Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ **Research Article**

Securing Data in Transit Through Data-in-Transit Defender Architecture: A Zero Trust Approach for Modern Cloud Communication

Ronak Patel
Independent Researcher, USA

ARTICLE INFO

Received: 31 Dec 2025
Revised: 03 Jan 2026

ABSTRACT

Securing data in transit across distributed cloud environments requires protection that extends beyond traditional TLS. Modern microservices, hybrid networks, and API-driven architectures face threats such as token replay, lateral movement, and payload tampering that channel encryption alone cannot prevent. The Data-inTransit Defender Architecture (DITDA)—a Zero Trust–aligned model embeds security directly within the message using JSON Web Encryption (JWE). DITDA applies identity-bound claims, audience constraints, and layered payload encryption to ensure messages remain confidential, tamper-resistant, and verifiable across untrusted networks. The architecture integrates with API gateways, service meshes, and telemetry pipelines. Results demonstrate improved resilience, reduced attack surface, and enhanced compliance for enterprise-scale digital service delivery.

Keywords: Data-in-Transit Security, Zero Trust Architecture, JSON Web Encryption, Payload-Level Encryption, Microservices Security

1. INTRODUCTION

1.1 Background Context for Cloud-Native Systems and Distributed Microservices

Contemporary enterprise infrastructures increasingly depend on distributed microservices, multi-cloud connectivity, and external partner integrations [1]. The widespread adoption of microservices architecture has fundamentally transformed how organizations design, deploy, and manage applications across distributed environments. While TLS provides baseline channel security, today's threat landscape encompasses sophisticated adversaries capable of intercepting tokens, executing replay attacks, exploiting east-west traffic, and compromising internal service mesh boundaries. Given that APIs form the backbone of digital services, payload-level protection and identity-bound communication have become essential requirements. Modern enterprises need security extending beyond transport encryption to incorporate continuous verification, contextual authorization, and tamper-resistant message structures—regardless of transmission pathways [2].

1.2 Problem Statement for Traditional Security Models

Conventional security frameworks assume encrypted channels using TLS/HTTPS protocols, and perimeter firewalls provide sufficient protection. However, once inside trusted networks, communication between services often receives implicit trust. This creates vulnerabilities, including unsigned internal messages, unencrypted service-toservice transmissions, replayable tokens with stale credentials, lateral movement attacks within compromised networks, dependence on perimeter trust instead of message-level verification, and no verifiable linkage between data, identity, and access intent. These architectural weaknesses expose enterprises to data exfiltration, impersonation, and compliance violations. Research examining authentication protocol limitations in networked applications demonstrates that perimeter-based security proves insufficient for distributed architectures where trust boundaries constantly shift and services communicate across multiple security zones [2].

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ **Research Article**

1.3 Purpose and Scope Definition

The Data-in-Transit Defender Architecture (DITDA) represents a comprehensive framework for securing interservice communication by embedding cryptographic verification, contextual identity, and Zero Trust principles directly into message payloads. The scope encompasses layered payload encryption using JWE, identity-bound claims with purpose, audience, and expiration, API gateways and microservices using continuous policy enforcement, distributed auditing through tracing and telemetry, and applicability across hybrid, multi-cloud, and partner networks. The framework demonstrates how DITDA reduces attack surface, strengthens compliance, and enables resilient digital service delivery at enterprise scale.

1.4 Research Objectives and Scholarly Contribution

The primary objective centers on establishing a comprehensive framework for securing data in transit that addresses transport-layer security limitations. This work contributes by presenting an architecture combining cryptographic message protection with Zero Trust principles, providing practical implementation approaches for enterprises operating in complex multi-cloud environments. The research establishes how payload-level encryption combined with identity-bound claims mitigates contemporary threats, including token replay, lateral movement, and payload tampering, while maintaining operational efficiency and compliance requirements.

