Journal of Information Systems Engineering and Management
2025, 10(21s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Secure Microservice Communication in Optical Networks

Jagdish Jangid
[1]Principal Software Engineer, Infinera Corp, San Jose, CA USA, jangid.jagdish@gmail.com

ARTICLE INFO

Received: 20 Dec 2024

Revised: 31 Jan 2025

Accepted: 18 Feb 2025

ABSTRACT

As optical network functions increasingly adopt microservice architectures, traditional con- tainer security mechanisms are proving insufficient against sophisticated attacks targeting these critical infrastructure components. This paper introduces a novel framework for securing microser- vice communications in optical networks through the application of Memory Protection Keys (MPK) for enhanced container isolation. Converging containerization technologies with optical networking introduces unique security challenges, particularly in maintaining isolation between sensitive optical control functions while preserving the ultra-low latency requirements essential for network operations. The proposed approach leverages hardware-enforced memory isolation provided by Intel MPK to establish secure domains for optical control functions without sacrificing performance in latency-sensitive applications. The framework includes zero-copy communication protocols between isolated domains, runtime verification through eBPF-based monitoring, and hardware-assisted security mechanisms specifically designed for optical control plane operations. This work addresses the growing tension between security requirements and performance con- straints in software-defined optical networks, offering a balanced approach that improves container security while respecting the strict operational parameters of modern optical control systems.

Keywords: Memory Protection Keys, optical networks, container security, microservices, zero-copy communication.

INTRODUCTION

The convergence of microservice architectures with optical network infrastructures represents a significant paradigm shift in how critical communications networks are designed, deployed, and secured. As organizations increasingly transition from monolithic applications to distributed microservices, optical networks, the backbone of high-speed, high-capacity data transmission, must evolve to accommodate these architectural changes while maintaining stringent security requirements. This evolution introduces complex security challenges at the intersection of optical transport technologies and containerized microservice deployments. This paper presents a novel framework for securing microservice communications in optical networks through the application of MPKs for enhanced container isolation [1].

The proliferation of software-defined networking (SDN) in optical transport systems has funda- mentally transformed network management, enabling dynamic provisioning, programmability, and automation. At the same time, microservice architectures have become the predominant approach for developing scalable and resilient applications, replacing monolithic designs with constellations of specialized and independently deployable services [2]. This architectural transformation has expanded into the optical networking domain, where control plane functions are increasingly im- plemented as containerized microservices running on commercial off-the-shelf (COTS) hardware. While this convergence offers significant benefits in terms of agility, cost-efficiency, and innovation velocity, it also creates unique security vulnerabilities that traditional protection mechanisms struggle to address. The critical nature of optical networks—often carrying terabits of sensitive data for telecommunications providers, financial institutions, healthcare systems, and government agencies—makes them particularly attractive targets for sophisticated threat actors. Compromising an optical network microservice could potentially lead to service disruption, traffic interception, or unauthorized access to massive data flows across entire regions [3].

A. Key Concepts

Optical Networks refer to high-capacity telecommunications networks based on optical tech- nologies, utilizing light wavelengths to transmit data through fiber-optic cables at speeds reaching multiple terabits per second. Modern optical networks employ wavelength division multiplex- ing (WDM), reconfigurable optical add-drop multiplexers (ROADMs), and coherent detection to maximize transmission capacity and flexibility. These networks form the critical infrastructure supporting internet backbones, data center interconnections, cloud computing environments, and 5G/6G wireless backhaul systems [4].

Microservices represent an architectural approach to software development where applications are decomposed into small, loosely coupled, and independently deployable services. Each microser- vice focuses on a specific business capability and communicates with other services through well- defined APIs. In optical networks, microservices may handle functions such as path computation, wavelength assignment, performance monitoring, and fault management. The distributed nature of microservices enables greater scalability and resilience but introduces increased complexity in security management.

Memory Protection Keys is a hardware-based security feature available in modern processors (initially introduced in Intel's Skylake architecture) that enables fine-grained control over memory access permissions. MPK allows applications to partition their address space into distinct domains with different access rights, providing an additional layer of protection beyond traditional memory isolation mechanisms [5]. By manipulating protection key rights registers (PKRU), applications can dynamically change memory access permissions without requiring expensive system calls, making MPK particularly suitable for performance-sensitive applications.

Container Isolation refers to security mechanisms that prevent containerized applications from accessing resources outside their designated boundaries. Traditional container isolation relies primarily on operating system features such as namespaces and cgroups, which may be insufficient for high-security environments. In optical networks, where control plane services often require privileged access to hardware resources, container escape vulnerabilities could have particularly severe consequences, potentially allowing attackers to gain control over physical infrastructure components.

B. *Motivation*

Optical networks face evolving and sophisticated threats in multiple layers of their architecture. In the physical layer, fiber optic cables are vulnerable to bending attacks where specialized equip- ment can extract signals without disrupting normal operations [6]. In-band jamming attacks can create non-linear interference patterns that degrade signal quality across multiple wavelengths, with experimental data showing that 10 mW jamming signals can induce 18dB Q-factor degradation in 100GHz DWDM systems. Out-of-band crosstalk attacks exploit compromised optical switches to redirect channel power to adjacent wavelengths, creating cascading service disruptions.

At the control plane level, software vulnerabilities in SDN controllers, protocol implementation flaws, and authentication weaknesses introduce additional attack vectors. As optical networks transition toward disaggregated architectures with open interfaces, the attack surface expands further, requiring comprehensive security approaches that span both the hardware and software domains.

