domain-level pipeline owners, which provided both opportunity to apply access control with context sensitivity as well as difficulty in the overall enforcement of enterprise-wide policies.

D. Cohesive Pipeline Security Constructs

Combined methods which incorporate several security control measures in pipeline settings are still relatively infrequent in literature. Mavi [17] has suggested a security framework of the IoT data pipelines, which treats authentication and sanitization of data at the device level but has not expanded their discussion to the field-level PII masking and overall access governance. Davis-Turak et al. [18] also gave a general overview of the topic of database security that briefly discusses pipeline settings but is not as up-to-date as stream processing architectures have become. The current paper builds upon this literature and suggests and confirms a comprehensive framework specifically to operate in modern stream and batch ingestion settings.

3. DANGERS IN DATA INGESTION PIPES SECURITY

Effective security controls are designed by first having a rigorous understanding of the threat scenario that is specific to data ingestion pipelines. Ingestion pipelines pose a broadened attack surface, which can be pinned on their distributed and multi-component design, unlike monolithic database systems. Threats are classified into the following taxonomy based on their main locus in the pipeline lifecycle.

A. Source-Level Threats

Pipelines at the data source border also become susceptible to injection attacks, in which the malformed or maliciously crafted records of data are introduced in order to gain control over processing logic downstream. An especially pernicious form of schema corruption, where a malicious user corrupts source data structural metadata to deceive schema inference engines, is known as schema poisoning, and is specific to schema-on-read systems [18]. Also, unauthorized source impersonation, in which a malicious attacker registers a bogus producer endpoint on a message broker can lead to the introduction of false records into streams that are integrity-guaranteed otherwise.

B. Transit-Level Threats

Interception, man-in-the-middle attacks and replay attacks may attack data in transit between pipeline components. Although TLS is employed extensively to secure inter-service communication, misconfiguration, certificate handling error, and utilizing outdated cipher suites can be considered a constant weakness [19]. Plaintext inter-broker communication is a typical misconfiguration in Kafka based architectures specifically, where sensitive data is exposed to network level attackers with access to internal cluster traffic.

C. Processing-Level Threats

Insecure deserialization vulnerabilities may enable enemies to run arbitrary code in pipeline worker processes with transformation and enrichment stages. Also, the operators of pipelines that have wide permissions can accidentally or intentionally leak information at the transformation phases where there is no or inadequate access logging.

D. Storage-Level Threats

Information in rest in staging locations, the message broker topic, or in-between object storage is accessible to unauthorized principals, especially in cloud environments where a misconfigured storage bucket permission is one of the most common sources of massive data exposures. Lack of adequate data retention controls can also lead to the continuing presence of the sensitive data even after the completion of its lifecycle, which opens up unwarranted windows of exposure.

TABLE I: Taxonomy of Security Threats in Ingestion Pipelines

Threat Category	Example Threat	Primary Impact	Pipeline Stage
Source-Level	Schema poisoning	Data integrity	Ingestion
Transit-Level	Man-in-the-middle	Confidentiality	Transport
Processing-Level	Insecure deserialization	Integrity/Availability	Transformation
Storage-Level	Misconfigured bucket ACLs	Confidentiality	Persistence
Governance	Privilege escalation	Access control	All stages

4. PII MASKING TECHNIQUES

PII masking represents a wide set of data transformations methods that diminish or removes the identifying strength of individual area of personal information, yet retain adequate analytical worth to allow legitimate downstream purposes. To choose the right masking technique, one must pay close attention to the reversibility needs, the amount of identifying information that has to be reduced, and the amount of work that can be done in the pipeline environment.

A. Deterministic Masking

Deterministic masking is an affixation that gives a uniform, predictable transformation of PII fields in the sense that a certain input value is continuously given the same masked output. This is a key property that plays the role of maintaining the relational integrity of the masked data in pipelines that supply pipelines into a system that does entity resolving or referential joins between datasets [20]. One of the most common deterministic masking methods, deterministic tokenization, substitutes the values of PII with format-preserving format-insensitive surrogate tokens, which are stored in a secure mapping table. The major threat of deterministic masking is that the fixed relationship between input and output can be used to conduct inference attacks, in case an adversary can observe enough input-output pairs.

