PROVISIONAL PATENT APPLICATION

TITLE: Dynamic Multi-Protocol Security Orchestration System with Real-Time Quantum Threat Assessment and Automated Algorithm Selection

DOCKET NUMBER: MWRASP-001-PROV

INVENTOR(S): MWRASP Development Team

FILED: September 3, 2025

FIELD OF THE INVENTION

This invention relates to dynamic security orchestration systems for quantum-resistant cryptographic infrastructure, specifically to multi-protocol systems that automatically adapt security protocols based on real-time quantum threat assessment, device capabilities, and operational conditions while maintaining seamless communication across diverse computing environments.

BACKGROUND OF THE INVENTION

The emergence of quantum computing poses an unprecedented threat to current cryptographic infrastructure. Organizations must transition from quantum-vulnerable algorithms (RSA, ECC, DH) to post-quantum alternatives while maintaining operational continuity. Current limitations include:

Static Security Protocol Limitations: - **Fixed protocol selection**: Cannot adapt to varying quantum threat levels or device capabilities - **Manual migration processes**: Requiring extensive human intervention for protocol transitions - **Compatibility issues**: Inability to maintain communication between classical and post-quantum systems - **Threat-insensitive operations**: No consideration of real-time quantum computing capabilities or threat intelligence - **Resource-agnostic deployment**: Same protocols regardless of device computational constraints

Need for Dynamic Quantum-Aware Security Orchestration: Modern computing environments span from IoT devices to high-performance servers, each with different computational capabilities and security requirements. As quantum computers advance, organizations need security systems that can automatically assess quantum threats and dynamically select appropriate cryptographic protocols while maintaining interoperability across diverse systems.

SUMMARY OF THE INVENTION

The present invention provides a Dynamic Multi-Protocol Security Orchestration System that automatically selects and deploys optimal cryptographic protocols based on real-time quantum threat assessment, device capabilities, and operational requirements while maintaining seamless interoperability across diverse computing environments.

Key innovations include:

1. **Real-Time Quantum Threat Assessment Engine**: Continuous monitoring of quantum computing capabilities and threat landscape 2. **Dynamic Protocol Selection Algorithm**: Intelligent selection of optimal cryptographic protocols based on multiple factors 3. **Seamless Protocol Bridging**: Transparent communication between systems using different cryptographic protocols 4. **Automated Vulnerability Assessment**: Real-time evaluation of cryptographic vulnerabilities and security gaps 5. **Adaptive Resource Management**: Optimization of cryptographic operations based on device capabilities 6. **Intelligent Migration Orchestration**: Coordinated transition of systems from vulnerable to quantum-safe protocols

The system provides robust, adaptive security that evolves with the quantum threat landscape while maintaining operational efficiency.

DETAILED DESCRIPTION OF THE INVENTION

System Architecture Overview

The Dynamic Multi-Protocol Security Orchestration System represents a revolutionary approach to quantum-aware cryptographic security through intelligent protocol selection and automated threat assessment. The system comprises multiple integrated components working in concert to provide adaptive, efficient security across diverse computing environments.

Core Architectural Principles

Quantum-Aware Security Framework: - Real-time quantum threat monitoring and assessment - Dynamic protocol selection based on threat intelligence - Automated security parameter adaptation - Seamless interoperability across protocol boundaries

Multi-Protocol Orchestration Engine: - Intelligent protocol negotiation and selection - Resource-aware cryptographic deployment - Performance optimization across diverse devices - Automated security level adjustment

Adaptive Learning System: - Continuous threat landscape monitoring - Historical performance analysis and optimization - Predictive protocol selection algorithms - Cross-system knowledge sharing