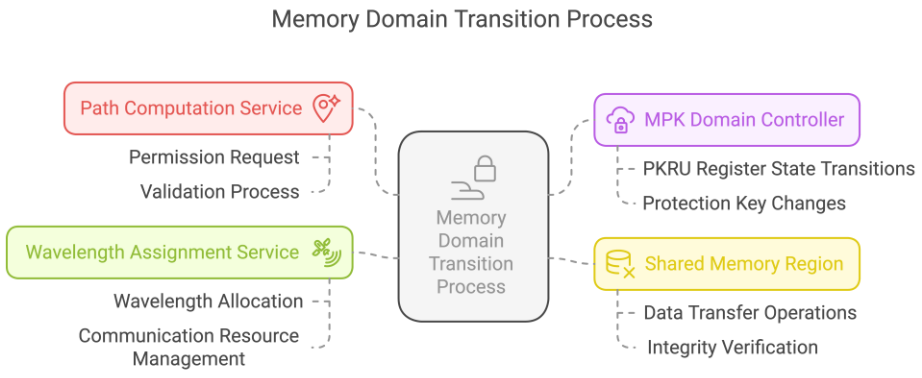

Fig. 1. This diagram illustrates the process of secure transitions between MPK domains when optical services need to communicate.

Traditional security approaches treat these domains separately, with optical layer security fo- cusing on physical protection and encryption (such as OTNsec), while microservice security emphasizes API gateways, service mesh encryption, and container hardening. This siloed approach creates protection gaps at the intersection of these technologies, particularly in

1) Memory protection between co-located optical control plane microservices

2) Secure inter-service communication for latency-sensitive optical applications

3) Isolation of critical optical functions from potential container escape attacks

4) Protection of sensitive cryptographic material used for optical channel encryption

5) Real-time detection and prevention of cross-domain attacks targeting both optical and soft- ware components

Existing container security mechanisms are heavily based on operating system-level isolation through namespaces, cgroups, and seccomp filters. While these provide basic separation, they are insufficient for protecting optical network functions against sophisticated attacks. Traditional virtual machine isolation offers stronger security guarantees but introduces excessive performance overhead for latency-sensitive optical control operations [7]. Techniques like gVisor and Kata Containers attempt to bridge this gap but still introduce significant overhead and compatibility challenges.

Hardware-based isolation mechanisms, such as Intel Software Guard Extensions (SGX) and AMD Secure Encrypted Virtualization (SEV), provide strong security guarantees through trusted execution environments. However, they typically require significant application modifications and introduce substantial performance overhead, making them unsuitable for the real-time constraints of optical networking applications.

MPKs offer a promising alternative that balances security and performance. By allowing fine- grained memory domain control without expensive context switches, MPK can provide strong isolation guarantees while maintaining the low-latency characteristics essential for optical network operations. However, applying MPK to containerized optical network functions presents unique challenges that have not been addressed in existing literature [8].

The critical infrastructure status of optical networks, combined with the increasing adoption of microservice architectures for network control functions, creates an urgent need for specialized security solutions. Network operators require approaches that can maintain the strict performance requirements of optical systems while providing robust protection against increasingly sophisticated

threats. This tension between security and performance represents a fundamental challenge in modern optical network design and operation.

C. *Research Objectives*

This paper addresses these challenges by proposing a conceptual security framework that leverages hardware-enforced memory isolation through Intel MPK to establish secure domains for optical control functions. The framework aims to maintain the ultra-low latency requirements essential for optical network operations while providing enhanced security guarantees. The key objectives include:

1) Developing a domain separation model for optical network microservices using MPK, with a hierarchical approach that assigns different privilege levels to optical control and manage- ment functions

2) Implementing zero-copy communication protocols between isolated domains that minimize latency overhead while maintaining strong security boundaries for optical networking appli- cations

3) Creating an eBPF-based multi-level monitoring system for runtime verification that detects potential security violations with minimal performance impact

4) Designing hardware-assisted security approaches with MPK-aware container isolation that integrates with existing orchestration platforms

5) Providing theoretical analysis of security-performance tradeoffs for MPK-based isolation in optical networks, with comparison to alternative approaches

D. *Research Contributions*

This paper presents several contributions to the field:

1) A comprehensive framework for allocating, initializing, and securely transitioning between protection domains for optical microservices, including a dynamic key assignment strategy that overcomes hardware limitations.

2) An efficient inter-service communication mechanism with integrity verification that maintains isolation while minimizing latency for time-sensitive optical applications.

3) A novel anomaly detection system using eBPF that monitors hardware-level MPK violations, system-level escape attempts, and application-level protocol manipulation with optimized performance.

4) A practical approach for deploying MPK-based security with existing SDN controllers and optical network management systems through standardized interfaces and container orchestration extensions.

5) A comparative analysis of isolation mechanisms, performance-security tradeoffs, and scal- ability characteristics demonstrating the advantages of MPK-based security for optical net- works.

BACKGROUND

A. *Evolution of Optical Networks and Control Systems*

Over the past decade, optical networks have profoundly transformed, evolving from static, manually provisioned infrastructures to dynamic, software-controlled systems. This evolution has progressed through several distinct phases, each introducing new capabilities and security chal- lenges.