B. Format-Preserving Encryption (FPE)

FPE modes, such as FF1 and FF3-1, as defined by NIST, allow the encryption of credit card numbers, social security numbers, and telephone numbers to be outputted in identically-length and character-set output, when masked, meets downstream format validation requirements with no schema alteration requirements [21]. FPE is especially useful in ingestion pipelines that are fed by systems with hard input format specifications and also in pipelines that store masked fields, where schema modification to represent these fields is unnecessary.

C. Differential Privacy

Differential privacy (DP) is a mathematically rigorous privacy guarantee, where the computation is applied to data or query responses with random noise added to them, so that the statistical output of a computation is statistically indistinguishable with or without the data of a specific individual present in the data [22]. In pipeline applications, DP mechanisms like the Laplace mechanism and Gaussian mechanism may be used during aggregation/summarization phases in order to secure an individual record whilst maintaining statistical characteristics at the population level. The parameter most important is epsilon which determines privacy-utility trade-off whereby lower epsilon values ensure greater privacy guarantees at the expense of decreased data utility.

D. Generalization and Suppression of Data

Generalization the specific PII values are substituted with less specific, more generalized categories, such as an exact date of birth being substituted by a range of years of birth. The suppression obliterates the PII fields in case there is no acceptable generalized representation. These are methods, which form the basis of k-anonymity and l-diversity models, and are especially useful when the data being consumed feeds reporting or analytics pipelines in which fine-grained data about individuals is not required [23]. In order to determine which fields need to mask, automated PII recognition based on named entity recognition (NER) models and regular expression matching is often employed to detect fields that need masking in a schema less or semi-structured ingestion stream.

TABLE II: Comparison of PII Masking Techniques

Technique	**Reversibility**	**Utility Preservation**	**Streaming Suitability**	**Computational Cost**
Deterministic Tokenization	Yes (with mapping)	High	High	Low
Format-Preserving Encryption	Yes (with key)	Very High	High	Medium
Differential Privacy	No	Medium	Medium	Medium
Generalization/Suppression	No	Low-Medium	High	Low
Pseudonymization	Yes (with key)	High	High	Low-Medium

5. ENCRYPTION STRATEGIES OF INGESTION PIPELINES

The encryption of data ingestion pipelines should address three unique protection specifications: the data that travels between the structures of the pipeline, the data that rests in the persistent storage and staging areas, and the data that is used during the active transformation and processing.

A. Transport-Layer Encryption

The modern standard of encrypting the data in transit among the pipeline producers, message brokers, transformation workers, and sink systems is transport-layer security (TLS) 1.3. TLS 1.3 provides 1-RTT and 0-RTT handshake modes that greatly decrease the latency of the handshake relative to its predecessors, which is essential when requiring pipelines of high frequency to run micro-batches. Mutual TLS (mTLS) is more authenticated by both the client and the server with valid certificates and is a more preferred process than unidirectional TLS and should be used in inter-service communication when in the pipeline cluster [24].

B. At-Rest Encryption

At-rest encryption secures data that has been stored in the message broker topic logs as well as in the object storage and data lake storage layers. Cloud object storage providers widely support server-side encryption (SSE) based on the AES-256 in the GCM mode, which offers authenticated encryption, at the same time providing the confidentiality and integrity of the data stored [25]. Envelope encryption architectures, in which data encryption keys (DEKs) are encrypted by key encryption keys (KEKs), which are in turn encrypted by key management services (KMS) are a scalable key lifecycle management architecture in pipelines with large volumes of diverse data.

C. Field-Level and Application-Level Encryption

The field-level encryption (FLE) is a cryptographic protection applied to a set of separate sensitive items in the record, which enables non-sensitive fields of records to be handled by pipeline elements without revealing the items that are under the protection. This method is especially useful in pipelines in which processing logic must access some

record fields (typically routing keys or temporal attributes), but not PII fields [26]. The open-source cryptographic library Tink, part of Google, offers standard primitives of field encryption that are easily integrable into pipelines because of their key rotation functions, as well as because of their deterministic encryption mode option, which allows pipeline join operations.