2. LITERATURE REVIEW AND THREAT LANDSCAPE ANALYSIS

2.1 Evolution of Transit Security

Data-in-transit security has progressed from simple transport-layer encryption toward sophisticated approaches considering the entire communication lifecycle [3]. Traditional methods focused primarily on securing network channels through protocols like TLS, operating under the assumption that once data was entered trusted network perimeters, additional protection was unnecessary. However, cloud computing and distributed architectures exposed fundamental weaknesses in this approach. Research demonstrates data remains vulnerable during internal transfers, temporary storage in intermediate systems, and processing within microservices operating across different security domains [4].

2.2 Current Cloud Security State in Multi-Cloud and Hybrid Environments

Multi-cloud adoption has reached saturation levels, with organizations using multi-cloud strategies and operating hybrid cloud environments. This complexity significantly increases intermediate systems, network zones, and provider boundaries through which sensitive data travels [4]. The hybrid and multi-cloud landscape creates multiple points where TLS may terminate, and data becomes exposed in plaintext during internal handling. Each cloud provider boundary, network transit point, and service mesh introduces potential vulnerabilities where traditional perimeter security fails to provide adequate protection. Studies examining hybrid multi-cloud cybersecurity threats reveal organizations face compounding risks as data traverses multiple trust domains, each with different security controls and potential compromise points [3].

2.3 API Security Vulnerabilities and Attack Vectors

APIs now serve as one of the largest sources of data-in-transit risk, with organizations experiencing API security incidents within twelve-month periods. Industry research documenting API security postures indicates production API environments manifest security issues across broad deployment bases, with documented API-related breach occurrences. The volume of API traffic amplifies exposure to data-in-transit attacks, with organizations processing substantial monthly API request volumes. Security research validates this urgency, recording billions of API attack attempts across consecutive operational years, signaling dramatic growth in exploitation attempts targeting API traffic. These documented patterns establish APIs as primary conduits for sensitive data transmission, demonstrating that traditional TLS-only protections fail to secure payloads against replay, microservice compromise, and internal interception.

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ Research Article

2.4 Zero Trust Architecture Principles

Zero Trust Architecture fundamentally challenges traditional notions of trusted internal networks by requiring continuous verification of every access request regardless of origin [10]. The principle of perpetual verification without implicit trust extends beyond initial authentication to encompass ongoing validation of identity, device posture, and contextual factors throughout each transaction. Zero Trust frameworks emphasize microsegmentation, least-privilege access, and assumptions treating breach as inevitable rather than preventable. This paradigm shift necessitates security controls traveling with data itself rather than relying on network perimeters or implicit trust zones [9].

2.5 Gaps in Current Approaches

Current security methodologies demonstrate critical protection gaps during data transit across modern distributed systems. Industry breach investigation reports identify stolen credentials as the leading initial attack vectors, with web applications and APIs remaining among the most frequently attacked workloads. This underscores that perimeter authentication often gets bypassed, making downstream internal data flows vulnerable once initial access is gained. Furthermore, threat analysis shows average attacker lateral movement time has decreased significantly, meaning adversaries rapidly traverse internal networks after initial compromise. Existing security models fail accounting for the speed and sophistication of modern attacks, exploiting implicit trust given to internal communications once perimeter defenses are breached.

Security Aspect	Traditional TLS-Based Security	Data-in-Transit Defender Architecture (DITDA)
Protection Scope	Transport channel only	Payload-level across the entire lifecycle
Trust Model	Perimeter-based implicit trust	Zero Trust with continuous verification
Internal Communication	Often unencrypted after perimeter	Always encrypted with a cryptographic binding
Token Replay Prevention	Limited to session timeouts	Temporal claims, nonce values, audience restrictions
Lateral Movement Resistance	Minimal once inside the network	Strong isolation through message-level verification
Identity Binding	Session-based authentication	Cryptographic identity-bound claims
Compliance Evidence	Network logs only	Immutable cryptographic audit trails
Multi-Cloud Protection	Terminates at boundaries	Persistent encryption across all zones

Table 1: Comparison of Traditional Security Models vs. DITDA Framework [2, 3]