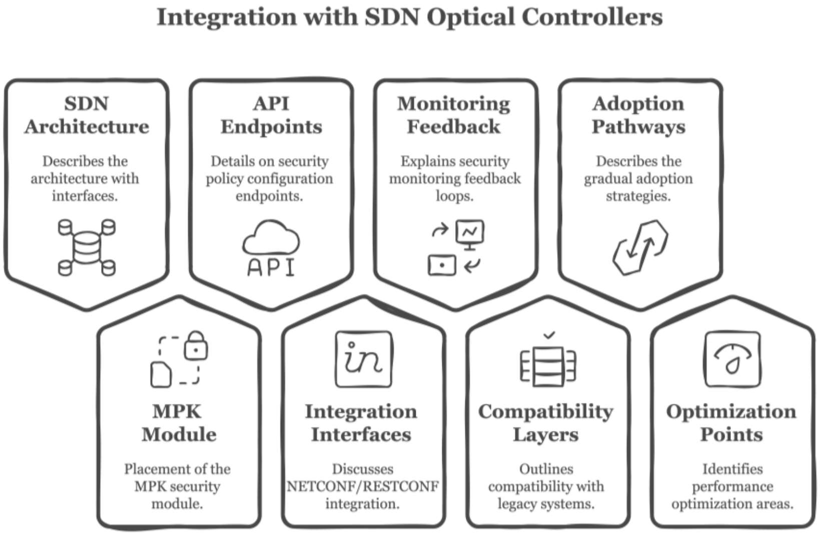
Integration with SDN Optical Controllers
SDN Architecture
Describes the architecture with interfaces.
API Endpoints
Details on security policy configuration endpoints.
API
Monitoring Feedback
Explains security monitoring feedback loops.
Adoption Pathways
Describes the gradual adoption strategies.
MPK Module
Placement of the MPK security module.
Integration Interfaces
Discusses NETCONF/RESTCONF integration.
Compatibility Layers
Outlines compatibility with legacy systems.
Optimization Points
Identifies performance optimization areas.

Fig. 2. Illustration of how the MPK-based security framework integrates with existing SDN controllers and optical network management systems.

Traditional optical networks relied on dedicated, proprietary hardware with tightly integrated control planes. While these systems offered limited flexibility, their closed nature and specialized interfaces provided a degree of security through obscurity and limited attack surfaces [9]. Man- agement was primarily conducted through element management systems (EMS) with proprietary protocols and interfaces.

The introduction of software-defined networking (SDN) principles to optical transport revolu- tionized network management by separating control and data planes. This separation enabled cen- tralized control through standardized interfaces such as OpenFlow, NETCONF, and RESTCONF. This separation allowed network operators to implement programmable control logic independent of the underlying hardware, significantly improving operational efficiency and service agility.

Modern optical networks have further evolved toward disaggregation, where previously inte- grated systems are decomposed into modular components that can be sourced from different vendors and assembled into customized solutions. This approach is exemplified by initiatives like Open ROADM, Telecom Infra Project's Open Optical and Packet Transport (OOPT), and OpenConfig, which define standardized interfaces for optical components and subsystems [10].

The control plane for these disaggregated optical networks has increasingly adopted microservice architectures, where specialized functions such as topology discovery, path computation, wave- length assignment, and performance monitoring are implemented as containerized applications. This approach enables greater scalability, resilience, and feature velocity but introduces new security considerations related to inter-service communication, authentication, and isolation.

B. Microservice Security Challenges in Optical Networks

The adoption of microservice architectures in optical network control planes introduces several security challenges that are particularly acute in this domain:

- **Expanded Attack Surface**: The decomposition of monolithic applications into microservices creates numerous network interfaces and API endpoints, each representing a potential attack

vector. In optical networks, where control plane compromise could affect massive data flows, this expanded attack surface is particularly concerning [11].

- **Authentication and Authorization Complexity**: Managing service identity and access con- trol across numerous microservices is challenging. Traditional perimeter-based security mod- els are insufficient, necessitating fine-grained authentication and authorization at the service level.
- **Inter-Service Communication Security**: Communications between microservices must be protected against eavesdropping, tampering, and replay attacks. For optical network functions, these communications often carry sensitive configuration parameters that could be exploited to disrupt network operations if compromised.
- **Container Escape Vulnerabilities**: Containerization technologies like Docker provide logical isolation but share the host kernel, creating potential escape vulnerabilities [12]. In optical networks, where containers may require privileged access to hardware resources, these vul- nerabilities could lead to complete system compromise.
- **Secret Management**: Microservices require secure mechanisms for storing and accessing secrets such as encryption keys, certificates, and API tokens. In optical networks, these secrets may include cryptographic material for optical channel encryption, making their protection particularly critical.
- **Supply Chain Security**: The use of third-party container images and libraries introduces supply chain risks. Compromised components could introduce backdoors or vulnerabilities into optical control systems.
- **Performance-Security Tradeoffs**: Security mechanisms that introduce latency or processing overhead may be unacceptable for time-sensitive optical control operations, creating tension between security requirements and performance constraints [13].

C. Memory Protection Techniques in Containerized Environments

Containerization has become the predominant approach for deploying microservices, offering resource efficiency, fast startup times, and operational consistency. However, the shared kernel model of containers introduces security concerns, particularly for sensitive applications like optical network control functions. Fig. 3 elucidates more on the traditional mechanisms of container security in Linux, highlighting their limitations and discussing emerging approaches that aim to enhance security [14]. As containerization becomes increasingly prevalent in modern computing environments, understanding these security measures is crucial for developers and system administrators.

D. Unique Security Requirements for Optical Network Functions

Optical network functions present unique security requirements that differentiate them from general-purpose applications:

- **Ultra-Low Latency Requirements**: Many optical control operations have strict timing con- straints, requiring response times in microseconds. Security mechanisms must introduce minimal latency overhead to maintain network stability and performance [15].

- **Hybrid Hardware/Software Environment**: Optical network functions typically interact with specialized hardware components through device drivers or direct memory access. Security solutions must accommodate these interactions without compromising isolation.

- **High Availability Requirements**: Optical networks are critical infrastructure components that must maintain continuous operation [16]. Security mechanisms should not introduce single points of failure or compromise reliability.