System Components Architecture

The system architecture is built on a modular, scalable foundation that enables independent operation of components while maintaining seamless integration:

```
python class DynamicMultiProtocolSecurityOrchestrator: """
Master orchestrator for dynamic multi-protocol security systems
Coordinates threat assessment, protocol selection, and
deployment """ def __init__(self): # Initialize core security
engines self.quantum_threat_engine =
QuantumThreatAssessmentEngine() self.protocol_selector =
DynamicProtocolSelectionEngine() self.bridge_manager =
SeamlessProtocolBridgeManager() self.vulnerability_scanner =
AutomatedVulnerabilityScanner() self.resource_optimizer =
AdaptiveResourceManager() self.migration_orchestrator =
IntelligentMigrationOrchestrator() # Initialize supporting
systems self.threat_intelligence =
ThreatIntelligenceAggregator() self.performance_monitor =
SystemPerformanceMonitor() self.compliance_manager =
ComplianceAndPolicyManager() self.analytics_engine =
SecurityAnalyticsEngine() # Initialize system state management
self.system_state = OrchestrationSystemState()
self.configuration_manager = DynamicConfigurationManager()
self.security_monitor = SecurityIntegrityMonitor() def
orchestrate_security_protocol(self, communication_request,
system_context): """Main orchestration entry point with
comprehensive security analysis""" try: # Pre-orchestration
system validation system_readiness =
self._verify_system_readiness() if not
system_readiness['ready']: return
self._handle_system_not_ready(system_readiness) # Real-time
quantum threat assessment quantum_assessment =
self.quantum_threat_engine.assess_current_threats(
communication_request, system_context ) # Device capability and
resource analysis resource_analysis =
self.resource_optimizer.analyze_system_capabilities(
communication_request, system_context ) # Vulnerability
assessment of current protocols vulnerability_analysis =
self.vulnerability_scanner.assess_vulnerabilities(
communication_request, system_context, quantum_assessment ) #
Dynamic protocol selection based on all factors
optimal_protocols =
self.protocol_selector.select_optimal_protocols(
quantum_assessment, resource_analysis, vulnerability_analysis,
communication_request ) # Protocol bridging and
interoperability management bridging_strategy =
self.bridge_manager.design_bridging_strategy(
optimal_protocols, system_context ) # Execute comprehensive
security orchestration orchestration_result =
self._execute_security_orchestration( optimal_protocols,
bridging_strategy, quantum_assessment, resource_analysis ) #
Post-orchestration learning and optimization learning_result =
self._apply_orchestration_learning( orchestration_result,
quantum_assessment ) # Update system performance metrics
self.performance_monitor.record_orchestration_event(orchestration_result)
return orchestration_result except Exception as e: return
self._handle_orchestration_error(e, communication_request)
```

The Dynamic Multi-Protocol Security Orchestration System comprises several integrated components:

1. Quantum Threat Assessment Engine

Real-Time Quantum Computing Threat Analysis:

```
python class
QuantumThreatAssessmentEngine: """Real-time assessment of
quantum computing threats to cryptographic systems""" def
assess_quantum_threat_landscape(self, target_systems,
threat_context): # Monitor quantum computing capabilities
quantum_capabilities =
self._monitor_quantum_computing_progress() # Analyze threat
intelligence feeds threat_intelligence =
self._aggregate_quantum_threat_intelligence() # Assess timeline
for quantum computer threats threat_timeline =
self._assess_quantum_threat_timeline(quantum_capabilities) #
Evaluate specific algorithm vulnerabilities
algorithm_vulnerabilities =
self._assess_algorithm_vulnerabilities( target_systems,
quantum_capabilities ) return { 'quantum_threat_level':
self._calculate_overall_threat_level( quantum_capabilities,
threat_timeline, algorithm_vulnerabilities ),
'quantum_capabilities': quantum_capabilities,
'threat_timeline': threat_timeline,
'algorithm_vulnerabilities': algorithm_vulnerabilities,
'recommended_actions':
self._generate_threat_response_recommendations(
threat_timeline, algorithm_vulnerabilities ) } def
_monitor_quantum_computing_progress(self): """Monitor global
quantum computing capability development""" return {
'qubit_count_progression': self._track_qubit_development(),
'error_rate_improvements': self._track_error_rate_progress(),
'quantum_volume_metrics':
self._assess_quantum_volume_progress(),
'algorithm_implementation_status':
self._monitor_shor_grover_implementations(),
'commercial_availability':
self._assess_quantum_computer_availability() }
```