D. Key Management and Rotation

One of the hardest parts of pipeline encryption is the cryptographic key management. The main rotation policies should strike a balance between the security advantage of high-frequency key rotation and the high cost of re-encryption of the current data and the complexity of key distribution among distributed pipeline workers. Hardware Security Modules (HSMs) are tamper-resistant storage of keys that are recommended in pipelines that require the PCI-DSS or FIPS 140-2 standards.

6. FRAMEWORKS OF ACCESS GOVERNANCE

Data ingestion pipeline access governance refers to the policies, models, and technical mechanisms that govern the access of particular operations that can be carried out by particular principals, such as users, service accounts, and automated components of the pipeline.

A. Role-Based Access Control (RBAC)

RBAC does not deal with permissions per-principal, but per-role, eliminating administrative burden and the concept of least privilege [13]. In pipeline applications, they are represented by Pipeline Administrator, Pipeline Developer, Data Steward, Analytics Consumer, and Compliance Auditor and each role is associated with a specific set of allowed operations on particular pipeline elements, topics, and datasets. The ACL functionality in Apache Kafka and Apache Ranger offer RBAC to the message broker and Hadoop ecosystem components respectively, allowing the provision of similarity in the allocation of permissions across the heterogeneous infrastructure of the pipeline infrastructure.

B. Access Control based on Attributes (ABAC)

ABAC compares the access decision with the policies written in terms of logical conditions on attributes of the subject, resource, action and environment [14]. The subject attributes can be departmental affiliation and security clearance level, resource attributes can be data sensitivity classification and geographic jurisdiction of data origin as well as regulatory category (e.g., GDPR-regulated, HIPAA-regulated), and the environmental attributes can be time of access, network zone, and context of an authenticated session. ABAC in a pipeline is especially useful when the data stream being handled by a given pipeline has different regulatory demands depending on the geography of data origin, allowing automatic imposition of access limits based on jurisdiction without requiring manual reconfiguring of the policy.

C. Policy-as-Code and Open Policy Agent

Open Policy Agent (OPA) has become a policy-as-code de facto standard in the cloud-native world, offering a policy language (Rego) of declarative policies and a decoupled policy evaluation engine which can be embedded in pipeline components as a sidecar or central policies server [16]. OPA allows the same policy to be enforced on heterogeneous pipeline components with each component not needing to execute proprietary access control logic, which greatly minimizes the attack surface that can be attributed to varying policy implementation across pipeline stages.

D. Data Masking Governance and Dynamic Data Access Control

Dynamic data access control is an extension of access control to fine-grained data delivery policies based on content-aware access control decisions that are not limited to coarse-grained resource access decisions. With a dynamic masking model of governance, underlying pipeline data can be presented in various representations to

different consumer roles: a data scientist can be presented with pseudonymized records whereas a compliance auditor can be presented with fully unmasked records and with the correct masking transformation applied upon querying based on the consumer role and data governance policy [27]. The Row-Level Security and Column Masking features offered by Apache Ranger and custom data governance solutions like Privacera or Immuta offer dynamic data access control features that can be used in pipeline-linked analytical systems.

E. Audit Logging and Non- Repudiation

The performance of full audit recording of pipeline accesses is important to the investigations of security incidents as well as the demonstration of compliance with regulations. The audit logs should record the main identity, time, type of action, resource that was accessed in a pipeline, and the result of the access decision made on the entity of every important pipeline operation. Non-repudiation is offered by immutable audit logs in which the use of append-only storage along with cryptographic hash chaining ensures that past alterations to access logs cannot be made.

7. PROPOSED INTEGRATED SECURITY CONTROL FRAMEWORK (ISCF)

The PII masking, encryption, and access control framework suggested in this paper integrates the three components in one layered reference architecture that can be used by both stream-based and batch ingestion pipeline architecture.

A. Framework Architecture

The ISCF architecture has three vertically integrated control planes that are consistent with the main pipeline security requirements:

Data Protection Plane: PII masking, field-level encryption as well as data classification. Administers TLS certificate policies, assists with KMS to assist in DEK/KEK policy, and encrypted policy compliance by conducting pre-ingestion verification checks to deny records sent via non-encrypted transport routes [28].