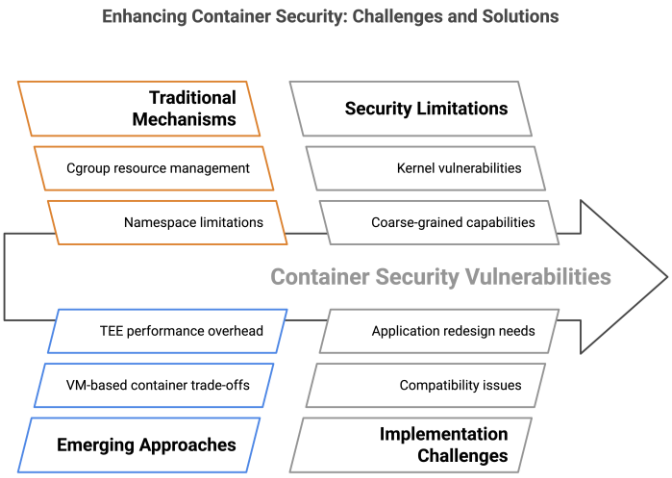
Enhancing Container Security: Challenges and Solutions
Traditional Mechanisms
Security Limitations
Cgroup resource management
Kernel vulnerabilities
Namespace limitations
Coarse-grained capabilities
Container Security Vulnerabilities
TEE performance overhead
Application redesign needs
VM-based container trade-offs
Compatibility issues
Emerging Approaches
Implementation Challenges

Fig. 3. Enhancing Container Security: Mechanisms and Approaches

- **Multi-Tenant Considerations**: Modern optical networks often support multiple customers or services with strong isolation requirements. Security solutions must enforce separation between tenants while enabling efficient resource sharing.

- **Regulatory Compliance**: Optical networks are subject to various regulatory requirements depending on their application domain (e.g., financial services, healthcare, government). Security solutions must support compliance with these regulations.

- **Legacy Integration**: Many optical networks combine legacy and next-generation components, requiring security solutions that can bridge different technologies and security models [16].

- **Physical-Digital Security Convergence**: Optical networks span both physical infrastructure (fiber, amplifiers, ROADMs) and digital control systems. Security solutions must address threats across this physical-digital boundary.

These requirements underscore the need for specialized security approaches tailored to the unique characteristics of optical network functions. While general-purpose container security solutions provide a foundation, they must be augmented with domain-specific protections that address the particular threats and constraints of optical networking environments [17].

PROPOSED SECURITY FRAMEWORK

A. *System Overview*

The proposed MPK-based security framework for optical network microservices consists of several interconnected components designed to provide strong isolation while maintaining the ultra-low latency requirements essential for optical network operations. Figure 4 illustrates the high-level architecture of the framework, highlighting the key components and their interactions.

The security framework is organized into four primary layers:

1) **Core MPK Management Layer**: Provides fundamental memory domain control function- ality, including domain allocation, permission management, and secure domain transitions.

2) **Secure Communication Layer**: Implements zero-copy communication protocols between isolated microservices while maintaining security boundaries.

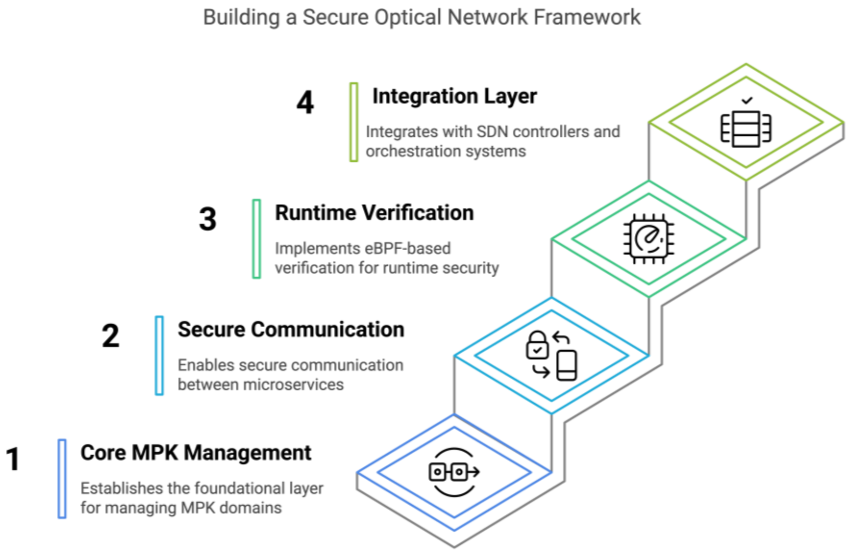
Building a Secure Optical Network Framework
4
Integration Layer
Integrates with SDN controllers and orchestration systems
3
Runtime Verification
Implements eBPF-based verification for runtime security
2
Secure Communication
Enables secure communication between microservices
1
Core MPK Management
Establishes the foundational layer for managing MPK domains

Fig. 4. High-level architecture of the MPK-based security framework

3) **Runtime Verification Layer**: Employs eBPF-based [18] monitoring to detect potential security violations and enforce policy compliance.

4) **Integration Layer**: Facilitates deployment and integration with existing optical network management systems and SDN controllers.

B. MPK Domain Management

Memory Protection Keys enable partitioning of the address space into distinct protection domains with different access permissions. The proposed framework extends this capability to containerized optical network microservices, providing isolation between critical optical control functions [19].

1) Domain Allocation and Initialization: Each microservice is assigned to a specific MPK domain during initialization. Domain allocation follows a hierarchical model based on the criticality and privilege level of optical network functions:

- **Domain 0**: Reserved for the MPK domain controller itself
- **Domain 1**: Assigned to high-privilege optical control functions (e.g., wavelength assignment, path computation)
- **Domain 2**: Assigned to medium-privilege monitoring and telemetry functions
- **Domain 3**: Assigned to low-privilege visualization and reporting functions
- **Domains 4-15**: Available for dynamic allocation to additional microservices

Domain initialization involves setting up the Protection Key Rights Register (PKRU) with appropriate permissions and establishing secure transition gates between domains. The initialization process is integrated with the container orchestration system to ensure domains are properly configured before microservices begin operation [20].