2. Dynamic Protocol Selection Engine

Intelligent Cryptographic Protocol Selection:

```
python class
DynamicProtocolSelectionEngine: """Intelligent selection of
optimal cryptographic protocols based on multiple factors"""
def select_optimal_protocols(self, quantum_assessment,
resource_constraints, performance_requirements): # Analyze
available cryptographic protocols available_protocols =
self._catalog_available_protocols() # Score protocols against
quantum resistance quantum_scores =
self._score_quantum_resistance( available_protocols,
quantum_assessment ) # Evaluate protocols against resource
constraints resource_compatibility =
self._assess_resource_compatibility( available_protocols,
resource_constraints ) # Analyze performance characteristics
performance_analysis = self._analyze_protocol_performance(
available_protocols, performance_requirements ) # Select
optimal protocol combination optimal_selection =
self._optimize_protocol_selection( quantum_scores,
resource_compatibility, performance_analysis ) return {
'primary_protocol': optimal_selection['primary'],
'fallback_protocols': optimal_selection['fallbacks'],
'protocol_configuration': optimal_selection['configuration'],
'selection_reasoning':
self._document_selection_reasoning(optimal_selection),
'expected_performance':
self._predict_protocol_performance(optimal_selection) } def
_optimize_protocol_selection(self, quantum_scores,
resource_scores, performance_scores): """Multi-objective
optimization for protocol selection""" optimization_weights = {
```

```
'quantum_resistance': 0.4, 'resource_efficiency': 0.3,
'performance': 0.3 } protocol_rankings = {} for protocol in
quantum_scores.keys(): composite_score = (
quantum_scores[protocol] *
optimization_weights['quantum_resistance'] +
resource_scores[protocol] *
optimization_weights['resource_efficiency'] +
performance_scores[protocol] *
optimization_weights['performance'] )
protocol_rankings[protocol] = composite_score return
self._select_protocols_from_rankings(protocol_rankings)
```

3. Seamless Protocol Bridge Manager

Interoperability Between Different Cryptographic Protocols:

```
python class
SeamlessProtocolBridgeManager: """Manages interoperability
between different cryptographic protocols""" def
create_protocol_bridge(self, source_protocol, target_protocol,
communication_context): # Analyze protocol compatibility
requirements compatibility_analysis =
self._analyze_protocol_compatibility( source_protocol,
target_protocol ) # Design bridge architecture
bridge_architecture = self._design_bridge_architecture(
source_protocol, target_protocol, compatibility_analysis ) #
Implement protocol translation layer translation_layer =
self._implement_protocol_translation( bridge_architecture,
communication_context ) # Configure security parameter mapping
security_mapping = self._configure_security_parameter_mapping(
source_protocol, target_protocol, translation_layer ) # Deploy
and validate bridge bridge_deployment =
self._deploy_and_validate_bridge( translation_layer,
security_mapping ) return { 'bridge_implementation':
bridge_deployment, 'translation_layer': translation_layer,
'security_mapping': security_mapping, 'bridge_performance':
self._measure_bridge_performance(bridge_deployment),
'maintenance_requirements':
self._define_bridge_maintenance(bridge_deployment) }
```

4. Automated Vulnerability Scanner

Real-Time Cryptographic Vulnerability Assessment:

```
python class
AutomatedVulnerabilityScanner: """Automated assessment of
cryptographic vulnerabilities""" def
perform_comprehensive_vulnerability_scan(self, target_systems,
quantum_context): # Scan for quantum-vulnerable algorithms
quantum_vulnerabilities = self._scan_quantum_vulnerabilities(
target_systems, quantum_context ) # Assess implementation
vulnerabilities implementation_vulnerabilities =
self._scan_implementation_vulnerabilities( target_systems ) #
Analyze configuration weaknesses configuration_vulnerabilities
= self._scan_configuration_vulnerabilities( target_systems ) #
Evaluate protocol-level vulnerabilities
protocol_vulnerabilities = self._scan_protocol_vulnerabilities(
target_systems, quantum_context ) # Generate comprehensive
vulnerability assessment vulnerability_report =
self._generate_vulnerability_report( quantum_vulnerabilities,
implementation_vulnerabilities, configuration_vulnerabilities,
protocol_vulnerabilities ) # Prioritize vulnerabilities by risk
and impact vulnerability_priorities =
self._prioritize_vulnerabilities(vulnerability_report) return {
'vulnerability_assessment': vulnerability_report,
'vulnerability_priorities': vulnerability_priorities,
'remediation_recommendations':
self._generate_remediation_recommendations(
vulnerability_priorities ), 'quantum_risk_timeline':
self._assess_quantum_risk_timeline( quantum_vulnerabilities )
}
```