Access Governance Plane: This implements hybrid RBAC/ABAC access control as a pipeline stage boundary by using OPA-based policy evaluation. Maintains a dynamic data sensitivity catalogue, which contaminates masking and access control decisions and mass audit documentation with tamper-evident storage of logs.

There are four cross-cutting concerns to all three control planes, which are (1) Observability, which is used to ensure that all the security control performance is monitored and that all policy violation notifications are provided; (2) Configuration Management, which is used to ensure that all the security control configuration is consistent across distributed components of the pipeline; (3) Regulatory Compliance Mapping, which is used to maintain a clear mapping of ISCF controls to regulatory requirements to facilitate compliance reporting; and (4) Key Management Integration, which is used to provide a consistent interface to underlying KMS systems

B. Regulatory Compliance Strategy

ISCF is expected to prove the adherence to the GDPR Article 32 (technical measures to protect the data), HIPAA Security Rule 164.312 (technical safeguards), and PCI-DSS Requirement 3 (stored cardholder data protection) and Requirement 4 (transmission encryption). The automated PII discovery and masking features of the framework directly respond to the GDPR data minimization principles, whereas field-level encryption offers the protection of the cardholder data mandated by the PCI-DSS. Extensive audit logging facilitates the GDPR accountability requirements, as well as the HIPAA access control audit trail requirements. Table III aligns ISCF controls with certain regulatory provisions.

TABLE III: ISCF Control-to-Regulation Mapping

ISCF Control	**GDPR**	**HIPAA**	**PCI-DSS**
PII Discovery & Masking	Art. 5(1)(c), Art. 25	§164.514	Req. 3

Transport Encryption (TLS)	Art. 32(1)(a)	§164.312(e)(1)	Req. 4.2
Field-Level Encryption	Art. 32(1)(a)	§164.312(a)(2)(iv)	Req. 3.5
RBAC/ABAC Access Control	Art. 32(1)(b)	§164.312(a)(1)	Req. 7
Audit Logging	Art. 30, Art. 5(2)	§164.312(b)	Req. 10
Key Management	Art. 32(1)(a)	§164.312(a)(2)(iv)	Req. 3.6

8. EXPERIMENTAL EVALUATION

A. Experimental Setup

The ISCF was tested and deployed on a test cluster of eight nodes that had 32 GB RAM, 16 vCPU cores, and 1 Gbps network interconnect, and ran Ubuntu 22.04 LTS. Apache Kafka 3.5.0 has been used as the pipeline substrate to ingest the streams and Apache Spark 3.4.1 to transform the batch. OPA 0.57.0 was set up as a centralised policy decision point and had a special policy evaluation sidecar on each worker node of a pipeline. Cryptographic key management was done using AWS KMS, and tokenization was done using a locally deployed Vault instance that offered PII masking operations.

B. Performance Results

Table IV presents the throughput overheads that were added by each ISCF control layer and the entire integrated architecture on both workloads. In the streaming financial transactions workload, the combined ISCF overhead was 7.3 that is less than the 10 percent overhead threshold that had been found to be tolerable in the previous literature as being used in the production pipeline deployments [14]. PII masking with FPE had the highest individual overhead 3.8% followed by OPA policy evaluation 2.1 and field-level encryption 1.4. There was negligible overhead due to transport-layer encryption (Less than 0.1%).

TABLE IV: ISCF by Control Layer Throughput Overhead

Control Layer	**Stream Overhead (%)**	**Batch Overhead (%)**	**Notes**
Transport Encryption (TLS 1.3)	< 0.1	< 0.1	HW-accelerated
PII Masking (FPE)	3.8	4.2	23 PII fields
Field-Level Encryption (AES-GCM)	1.4	1.6	Sensitive fields only
Access Governance (OPA)	2.1	1.8	Centralised + sidecar
Audit Logging	0.7	0.4	Async append-only
Total ISCF Overhead	7.3	8.1	Combined framework

In the case of the batch healthcare workload, the total ISCF overhead was 8.1 per cent, which was due to the increased number of PII fields per record (41 fields instead of 23). In the healthcare batch workload, PII masking overhead of 4.2% indicates the increased concentration of sensitive fields, and this overhead is predicted to be proportional to the number of PII fields depending on the complexity of per-field FPE work.