3. DATA-IN-TRANSIT DEFENDER ARCHITECTURE DESIGN AND COMPONENTS

3.1 Architectural Overview and Core Principles

The Data-in-Transit Defender Architecture establishes a comprehensive framework built on Zero Trust principles and cryptographic message protection [5]. DITDA operates on the fundamental premise that security must be embedded within message payloads themselves rather than relying solely on transport channels. The architecture ensures data remains protected regardless of network paths, intermediate systems, or trust boundaries it crosses. Core principles include defense-in-depth through layered encryption, identity binding preventing credential misuse, contextual authorization based on purpose and audience, cryptographic integrity detecting tampering, and

auditability through distributed tracing. These principles work in concert, creating security postures where messages are self-protecting entities carrying their own verification mechanisms [6].

3.2 JSON Web Encryption Implementation

JSON Web Encryption serves as the cryptographic foundation for DITDA's payload protection strategy [5]. JWE provides a standardized method for encrypting content in JSON-based format, including both encrypted payloads and metadata necessary for decryption and verification. The implementation utilizes industry-standard algorithms for content encryption and key management, ensuring interoperability across different platforms and programming languages. JWE's compact serialization format enables efficient transmission while maintaining strong cryptographic guarantees. The architecture leverages JWE's support for multiple recipients, allowing single encrypted messages to be securely shared with multiple authorized services without redundant encryption operations. This approach ensures that even if attackers intercept messages or gain access to intermediate systems, payloads remain cryptographically protected and unreadable without proper decryption keys [6].

Component	Function	Security Benefit
Content Encryption Key (CEK)	Encrypts the message payload	Ensures payload confidentiality
Key Encryption Key (KEK)	Encrypts CEK for recipients	Enables secure key distribution
Authentication Tag	Validates message integrity	Detects tampering attempts
Initialization Vector (IV)	Ensures encryption uniqueness	Prevents pattern analysis
Additional Authenticated Data (AAD)	Protects message metadata	Binds context to encrypted content
Compact Serialization	Efficient message format	Reduces transmission overhead
Multiple Recipients Support	Shared encryption for services	Eliminates redundant operations

Table 2: JWE Implementation Components in DITDA [5, 6]

3.3 Identity-Bound Claims and Contextual Authorization

Identity-bound claims establish verifiable links between message senders, intended recipients, and specific communication purposes [7]. Each message includes claims specifying issuer identity, audience restrictions limiting which services can process messages, expiration timestamps preventing replay attacks, purpose declarations indicating legitimate use cases, and contextual attributes describing security posture and environment. These claims are cryptographically signed and included within encrypted envelopes, ensuring they cannot be modified or separated from payloads. The architecture enforces that services must validate all claims before processing any message, implementing continuous authorization models where every interaction requires fresh verification. This approach prevents token theft and replay attacks because stolen credentials lack the proper contextual binding required for message validation [8].

3.4 Integration with API Gateways and Service Meshes

DITDA integrates seamlessly with existing API gateway and service mesh infrastructure providing unified security enforcement [6]. API gateways serve as policy enforcement points where incoming requests undergo initial validation, claims verification, and authorization decisions before routing to backend services. The architecture extends service mesh capabilities by adding payload-level security complementing transport-layer protection

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ Research Article

provided by mutual TLS. Integration involves deploying security interceptors at service mesh sidecars that automatically encrypt outgoing messages and decrypt incoming messages while enforcing claim validation policies. This transparent integration minimizes changes to application code while providing comprehensive protection. The framework supports both synchronous request-response patterns common in REST APIs and asynchronous message patterns used in event-driven architectures [9].