5. Adaptive Resource Manager

Resource-Aware Cryptographic Operations:

```
python class
AdaptiveResourceManager: """Manages cryptographic operations
based on device capabilities and constraints""" def
optimize_cryptographic_resources(self, available_resources,
security_requirements): # Assess device computational
capabilities computational_assessment =
self._assess_computational_capabilities( available_resources )
# Analyze memory and storage constraints memory_analysis =
self._analyze_memory_constraints(available_resources) #
Evaluate network bandwidth and latency network_analysis =
self._evaluate_network_characteristics(available_resources) #
Optimize algorithm selection for resource constraints
resource_optimized_algorithms =
self._optimize_algorithm_selection( computational_assessment,
memory_analysis, network_analysis, security_requirements ) #
Configure adaptive performance parameters
performance_configuration =
self._configure_adaptive_performance(
resource_optimized_algorithms, available_resources ) return {
'optimized_algorithms': resource_optimized_algorithms,
'performance_configuration': performance_configuration,
'resource_utilization_forecast':
self._forecast_resource_utilization(
resource_optimized_algorithms ), 'scaling_recommendations':
self._generate_scaling_recommendations(
computational_assessment ) }
```

6. Intelligent Migration Orchestrator

Coordinated Migration from Vulnerable to Quantum-Safe Protocols:

```
python class
IntelligentMigrationOrchestrator: """Orchestrates intelligent
migration from quantum-vulnerable to quantum-safe protocols"""
def orchestrate_protocol_migration(self, current_systems,
target_security_level): # Analyze current cryptographic
deployment current_deployment_analysis =
self._analyze_current_deployment(current_systems) # Plan
migration phases and dependencies migration_plan =
self._plan_migration_phases( current_deployment_analysis,
target_security_level ) # Assess migration risks and mitigation
strategies risk_assessment =
self._assess_migration_risks(migration_plan) # Execute phased
migration with rollback capabilities migration_execution =
self._execute_phased_migration( migration_plan, risk_assessment
) # Monitor migration progress and performance
migration_monitoring =
self._monitor_migration_progress(migration_execution) #
Validate migration success and security posture
migration_validation = self._validate_migration_success(
migration_execution, target_security_level ) return {
'migration_execution_result': migration_execution,
'migration_monitoring': migration_monitoring,
'migration_validation': migration_validation,
'post_migration_optimization':
self._optimize_post_migration_performance( migration_validation
), 'rollback_capabilities':
self._maintain_rollback_capabilities(migration_plan) }
```

Advanced Threat Intelligence Integration

Multi-Source Threat Intelligence Aggregation

Comprehensive Threat Intelligence Collection and Analysis:

```
python class
ThreatIntelligenceAggregator: """Aggregates and analyzes threat
intelligence from multiple sources""" def
aggregate_quantum_threat_intelligence(self): # Government and
academic quantum research monitoring research_intelligence =
self._monitor_quantum_research_publications() # Commercial
quantum computer development tracking commercial_intelligence =
self._track_commercial_quantum_development() # Cybersecurity
threat feed integration cybersecurity_intelligence =
self._integrate_cybersecurity_feeds() # Patent and intellectual
property analysis patent_intelligence =
self._analyze_quantum_patent_landscape() # Social media and
news sentiment analysis public_sentiment =
self._analyze_quantum_public_sentiment() # Synthesize
comprehensive threat picture comprehensive_assessment =
self._synthesize_threat_intelligence( research_intelligence,
commercial_intelligence, cybersecurity_intelligence,
patent_intelligence, public_sentiment ) return {
'comprehensive_threat_assessment': comprehensive_assessment,
'threat_confidence_levels': self._calculate_threat_confidence(
comprehensive_assessment ), 'threat_timeline_predictions':
self._predict_threat_timeline( comprehensive_assessment ),
'actionable_intelligence':
self._extract_actionable_intelligence( comprehensive_assessment
) }
```