C. PII Coverage and Accuracy

The accuracy of automated PII detection was tested with respect to a labelled reference set of 10, 000 records of each workload. The ISCF PII discovery engine obtained a recall of 97.4% and 95.8% with a precision of 94.3 with

the financial transactions dataset and 96.1 with the healthcare dataset. The more severe failure mode is false negatives whereby PII fields have not been identified and thus it is not masked; the obtained recall rates are comparable with production-grade PII discovery in a financial services regulatory environment [29].

9. DISCUSSION

The experimental findings indicate that a holistic security control integration in the data ingestion pipelines can be executed within an operationally reasonable performance limit. The 7.3% total throughput overhead of the streaming workload and 8.1% of the batch workload are significantly lower than the common belief that effective security in the pipeline must be prohibitive to performance [14]. The practical relevance of this finding is important: organisations that have not invested in pipeline security based on their perceived performance trade-offs can discover that modern cryptographic hardware acceleration and policy evaluation structure optimization can reduce the practical cost of full pipeline security measures by a substantial margin.

Breaking down the overhead of ISCF shows that PII masking with format-preserving encryption is the largest performance cost at about 52 percent of overall framework overhead in the streaming test. The RBAC/ABAC access governance model that was executed over OPA showed good policy expressiveness at a reasonable overhead contribution of 2.1%. Nevertheless, policy compilation and caching was identified as crucial to high-velocity streaming workloads with the deployment of the test: without results of policy evaluation caching across repeated principal-resource-action combinations, OPA evaluation overhead rose to 6.8% making the overall ISCF overhead of 11.4% operationally insignificant. This observation supports other studies [30] on the significance of policy evaluation optimization to high-throughput access control.

Table III above shows that the ISCF controls are well-coordinated to meet the fundamental technical stipulations of GDPR, HIPAA, and PCI-DSS. It should also be mentioned, though, that in data pipeline situations the matter of regulatory compliance is not limited to the technical controls, but also includes the governance processes, documentation demands, and organizational accountability schemes that are outside of the current technical framework. Organizations implementing ISCF must complement it with relevant data governance procedures, data protection impact assessment (DPIAs) on high-risk processing operation, and periodic security control reviews that are in accordance with its regulatory requirements.

Another serious weakness of a current work is the fact that the evaluation setting is based on synthetic datasets, which, although intended to reflect a realistic production situation involving PII field distributions, might not become representative enough to represent the variety of production data that would be seen in large-scale enterprise pipeline implementation. Also, the current test did not evaluate the performance of the framework when subjected to adversarial situations; further research ought to integrate red team testing to confirm the strength of ISCF controls in the face of active attack situations such as injecting PII, evading policy, and attempting encrypted channel compromise.

10. CONCLUSION

The paper has introduced the Integrated Security Control Framework (ISCF), a standardized reference system to incorporate the data ingestion pipelines with PII masking, encryption, and access governance controls. The paper established the disintegration of the previous work in these three domains of security and, as a result, the lack of integrated, pipeline-specific security models through a recent review of related literature and extensive analysis of the security threats specific to the pipeline.

To fill this gap, the proposed ISCF provides a three-plane control architecture including Data Protection, Transport Security, and Access Governance planes, which are combined at four different points of defining pipeline enforcement. Apache Kafka and Apache Spark deployment evaluation showed that the full ISCF overheads streaming workloads and batch workloads by 7.3% and 8.1%, respectively, operationally acceptable overheads that undermine the currently accepted viewpoint of the fundamental incompatibility between comprehensive security controls and pipeline performance.

The framework is demonstrably compliant with GDPR, HIPAA, and PCI-DSS requirements in six major categories of controls giving organizations a systematic foundation on which to justify compliance in regulatory terms. With growing regulatory attention to data pipeline security practices across the world, the combined strategy to pipeline security shown by ISCF will be all the more vital to accountable data engineering practice.