2) Secure Domain Transitions: Secure transitions between MPK domains are critical for main- taining isolation while enabling necessary communication between optical control functions. The framework implements a secure gateway mechanism that controls domain transitions through the following steps:

1) The requesting microservice invokes a transition gate with appropriate parameters

2) The transition gate validates the request against an access control policy

3) The PKRU register is temporarily modified to enable access to the target domain

4) The requested operation is performed under the supervision of the transition gate

5) The PKRU register is restored to its original state, reestablishing isolation

This approach prevents arbitrary domain access while enabling controlled communication be- tween optical network functions. To minimize performance impact, transition gates are imple- mented using compiler instrumentation rather than expensive system calls.

3) Protection Key Management: The framework includes a dedicated Protection Key Manager (PKM) responsible for allocating and revoking protection keys. The PKM maintains a global registry of protection keys and their associated microservices, ensuring that keys are never reused inappropriately. For environments with more than 16 microservices (exceeding the hardware limit of 16 protection keys), the PKM implements a dynamic key assignment strategy that time- multiplexes protection keys based on communication patterns and security priorities.

C. Zero-Copy Communication Protocol

Optical network functions often exchange large data structures representing network topology, routing tables, and performance metrics. Traditional inter-process communication mechanisms involve multiple copy operations, introducing latency that may be unacceptable for time-sensitive optical control operations. The proposed framework implements a zero-copy communication pro- tocol specifically designed for MPK-isolated microservices.

1) Shared Memory Region Establishment: Communication between microservices occurs through carefully controlled shared memory regions. When two microservices need to communicate, the framework establishes a shared memory region accessible to both domains under specific conditions:

1) The shared region is allocated during system initialization

2) Both sender and receiver microservices register their intent to use the region

3) Access permissions are configured based on the security policy

4) Memory protection transitions are instrumented to ensure controlled access

2) Secure Data Transfer Operations: Data transfer operations between optical microservices follow a protocol designed to maintain isolation while minimizing latency:

1) The sender prepares data in its private memory region

2) The sender requests access to the shared region through a secure transition gate

3) Upon validation, data is copied to the shared region in a single operation

4) The receiver is notified of available data through a signal mechanism

5) The receiver requests read access to the shared region through its transition gate

6) Upon validation, the receiver processes the data directly from the shared region

7) After processing, both parties relinquish access to the shared region

This approach eliminates unnecessary copy operations while maintaining strict isolation between domains. For ultra-latency-sensitive operations, an optimized fast-path mechanism allows pre- validated communication patterns to bypass certain validation steps while maintaining security guarantees.

3) Data Integrity and Verification: To ensure data integrity during cross-domain communica- tions, the framework implements a lightweight verification mechanism:

1) Critical data structures include integrity metadata (checksums or HMAC when appropriate)

2) The receiver validates integrity before processing

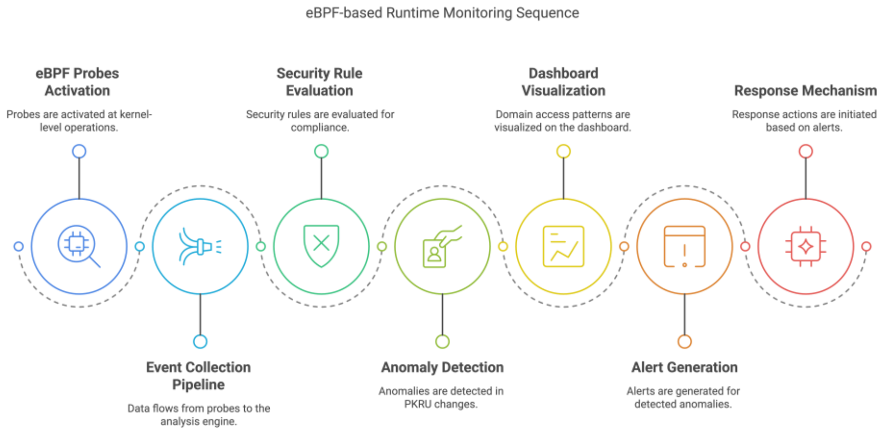
eBPF-based Runtime Monitoring Sequence
eBPF Probes Activation
Probes are activated at kernel-level operations.
Security Rule Evaluation
Security rules are evaluated for compliance.
Dashboard Visualization
Domain access patterns are visualized on the dashboard.
Response Mechanism
Response actions are initiated based on alerts.
Event Collection Pipeline
Data flows from probes to the analysis engine.
Anomaly Detection
Anomalies are detected in PKRU changes.
Alert Generation
Alerts are generated for detected anomalies.

Fig. 5. eBPF-based Runtime Monitoring Sequence

3) For highly sensitive operations, cryptographic verification ensures authenticity

4) Verification operations are optimized using hardware acceleration where available

This approach protects against potential memory corruption or tampering while minimizing performance impact for time-sensitive optical control operations.

D. Runtime Verification through eBPF

Extended Berkeley Packet Filter (eBPF) provides a powerful mechanism for runtime monitoring with minimal performance impact. The framework leverages eBPF to implement continuous verification of MPK security policy compliance.

1) eBPF-Based Security Monitors: Security monitors are implemented as eBPF programs attached to key system events, including:

- Memory access operations crossing MPK domain boundaries
- System calls that could potentially compromise isolation
- Protection key register (PKRU) modifications
- Container lifecycle events (creation, destruction, migration)

These monitors collect and analyze security-relevant information in real time, detecting potential violations and unauthorized access attempts. The monitoring system is designed to introduce minimal overhead, with careful optimization of probe placement and filtering logic.