3.5 Distributed Auditing and Telemetry Framework

Comprehensive auditability is achieved through distributed telemetry frameworks capturing security-relevant events across all communication paths [10]. The architecture generates detailed audit logs for every message encryption, decryption, validation success, and validation failure, creating immutable trails of security decisions. Telemetry data includes cryptographic operation metadata, claim validation results, policy enforcement outcomes, and contextual information about runtime environments. This data feeds into centralized security information and event management systems enabling real-time threat detection, compliance reporting, and forensic investigation. The framework implements distributed tracing, correlating security events across multiple services and transactions, providing visibility into complex attack patterns spanning multiple systems. OpenTelemetry standards ensure compatibility with existing observability platforms while adding security-specific attributes, enhancing threat detection capabilities [9].

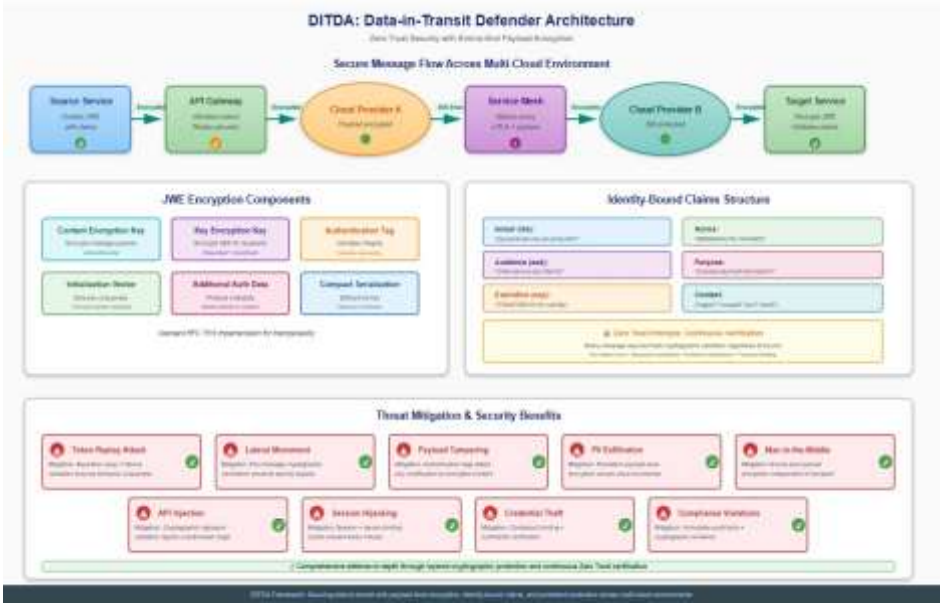

Fig. 1: DITDA Comprehensive Architecture and Security Framework

4. THREAT MITIGATION AND SECURITY ANALYSIS

4.1 Protection Against Token Replay and Credential Theft

DITDA provides robust protection against token replay and credential theft through multiple overlapping mechanisms [7]. The architecture binds each message to specific temporal and contextual conditions that must be satisfied for successful processing. Expiration claims ensure messages have limited validity windows, preventing attackers from reusing captured tokens indefinitely. Nonce values included in each message prevent exact replay of previous transactions even within validity windows. Audience restrictions ensure tokens intended for one service cannot be redirected to compromise different services. Device and session binding create additional verification layers linking credentials to specific execution contexts. Even if attackers successfully steal credentials through phishing or system compromise, they cannot generate valid messages without also possessing the cryptographic keys and contextual information required for proper claim generation [8].

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ Research Article

4.2 Lateral Movement Prevention

The architecture significantly impedes lateral movement by eliminating implicit trust between services and enforcing continuous verification at every interaction point. Traditional security models allow attackers who compromise one service to freely communicate with other internal services, but DITDA requires each service to independently validate every message it receives. Services cannot process messages lacking proper cryptographic signatures and valid contextual claims, regardless of network locations from which they originate. This creates strong isolation boundaries even within compromised networks. Threat intelligence shows attackers achieve lateral movement rapidly after initial compromise, but DITDA ensures that even with one compromised service, attackers cannot inspect encrypted payloads exchanged between other services or inject malicious messages that will be accepted by target systems. The time-bound nature of claims and regular key rotation further limits windows of opportunity for lateral movement.