REFERENCES

1. M. Kleppmann, "Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems," O'Reilly Media, Inc., 2017. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=p1heDgAAQBAJ&oi=fnd&pg=PP15&dq=J.+Kleppmann (accessed Mar. 23, 2026).
2. The Hacker News, "Cost of a Data Breach Report 2023: Insights, Mitigators and Best Practices," The Hacker News, Dec. 21, 2023. [Online]. Available: https://thehackernews.com/2023/12/cost-of-data-breach-report-2023.html
3. I. Makhdoom, M. Abolhasan, J. Lipman, N. Shariati, D. Franklin, and M. Piccardi, "Securing Personally Identifiable Information: A Survey of SOTA Techniques, and a Way Forward," IEEE Access, vol. 12, pp. 116740–116770, 2024, doi: https://doi.org/10.1109/access.2024.3447017.
4. R. Ghazal, A. K. Malik, N. Qadeer, B. Raza, A. R. Shahid, and H. Alquhayz, "Intelligent Role-Based Access Control Model and Framework Using Semantic Business Roles in Multi-Domain Environments," IEEE Access, vol. 8, pp. 12253–12267, 2020, doi: https://doi.org/10.1109/access.2020.2965333.
5. K. Vidyasankar, "On Continuous Queries in Stream Processing," Procedia Computer Science, vol. 109, pp. 640–647, Jun. 2017, doi: https://doi.org/10.1016/j.procs.2017.05.370.
6. S. Karagiannis, C. Ntantogian, E. Magkos, A. Tsohou, and L. L. Ribeiro, "Mastering data privacy: leveraging K-anonymity for robust health data sharing," International Journal of Information Security, Mar. 2024, doi: https://doi.org/10.1007/s10207-024-00838-8.
7. S. Yoo, M. Shin, and D. Lee, "An Approach to Reducing Information Loss and Achieving Diversity of Sensitive Attributes in k-anonymity Methods," Interactive Journal of Medical Research, vol. 1, no. 2, p. e14, Nov. 2012, doi: https://doi.org/10.2196/ijmr.2140.
8. A. Majeed and S. Lee, "Anonymization techniques for privacy preserving data publishing: A comprehensive survey," IEEE Access, vol. 9, pp. 8512–8545, 2020, doi: https://doi.org/10.1109/access.2020.3045700.
9. E. G. Reddy, "Data Privacy & Encryption Practices in Cloud-Based Guidewire Deployments," International Journal of AI, BigData, Computational and Management Studies, vol. 2, 2021, doi: https://doi.org/10.63282/3050-9416.ijaibdcms-v2i3p108.
10. P. Garg, A. Gupta, Y. P. Singh, and P. Goyal, "End-to-End Encryption: Evolution, Barriers, and Emerging Trends," Signals and Communication Technology, pp. 113–127, 2026, doi: https://doi.org/10.1007/978-981-95-1683-4_8.
11. AWS Amazon, "Introducing Amazon ECR server-side encryption using AWS Key Management System | Amazon Web Services," Amazon Web Services, Jul. 29, 2020. [Online]. Available: https://aws.amazon.com/blogs/containers/introducing-amazon-ecr-server-side-encryption-using-aws-key-management-system/ (accessed Mar. 23, 2026).
12. A. D. Khamis, "Security Framework for Distributed Database System," Journal of Data Analysis and Information Processing, vol. 07, no. 01, pp. 1–13, 2019, doi: https://doi.org/10.4236/jdaip.2019.71001.
13. R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: towards a unified standard," 2000.
14. O. Vakhula, I. Opirskyy, P. Vorobets, O. Bobko, and O. Kulinich, "Research on Policy-as-Code for Implementation of Role-based and Attribute-based Access Control," 2025. [Online]. Available: https://ceur-ws.org/Vol-3991/paper11.pdf
15. S. Ameer, J. Benson, and R. Sandhu, "Hybrid Approaches (ABAC and RBAC) Toward Secure Access Control in Smart Home IoT," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 1–18, 2022, doi: https://doi.org/10.1109/tdsc.2022.3216297.
16. A. Wider, S. Verma, and A. Akhtar, "Decentralized Data Governance as Part of a Data Mesh Platform: Concepts and Approaches," Jul. 2023, doi: https://doi.org/10.1109/icws60048.2023.00101.
17. A. Mavi, "Security Automation Framework for Digital Twins in Smart Manufacturing," Journal of Information Systems Engineering and Management, vol. 10, no. 33s, pp. 1088–1098, Mar. 2025, doi: https://doi.org/10.52783/jisem.v10i33s.5848.
18. J. Davis-Turak et al., "Genomics pipelines and data integration: challenges and opportunities in the research setting," Expert Review of Molecular Diagnostics, vol. 17, no. 3, pp. 225–237, Jan. 2017, doi: https://doi.org/10.1080/14737159.2017.1282822.
19. S. S. G. Venkata, "Secure software development: integrating encryption protocols from design to deployment," International Journal of Applied Mathematics, vol. 38, no. 2s, pp. 1190–1213, Oct. 2025, doi: https://doi.org/10.12732/ijam.v38i2s.714.
20. A. Satyanarayanan, "Optimizing Data Quality in Real-Time: A Self-Healing Pipeline Approach," International Journal of AI, BigData, Computational and Management Studies, vol. 3, no. 2, Jun. 2022, doi: https://doi.org/10.63282/3050-9416.ijaibdcms-v3i2p107.