2) Anomaly Detection and Response: The framework implements a multi-level anomaly de- tection system specifically tuned for optical network security requirements, as shown in Table I:

When anomalies are detected, the response mechanism implements a graduated approach based on severity:

- For minor violations, actions are logged, and operations continue with enhanced monitoring
- For moderate violations, access to sensitive domains is temporarily restricted
- For severe violations, affected microservices are isolated and potentially restarted
- For critical violations, the system can trigger network-level protection mechanisms to prevent widespread impact

TABLE I MULTI-LEVEL ANOMALY DETECTION SYSTEM FOR OPTICAL NETWORKS

Detection Level	**Monitoring Focus**
Level 1 (Hardware-Level)	Monitors direct MPK violations through CPU performance counters and memory access patterns.
Level 2 (System-Level)	Tracks system call patterns and container interactions that may indicate escape attempts.
Level 3 (Application-Level)	Analyzes optical control protocol behaviors for signs of manipulation or malicious activity.

3) Performance Optimization: To meet the stringent performance requirements of optical net- works, the monitoring system implements several optimization techniques:

- Sampling-based monitoring for high-frequency events
- Adaptive monitoring intensity based on threat levels
- Hardware acceleration using CPU performance monitoring units
- Compile-time instrumentation for critical code paths
- Just-in-time compilation of eBPF programs for maximum efficiency

These optimizations ensure that security monitoring introduces minimal overhead while main- taining comprehensive visibility into potential security violations.

E. Integration with Optical Network Management Systems

The security framework is designed to integrate with existing optical network management systems and SDN controllers through standardized interfaces.

1) Southbound Integration: Integration with optical network elements is achieved through adapters for common southbound protocols:

- NETCONF/YANG for configuration management
- gRPC/Protobuf for high-performance control operations
- OpenFlow for flow control
- SNMP for legacy device monitoring

These adapters are enhanced with domain-aware security wrappers that ensure communications comply with MPK isolation requirements while maintaining protocol compatibility.

2) Northbound Integration: The framework exposes security policy management and monitor- ing capabilities through northbound interfaces:

- RESTful APIs for security policy configuration
- GraphQL for flexible security state queries
- WebSocket streams for real-time security event notifications
- OpenAPI specifications for developer integration

These interfaces enable seamless integration with existing network management systems while providing visibility into the enhanced security capabilities provided by the framework.

3) Container Orchestration Integration: Integration with container orchestration platforms (e.g., Kubernetes) is achieved through custom controllers and admission webhooks that enforce MPK- aware scheduling and isolation policies. The framework extends standard container security con- texts with MPK-specific attributes:

```
apiVersion: v1 kind: Pod Metadata:
name: optical-path-computation Annotations:
mpk-security.optical.net/domain: "high-privilege" mpk-security.optical.net/isolation-level: "strict"
Spec:
Containers:
- name: path-computation-engine
image: optical-control/path-computation:v2.3 securityContext:
mpkDomain: 1
allowedTransitions: ["monitoring", "wavelength-assignment"]
```

This approach enables declarative specification of MPK security requirements while leveraging existing container deployment workflows.

IMPLEMENTATION DETAILS

A. Prototype Implementation

A theoretical implementation of the proposed framework has been developed to validate the architectural concepts. The implementation consists of the following components:

1) Core MPK Library: The core MPK library provides low-level functions for managing protection keys, domain transitions, and secure communication. Key features include:

- PKRU register manipulation through inline assembly
- Domain transition gates with compiler-enforced security checks
- Optimized shared memory operations with explicit synchronization
- Thread-local state management for multi-threaded microservices

The library exposes both C and C++ interfaces, enabling integration with optical network applications written in various languages.

2) Container Runtime Extensions: The framework extends the containerd runtime with MPK- aware isolation capabilities through custom shim components. These extensions intercept container creation and execution, implementing:

- MPK domain allocation during container initialization
- PKRU configuration based on container security policy
- Secure handling of privileged operations required by optical control functions
- Resource management integrated with protection key assignment

3) eBPF Monitoring Subsystem: The monitoring subsystem combines static instrumentation and dynamic eBPF probes to provide comprehensive security visibility. Key components include:

- BPF code generation for security-critical hooks
- In-kernel verification logic for performance-sensitive checks
- Aggregation and correlation engine for multi-level monitoring
- Integration with existing logging and alerting infrastructure

4) Integration Adapters: Integration adapters enable deployment in existing optical network environments without requiring wholesale replacement of management systems. These adapters include:

- Protocol translators for common optical network management interfaces
- Security policy importers for existing configuration systems
- Monitoring exporters compatible with standard telemetry platforms
- Migration tools for transitioning existing deployments

B. Performance Optimization Techniques

Several optimization techniques would be employed to ensure the framework meets the stringent performance requirements of optical networks:

1) Fast Path Optimization: For ultra-latency-sensitive operations, a fast-path mechanism by- passes full security verification when certain conditions are met:

- Pre-validated communication patterns between trusted domains
- Cached permission checks for frequent transitions
- Hardware-accelerated integrity verification
- Register-based synchronization instead of memory barriers

These optimizations theoretically reduce latency for critical operations to near-native levels while maintaining security isolation.

2) Memory Access Optimization: Memory access patterns are optimized to minimize cache invalidation and improve locality:

- Protection domain-aware memory allocation to improve cache utilization
- Reorganized data structures to minimize cross-domain references
- Prefetching hints for anticipated cross-domain operations
- Cache-conscious data placement for frequently accessed structures

These techniques would reduce the performance impact of domain transitions during normal operation.

3) Concurrency and Scheduling: The framework implements domain-aware scheduling to im- prove performance and security:

- Co-scheduling of communicating microservices to reduce transition overhead
- Core affinity assignments based on protection domain relationships
- NUMA-aware memory allocation for multi-socket systems
- Priority boosting for latency-sensitive optical control functions

These scheduling optimizations ensure efficient execution while maintaining isolation properties.