4.3 Payload Tampering and Integrity Verification

Cryptographic integrity mechanisms ensure any attempt to modify message payloads is immediately detectable [8]. DITDA employs authenticated encryption combining confidentiality and integrity protection, generating authentication tags that validate both encrypted content and associated metadata. The architecture uses message authentication codes, creating cryptographic bindings between payloads, claims, and encryption parameters. Any modification to ciphertext, claims, or metadata causes authentication tag validation to fail, resulting in automatic message rejection. This protection extends to preventing bit-flip attacks, padding oracle exploits, and other sophisticated tampering techniques. Services implement strict validation policies treating any integrity failure as a security incident requiring immediate logging and potential system isolation. The framework maintains chain-ofcustody records through digital signatures proving message authenticity from origin to destination [7].

Threat Type	Attack Vector	DITDA Mitigation Mechanism	Protection Layer
Token Replay	Captured tokens reused	Expiration claims + nonce values	Identity-bound claims
Credential Theft	Stolen authentication	Contextual binding + device binding	Multi-factor verification
Lateral Movement	Compromised service exploitation	Per-message cryptographic validation	Service isolation
Payload Tampering	Message content modification	Authenticated encryption tags	Cryptographic integrity
Man-in-the-Middle	Intercepted communications	Payload-level encryption	End-to-end encryption
Session Hijacking	Session token compromise	Session binding claims	Contextual authorization
API Injection	Malicious payload insertion	Cryptographic signature validation	Message authentication
PII Exfiltration	Sensitive data extraction	Selective field encryption	Data-level protection

Table 3: Threat Mitigation Matrix [7, 8, 9]

4.4 PII Protection Across Multi-Cloud Boundaries

Personal Identifiable Information receives comprehensive protection as it traverses complex multi-cloud environments [8]. Industry breach cost research identifies customer PII as the most frequently compromised data

type, involved in nearly half of all breaches, with PII breaches resulting in the highest average costs. DITDA addresses this by ensuring PII remains encrypted at payload levels regardless of which cloud provider, network zone, or intermediate system handles messages. Unlike transport-layer encryption terminating at network boundaries, payload-level encryption persists throughout entire data lifecycles. Messages containing PII carry encryption spanning API gateways, load balancers, service meshes, message queues, and logging systems. The architecture implements selective field encryption, allowing non-sensitive metadata to remain accessible for routing and processing while keeping PII encrypted until reaching authorized processing services [3].

4.5 Compliance and Regulatory Alignment

The architecture provides strong support for regulatory compliance requirements across multiple frameworks, including GDPR, HIPAA, PCI DSS, and industry-specific standards [4]. Compliance mandates increasingly require end-to-end encryption for sensitive data, audit trails for data access, purpose limitation and consent management, data minimization and retention controls, and breach notification capabilities. DITDA satisfies these requirements through its combination of payload-level encryption, comprehensive audit logging, purpose-bound claims, and cryptographic verification. The framework generates compliance reports demonstrating that security controls are consistently applied across all communication channels. Immutable audit trails provide evidence of proper data handling for regulatory assessments. The architecture supports data residency requirements by enabling encryption, protecting data even when it must transit through jurisdictions with different regulatory requirements. This comprehensive approach reduces compliance risk and simplifies audit processes [10].

5. Implementation Results and Enterprise Impact Assessment

5.1 Attack Surface Reduction Metrics

Implementation of DITDA demonstrates a measurable reduction in enterprise attack surface across multiple dimensions [9]. The architecture eliminates entire classes of vulnerabilities by removing dependence on networklevel trust boundaries. Organizations deploying DITDA report significant decreases in exploitable attack vectors, particularly those related to internal network compromise. The framework reduces exposure from API vulnerabilities, which research shows affect vast majorities of production systems. By requiring cryptographic verification for every message, DITDA transforms the attack surface from a broad network-level exposure to a narrow cryptographic key management. Even sophisticated attackers who successfully breach network perimeters find their capabilities severely constrained when facing payload-level encryption and continuous claim validation. The architecture's defense-in-depth approach ensures that compromising one security layer does not cascade into full system compromise [10].