21. D. Kim, H. Kim, K. Jang, S. Yoon, and H. Seo, “Deep-Learning-Based Neural Distinguisher for Format-Preserving Encryption Schemes FF1 and FF3,” Electronics, vol. 13, no. 7, p. 1196, Mar. 2024, doi: https://doi.org/10.3390/electronics13071196.
22. N. Ponomareva et al., “How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy,” Journal of Artificial Intelligence Research, vol. 77, pp. 1113–1201, Jul. 2023, doi: https://doi.org/10.1613/jair.1.14649.
23. R. Gangarde, A. Sharma, A. Pawar, R. Joshi, and S. Gonge, “Privacy Preservation in Online Social Networks Using Multiple-Graph-Properties-Based Clustering to Ensure k-Anonymity, l-Diversity, and t-Closeness,” Electronics, vol. 10, no. 22, p. 2877, Nov. 2021, doi: https://doi.org/10.3390/electronics10222877.
24. J. Hallin, “Evaluation of TLS and mTLS in Internet of things systems,” DIVA, 2025. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1937634 (accessed May 05, 2025).
25. G. V. Rajan, R. Mittal, and A. Gupta, “Secured Cloud Storage with Client Side Encryption,” SSRN Electronic Journal, 2025, doi: https://doi.org/10.2139/ssrn.5870642.
26. A. I. Khan, H. Saud, Syed, and M. I. Fakhir, “Database Security Empowered with Independent Field Encryption,” VAWKUM Transactions on Computer Sciences, vol. 12, no. 1, pp. 125–135, 2024, doi: https://doi.org/10.21015/vtcs.v12i1.1775.
27. U. T. Rangani, “Implementing data classification at the database schema level for data categorisation, identification and monitoring: Enhancing data security,” Theseus.fi, 2025, doi: https://www.theseus.fi/handle/10024/891326.
28. T. C. Herath, H. S. B. Herath, and D. Cullum, “An Information Security Performance Measurement Tool for Senior Managers: Balanced Scorecard Integration for Security Governance and Control Frameworks,” Information Systems Frontiers, vol. 25, Feb. 2022, doi: https://doi.org/10.1007/s10796-022-10246-9.
29. R. Alonso, F. Castaño, Y. J. Cruz, A. Villalonga, and R. E. Haber, “Early defect detection in thermo-insulating panels: A case study on quality inspection,” The International Journal of Advanced Manufacturing Technology, vol. 142, no. 1–2, pp. 493–509, Dec. 2025, doi: https://doi.org/10.1007/s00170-025-17067-5.
30. M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Zakeri Fardi, and S. Samad, “Authentication systems: A literature review and classification,” Telematics and Informatics, vol. 35, no. 5, pp. 1491–1511, Aug. 2018, doi: https://doi.org/10.1016/j.tele.2018.03.018.