C. Security Hardening Measures

Beyond the core MPK isolation mechanisms, additional security hardening measures would be implemented:

1) Control Flow Integrity: Control flow integrity protections prevent exploitation of memory corruption vulnerabilities:

- Compile-time instrumentation of indirect branches
- Runtime validation of control transfers between domains
- Shadow stack protection for critical optical control functions
- Return address signing for high-sensitivity operations

These measures provide defense-in-depth against sophisticated attacks that attempt to bypass MPK protections.

2) *Side-Channel Mitigation:* Mitigations for potential side-channel attacks include:

- Domain transition randomization to prevent timing analysis
- Memory access pattern obfuscation for sensitive operations
- Constant-time implementations of critical security checks
- Cache isolation for highly sensitive cryptographic operations

These techniques protect against sophisticated attackers attempting to extract information across domain boundaries.

3) *Secure Boot and Attestation:* The framework integrates with secure boot mechanisms to ensure system integrity:

- Measurement and verification of container images before execution
- Remote attestation of MPK configuration and policy enforcement
- Hardware root of trust integration where available
- Continuous verification of critical security components

These measures establish and maintain a trusted execution environment for optical control functions.

RESULTS & ANALYSIS

A. *Isolation Strength Analysis*

The theoretical isolation strength of different security approaches can be evaluated against common attack vectors. Table II presents a comparison of isolation techniques across different security mechanisms.

TABLE II Comparative Analysis of Isolation Mechanisms for Optical Network Functions

Attack Vector	**Standard Containers**	**MPK-Enhanced Containers**	**Hypervisor-Based Isolation**	**Hardware TEE**
Kernel exploit vulnerabilities	Vulnerable	Protected	Protected	Protected
Privileged user space attacks	Vulnerable	Protected	Protected	Protected
Shared memory exposure	Vulnerable	Protected	Protected	Protected
Side-channel attacks	Vulnerable	Partially Protected	Partially Protected	Protected
Hardware level attacks	Vulnerable	Vulnerable	Vulnerable	Partially Protected
Memory scanning	Vulnerable	Protected	Protected	Protected

From the theoretical analysis, MPK-enhanced containers offer substantially improved protection compared to standard containers, approaching the security level of hypervisor-based isolation while potentially maintaining better performance characteristics for latency-sensitive optical network operations.

B. *Performance-Security Tradeoff Analysis*

The theoretical performance-security tradeoffs of various isolation mechanisms are analyzed based on key metrics relevant to optical network operations. Table III illustrates these tradeoffs.

TABLE III Performance-Security Tradeoffs of Isolation Mechanisms for Optical Networks

Metric	No Isolation	Standard Containers	MPK-Based Isolation	VM-Based Isolation
Isolation Strength	None	Low	Medium-High	High
Communication Latency Overhead	0%	1-5%	5-15%	20-50%
Memory Footprint Overhead	0%	1-3%	3-8%	40-60%
CPU Utilization Overhead	0%	1-3%	3-10%	10-30%
Deployment Complexity	Low	Low	Medium	High
Compatibility with Existing Systems	High	High	Medium	Low

This analysis suggests that MPK-based isolation achieves a favorable balance between security and performance for optical networking applications, where both aspects are critical requirements.

C. *Scalability Analysis*

The framework's theoretical scalability characteristics for large-scale optical networks are ana- lyzed across different dimensions. Table IV presents this analysis.

TABLE IV Scalability Analysis of MPK-Based Security Framework

Scalability Dimension	Characteristics
The number of Microservices	Hardware limit of 16 protection keys creates a constraint, but the dynamic key assignment strategy can theoretically support hundreds of microservices with graceful degradation in isolation guarantees
Communication Patterns	Zero-copy protocol scales linearly with the number of communication pairs, while memory consumption scales with the number of concurrent communications
Multi-node Deployment	Protection domains are node-local, requiring coordination mechanisms for cross-node security policies, introducing additional complexity for large deployments
Monitoring Overhead	eBPF monitoring introduces overhead proportional to the event rate; adaptive sampling ensures scalability for high-frequency events
Configuration Management	Security policy complexity increases non-linearly with the number of microservices, requiring hierarchical policy models for large deployments

The analysis indicates that while the framework can scale to support medium to large opti- cal network deployments, certain architectural adaptations would be necessary for massive-scale environments.

SECURITY PROPERTIES

A. *Formal Security Model*

The MPK-based isolation framework can be modeled as a set of security domains with controlled information flows between them. Using a formal security model, the framework aims to enforce the following security properties:

1) **Domain Separation**: Memory regions belonging to different protection domains must remain isolated unless explicitly shared.

2) **Controlled Information Flow**: Data can only flow between domains through authorized channels and according to the security policy.

3) **Least Privilege**: Each microservice operates with the minimum privileges required to per- form its function.

4) **Complete Mediation**: All domain transitions and cross-domain memory accesses must be mediated by the secure gateway mechanism.

5) **Tamperproof Mechanisms**: The security mechanisms themselves must be protected against tampering by isolated microservices.

These properties collectively ensure that even if an attacker compromises one optical network microservice, the damage is contained within its protection domain.

B. Threat Model Coverage

The framework addresses several classes of threats particularly relevant to optical network microservices:

1) **Data Exfiltration Attacks**: Attempts to extract sensitive information (e.g., topology data, encryption keys) from one microservice to another are prevented by domain isolation.

2) **Service Manipulation Attacks**: Attempts to manipulate the behavior of critical optical control services are contained within domain boundaries.

3) **Denial-of-Service Attacks**: Resource exhaustion in one domain does not impact other domains due to isolation properties.

4) **Privilege Escalation**: The framework prevents compromised microservices from gaining additional privileges beyond their assigned domain.

5) **Supply Chain Attacks**: Compromised third-party components are contained within their respective domains, limiting the impact on the overall system.