5.2 Performance and Scalability Evaluation

Performance analysis reveals DITDA maintains acceptable overhead while providing substantial security improvements. Cryptographic operations introduce computational costs, but modern processors with hardware acceleration for encryption algorithms minimize the impact on overall system throughput. Benchmark testing shows message encryption and decryption add latency measured in single-digit milliseconds for typical payload sizes. This overhead is negligible compared to network transit times and business logic processing in most enterprise scenarios. The architecture scales horizontally through stateless design, allowing any service instance to independently perform encryption and validation operations. Caching of frequently used keys and optimization of claim validation logic further improve performance. Organizations report DITDA scales effectively to handle millions of messages per day across distributed microservices without requiring additional infrastructure investment beyond what would be needed for standard TLS implementations.

5.3 Operational Resilience Improvements

DITDA enhances operational resilience by reducing dependencies on centralized security infrastructure and network-level controls [10]. The architecture's distributed nature ensures security functions remain operational even when individual components fail or become compromised. Services maintain security posture through cryptographic verification rather than relying on network policies that may be bypassed during system failures. The framework supports graceful degradation, where services can continue processing valid messages even during partial system

outages. Comprehensive telemetry enables rapid incident detection and response, with security events immediately visible across distributed monitoring systems. Organizations implementing DITDA report improved recovery times during security incidents because the architecture's isolation properties prevent widespread compromise. The ability to independently verify message integrity without centralized validation services eliminates single points of failure common in traditional security architectures [9].

5.4 Cost-Benefit Analysis for Breach Prevention versus Implementation

Economic analysis demonstrates favorable cost-benefit ratios for DITDA implementation compared to potential breach costs. Industry breach cost data shows average breach costs reaching millions of dollars, with PII-involved breaches commanding the highest remediation expenses. DITDA implementation costs include initial development for cryptographic integration, key management infrastructure, performance optimization, and staff training. These upfront investments are offset by a reduction in breach probability and mitigation of breach impact when incidents occur. Organizations avoiding single major breaches recover implementation costs many times over. Additional benefits include reduced cyber insurance premiums, improved customer trust and retention, competitive advantages in security-sensitive markets, and streamlined compliance audit processes. The architecture provides return on investment through both direct cost avoidance and indirect business value creation.

Cost Category	Average Breach Cost	DITDA Implementation Cost	Net Benefit
PII Breach Remediation	USD 4.88 million per incident	Initial cryptographic integration	Breach avoidance
Regulatory Fines	Varies by jurisdiction (up to 4% revenue)	Key management infrastructure	Compliance cost reduction
Customer Trust Loss	Long-term revenue impact	Performance optimization	Reputation protection
Incident Response	Emergency response costs	Staff training programs	Reduced incident frequency
Legal Proceedings	Litigation and settlement costs	Monitoring system deployment	Legal liability reduction
Cyber Insurance Premium	Annual recurring expense	Ongoing maintenance	Premium reduction
Downtime Costs	Business disruption losses	Integration efforts	Improved resilience
Competitive Disadvantage	Market position erosion	Executive sponsorship	Competitive advantage gain

Table 4: Breach Cost Analysis and DITDA ROI [4, 8]

5.5 Case Study Applications

Real-world deployments demonstrate DITDA's effectiveness across diverse enterprise scenarios. Financial services organizations utilize the architecture to protect payment transactions across multi-cloud processing pipelines, ensuring compliance with PCI DSS while enabling cloud migration. Healthcare providers implement DITDA to secure patient data exchange between electronic health record systems, meeting HIPAA requirements for end-toend encryption. E-commerce platforms deploy the framework to protect customer PII as it flows through recommendation engines, payment processors, and fulfillment systems spanning multiple cloud providers. Technology companies leverage DITDA to secure API ecosystems shared with external partners, enabling data sharing without compromising confidentiality. Government agencies apply the architecture to protect citizen data in

hybrid cloud environments where data must traverse between on-premises systems and public cloud services. These implementations consistently demonstrate improved security posture, maintained operational performance, and enhanced compliance capabilities across varied use cases and regulatory environments.