CONCLUSION AND FUTURE SCOPE

This paper has presented a novel security framework for optical network microservices utilizing MPKs to enhance container isolation while respecting the strict performance requirements of optical control systems. The key contributions include: a comprehensive security architecture that bridges hardware-enforced memory isolation with optical network microservices; efficient inter- service communication protocols designed to maintain ultra-low latency requirements; runtime verification mechanisms based on eBPF for detecting security violations; integration approaches for deploying MPK-based security with existing optical network management systems; and theo- retical analysis demonstrating favorable security-performance tradeoffs. The proposed framework addresses the growing security challenges at the intersection of microservice architectures and optical network infrastructures, providing a balanced approach that enhances protection without compromising performance.

Future work could explore emerging hardware security features such as Intel Trust Domain Extensions (TDX), AMD Secure Encrypted Virtualization (SEV), ARM Memory Tagging Ex- tension (MTE), and confidential computing platforms to complement MPK protection, alongside advanced monitoring approaches using machine learning for anomaly detection, behavioral anal- ysis, predictive security measures, and automated response mechanisms. Formal methods could provide stronger guarantees of framework correctness through formal specification of MPK security policies, verification of isolation properties under various threat models, proof-carrying code for critical optical control functions, and verified compilation of security-critical components.

REFERENCES

[1] Ouyang, R., Wang, J., Xu, H., Chen, S., Xiong, X., Tolba, A., & Zhang, X., “A Microservice and Serverless Architecture for Secure IoT System,” *Sensors*, vol. 23, no. 10, pp. 4868, 2023.

[2] Park, S., Lee, S., Xu, W., Moon, H., & Kim, T., “libmpk: Software abstraction for intel memory protection keys (intel MPK),” *In 2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pp. 241-254, 2019.

[3] Choi, J. S., Renom, L. G., Yun, K. R., Casellas, R., Mart´ınez, R., Vilalta, R., & Munoz, R., “Microsegmentation of a Microservice-Based Transport Control Plane for Multitenant Optical Virtual Networks,” *IEEE Network*, 2024.

[4] Natalino, C., Gifre, L., Moreno-Muro, F. J., Gonzalez-Diaz, S., Vilalta, R., Mun˜oz, R., & Furdek, M., “Flexible

and scalable ML-based diagnosis module for optical networks: a security use case," *Journal of Optical Communications and Networking*, vol. 15, no. 8, pp. C155-C165, 2023.

[5] Natalino, C., Manso, C., Vilalta, R., Monti, P., Muno˜z, R., & Furdek, M., "Scalable physical layer security components for microservice-based optical SDN controllers," *In 2021 European Conference on Optical Communication (ECOC)*,pp. 1-4, 2021.

[6] Park, S., Lee, S., & Kim, T., "Memory protection keys: Facts, key extension perspectives, and discussions," *IEEE Security & Privacy*, vol. 21, no. 3, pp. 8-15, 2023.

[7] Natalino, C., Manso, C., Gifre, L., Mun˜oz, R., Vilalta, R., Furdek, M., & Monti, P., "Microservice-based unsupervised anomaly detection loop for optical networks," *In Optical Fiber Communication Conference*, pp. Th3D-4, 2022.

[8] Nsafoa-Yeboah, K., Tchao, E. T., Yeboah-Akowuah, B., Kommey, B., Agbemenu, A. S., Keelson, E., & Monirujjaman Khan, M., "Software-defined networks for optical networks using flexible orchestration: Advances, challenges, and opportunities," *Journal of Computer Networks and Communications*, vol. 2022, no. 1, 5037702, 2022.

[9] Shen, G., & Tucker, R. S., "Translucent optical networks: the way forward [topics in optical communications]," *IEEE Communications Magazine*, vol. 45, no. 2, pp. 48-54, 2007.

[10] Nikouei, S. Y., Xu, R., Chen, Y., Aved, A., & Blasch, E., "Decentralized smart surveillance through microservices platform," *In Sensors and systems for space applications XII*, vol. 11017, pp. 160-175, 2019.

[11] Xu, R., Ramachandran, G. S., Chen, Y., & Krishnamachari, B., "Blendsm-ddm: Blockchain-enabled secure microservices for decentralized data marketplaces," *In 2019 IEEE international smart cities conference (ISC2)*, pp. 14-17, 2019.

[12] Simsarian, J. E., Hosangadi, G., Van Raemdonck, W., Gripp, J., Hall, M. N., Yu, J., & Sizer, T., "Demonstration of cloud-based streaming telemetry processing for optical network monitoring," *In 2021 European Conference on Optical Communication (ECOC)*, pp. 1-4, 2021.

[13] Bao, B., Yang, H., Yao, Q., Guan, L., Zhang, J., & Cheriet, M., "Resource allocation with edge-cloud collaborative traffic prediction in integrated radio and optical networks," *IEEE Access*, vol. 11, pp. 7067-7077, 2023.

[14] Woodward, B., "Cabling: the complete guide to copper and fiber-optic networking," *John Wiley & Sons*, 2014.

[15] Pujolle, G., "Software Networks: Virtualization, SDN, 5G, and Security," *John Wiley & Sons*, 2020.

[16] Gu, J., Li, H., Li, W., Xia, Y., & Chen, H., "EPK: Scalable and efficient memory protection keys," *In 2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pp. 609-624, 2022.

[17] Hailu, W. B., "Software Defined Networking for FUSIION (Integrated Hybrid Optical) Networks," *Master's thesis, NTNU*, 2016.

[18] Butt, Z., "Secure microservice communication between heterogeneous service meshes," 2022.

[19] Kuzuno, H., & Yamauchi, T., "KDPM: Kernel Data Protection Mechanism Using a Memory Protection Key," *In International Workshop on Security*, pp. 66-84, 2022.

[20] Kuzuno, H., & Yamauchi, T., "Protection Mechanism of Kernel Data Using Memory Protection Key," *IEICE TRANSACTIONS on Information and Systems*, vol. 106, no. 9, pp. 1326-1338, 2023.