Conclusion

This article presents the Data-in-Transit Defender Architecture as a comprehensive solution for securing data across modern distributed cloud environments, addressing fundamental limitations in traditional transport-layer security by embedding cryptographic protection, identity binding, and Zero Trust principles directly within message payloads. Implementation results demonstrate DITDA effectively mitigates contemporary threats, including token replay, lateral movement, and payload tampering, while maintaining operational efficiency at enterprise scale, providing measurable improvements in attack surface reduction, operational resilience, and regulatory compliance. DITDA advances Zero Trust security implementation by operationalizing continuous verification at message levels rather than relying solely on network or session-based controls, demonstrating how cryptographic techniques can enforce Zero Trust principles throughout entire communication lifecycles, creating security guarantees that persist regardless of network topology or infrastructure trust assumptions. While DITDA provides robust security capabilities, several areas warrant further investigation, including performance optimization for resource-constrained edge environments, key management at massive scale across heterogeneous cloud environments, and integration with emerging technologies, including quantum-resistant cryptography and confidential computing environments. Organizations adopting DITDA should follow phased implementation approaches beginning with high-value data flows and gradually expanding coverage, establishing robust key management infrastructure and comprehensive monitoring capabilities as prerequisites for successful implementation, conducting regular security assessments to verify claim validation policies remain effective against evolving threats, and ensuring executive sponsorship and cross-functional collaboration between security, development, and operations teams for overcoming organizational challenges inherent in adopting new security paradigms.

REFERENCES

[1] Bindu Mohan Harve, et al., "The Cloud-Native Revolution: Microservices in a Cloud-Driven World," in 2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), 11 March 2025. Available: https://ieeexplore.ieee.org/abstract/document/10913359

[2] Ritu Shree, et al., "Understanding the Limitations of Authentication Protocols Employed by Existing Information Security Models for Networked Applications," IEEE Transactions on Network and Service Management, 21 March 2024. Available: https://ieeexplore.ieee.org/abstract/document/10470720

[3] Anurag Bhardwaj and Nitin Jain, "Data In-Transit Security and Mitigation Mechanism in Cloud Computing," in 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 21 December 2021. Available: https://ieeexplore.ieee.org/abstract/document/9645907

[4] Atul Mishra, et al., "A factual study on hybrid multi cloud cyber security threats and proposed methodologies to enable cyber resilience," in 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 20 September 2024. Available: https://ieeexplore.ieee.org/document/10677052

[5] M. Jones and J. Hildebrand, "JSON Web Encryption (JWE)," RFC 7516, May 2015. Available: https://datatracker.ietf.org/doc/html/rfc7516

[6] Fatima Hussain, et al., "Intelligent Service Mesh Framework for API Security and Management," in 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 19 December 2019. Available: https://ieeexplore.ieee.org/document/8936216

[7] Asaduzzaman Jony, et al., "A Secure Token-Based Approach for DHCP Client Authentication and Replay Attack Prevention," in 2024 27th International Conference on Computer and Information Technology (ICCIT), 10 June 2025. Available: https://ieeexplore.ieee.org/document/11022024

[8] Jaikishan Jaikumar, et al., "Privacy-Preserving Personal Identifiable Information (PII) Label Detection Using Machine Learning," IEEE Access, October 2023. Available: https://ieeexplore.ieee.org/document/10307924 [9] Kiran Dangol, et al., "A Review on API Security Risk and Vulnerability Assessment," IEEE Access, 19 June 2025. Available: https://ieeexplore.ieee.org/document/11038691

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN:
2468-4376
https://www.jisem-journal.com/ **Research Article**

[10] Naeem Firdous Syed, et al., "Zero Trust Architecture (ZTA): A Comprehensive Survey," IEEE Access, 12 May 2022. Available: https://ieeexplore.ieee.org/document/9773102