CLAIMS

1. A dynamic multi-protocol security orchestration system comprising: (a) a quantum threat assessment engine that continuously monitors quantum computing capabilities, threat intelligence, and algorithm vulnerabilities to generate real-time quantum threat assessments; (b) a dynamic protocol selection engine that intelligently selects optimal cryptographic protocols based on quantum resistance scores, resource compatibility, and performance requirements; (c) a seamless protocol bridge manager that enables interoperability between different cryptographic protocols through protocol translation layers and security parameter mapping; (d) an automated vulnerability scanner that performs comprehensive assessment of quantum vulnerabilities, implementation weaknesses, and configuration issues; (e) an adaptive resource manager that optimizes cryptographic operations based on device computational capabilities, memory constraints, and network characteristics; (f) an intelligent migration orchestrator that coordinates phased migration from quantum-vulnerable to quantum-safe protocols with rollback capabilities. **2.** The system of claim 1, wherein the quantum threat assessment engine further comprises: (a) quantum computing progress monitoring that tracks qubit development, error rate improvements, quantum volume metrics, and algorithm implementation status; (b) threat intelligence aggregation from government research, commercial development, cybersecurity feeds, patent analysis, and public sentiment monitoring; (c) threat timeline prediction algorithms that assess quantum computer threat emergence timelines; (d) algorithm vulnerability analysis that evaluates specific cryptographic algorithm susceptibility to quantum attacks. **3.** The system of claim 1, wherein the dynamic protocol selection engine further comprises: (a) multi-objective optimization algorithms that balance quantum resistance, resource efficiency, and performance factors; (b) protocol cataloging systems that maintain comprehensive databases of available cryptographic protocols and their characteristics; (c) resource compatibility assessment that evaluates protocol suitability for specific device capabilities and constraints; (d) performance prediction models that forecast protocol performance under various operational conditions. **4.** The system of claim 1, wherein the seamless protocol bridge manager further comprises: (a) protocol compatibility analysis systems that assess interoperability requirements between different cryptographic protocols; (b) bridge architecture design algorithms that create optimal translation layers for protocol interoperability; (c) security parameter mapping systems that maintain security equivalence across protocol boundaries; (d) bridge performance monitoring that ensures optimal operation and identifies optimization opportunities. **5.** The system of claim 1, wherein the automated vulnerability scanner further comprises: (a) quantum vulnerability detection that identifies quantum-susceptible algorithms and implementations; (b) implementation vulnerability analysis that assesses cryptographic library and code implementation weaknesses; (c) configuration security scanning that identifies misconfigurations and policy violations; (d) vulnerability prioritization algorithms that rank vulnerabilities by quantum risk timeline and impact severity. **6.** A method for dynamic multi-protocol security orchestration comprising: (a) continuously assessing quantum computing threats through monitoring quantum capabilities, threat intelligence aggregation, and vulnerability analysis; (b) dynamically selecting optimal cryptographic protocols based on quantum resistance, resource constraints, and performance requirements; (c) implementing seamless protocol bridging to enable communication between systems using different cryptographic protocols; (d) performing automated vulnerability scanning to identify and prioritize quantum and implementation vulnerabilities; (e) optimizing cryptographic resource utilization based on device capabilities and operational constraints; (f) orchestrating intelligent migration from quantum-vulnerable to quantum-safe protocols with comprehensive risk management. **7.** The method of claim 6, further comprising: (a) aggregating threat intelligence from research publications, commercial development, cybersecurity feeds, patent landscapes, and public sentiment analysis; (b) predicting quantum threat timelines through analysis of technological progress indicators and capability assessments; (c) generating actionable threat intelligence with confidence levels and recommended response strategies; (d) maintaining continuous threat monitoring with real-time updates and alert generation. **8.** The method of claim 6, further comprising: (a) performing multi-objective optimization for protocol selection weighing quantum resistance, resource efficiency, and performance factors; (b) implementing adaptive algorithm selection that responds to changing threat levels and operational conditions; (c) maintaining protocol performance databases with historical performance data and predictive models; (d) providing protocol selection reasoning documentation for audit and compliance purposes. **9.** The system of claim 1, further comprising a threat intelligence aggregator that: (a) monitors quantum research publications and academic developments; (b) tracks commercial quantum computer development and availability; (c) integrates cybersecurity threat feeds and vulnerability databases; (d) analyzes quantum-related patent landscapes and intellectual property developments; (e) performs sentiment analysis of quantum computing public discourse and media coverage. **10.** The system of claim 1, further comprising a performance monitoring system that: (a) tracks system orchestration performance metrics including latency, throughput, and resource utilization; (b) monitors protocol bridge performance and identifies optimization opportunities; (c) analyzes migration success rates and performance impact assessments; (d) generates performance reports and recommendations for system optimization. **11.** The method of claim 6, further comprising: (a) implementing protocol bridge validation to ensure security parameter consistency across protocol boundaries; (b) performing bridge performance optimization through adaptive parameter tuning and load balancing; (c) maintaining bridge security monitoring to detect and respond to interoperability vulnerabilities; (d) providing bridge maintenance automation including updates, patches, and configuration management. **12.** The system of claim 1, wherein the adaptive resource manager further comprises: (a) computational capability assessment algorithms that evaluate device processing power, memory, and storage resources; (b) network characteristic analysis that assesses bandwidth, latency, and connectivity patterns; (c) algorithm optimization engines that select cryptographic algorithms based on resource constraints; (d) performance scaling systems that adapt cryptographic operations to varying resource availability. **13.** The system of claim 1, wherein the intelligent migration orchestrator further comprises: (a) migration planning algorithms that analyze current deployments and design phased migration strategies; (b) risk assessment systems that evaluate migration risks and develop mitigation strategies; (c) phased execution engines that coordinate migration activities with rollback capabilities; (d) migration validation systems that verify successful protocol transitions and security posture. **14.** A quantum-aware protocol selection method comprising: (a) monitoring global quantum computing capability development including qubit counts, error rates, and algorithm implementations; (b) assessing cryptographic algorithm quantum vulnerability based on current and projected quantum computing capabilities; (c) selecting optimal cryptographic protocols through multi-factor analysis of quantum resistance, performance, and resource requirements; (d) implementing adaptive protocol switching based on real-time threat assessment and operational conditions. **15.** The method of claim 14, further comprising: (a) maintaining quantum threat databases with continuous updates from research, commercial, and intelligence sources; (b) performing predictive threat modeling to forecast quantum computer capability timelines; (c) generating quantum risk assessments for specific cryptographic deployments and use cases; (d) providing quantum threat visualization and reporting for decision-making and compliance. **16.** The system of claim 1, further comprising compliance and policy management systems that: (a) ensure cryptographic protocol selections comply with regulatory requirements and industry standards; (b) maintain audit trails of protocol selection decisions and migration activities; (c) implement policy-driven protocol restrictions and preferences based on organizational security policies; (d) generate compliance reports and documentation for regulatory and audit purposes. **17.** The system of claim 1, further comprising security analytics engines that: (a) analyze historical security events and protocol performance data to identify patterns and trends; (b) perform predictive analytics to forecast security requirements and optimal protocol configurations; (c) implement machine learning algorithms for continuous improvement of protocol selection and threat assessment; (d) provide security dashboards and visualization tools for system monitoring and management. **18.** A method for seamless cryptographic protocol interoperability comprising:

(a) analyzing protocol compatibility requirements between source and target cryptographic systems; (b) designing bridge architectures that enable secure communication across protocol boundaries; (c) implementing protocol translation layers that maintain security equivalence during protocol conversion; (d) monitoring bridge performance and security to ensure optimal interoperability and threat protection. **19.** The method of claim 18, further comprising: (a) validating security parameter consistency across protocol bridges to prevent security degradation; (b) implementing adaptive bridge optimization based on communication patterns and performance requirements; (c) maintaining bridge security through continuous monitoring and threat detection; (d) providing bridge management automation including deployment, configuration, and maintenance. **20.** The system of claim 1, wherein the system provides enterprise deployment capabilities comprising: (a) scalable architecture supporting deployment across large organizational infrastructures; (b) centralized management interfaces for system configuration, monitoring, and control; (c) integration capabilities with existing security infrastructure and management systems; (d) high availability and disaster recovery features ensuring continuous security orchestration operation.

DRAWINGS

[FIGURE 1: System Architecture Overview - Shows the complete dynamic multi-protocol security orchestration system with all major components and their interactions] [FIGURE 2: Quantum Threat Assessment Engine - Detailed view of threat intelligence aggregation, quantum capability monitoring, and threat timeline prediction] [FIGURE 3: Dynamic Protocol Selection Process - Flowchart showing multi-objective optimization for protocol selection based on quantum resistance, resources, and performance] [FIGURE 4: Seamless Protocol Bridging Architecture - Technical diagram of protocol translation layers and security parameter mapping] [FIGURE 5: Automated Vulnerability Scanning Process - Comprehensive vulnerability assessment workflow with prioritization and remediation recommendations] [FIGURE 6: Adaptive Resource Management - Resource-aware algorithm selection and performance optimization across diverse device capabilities] [FIGURE 7: Intelligent Migration Orchestration - Phased migration strategy with risk assessment and rollback capabilities] [FIGURE 8: Enterprise Integration Architecture - Scalable deployment architecture with centralized management and monitoring capabilities]

ATTORNEY DOCKET: MWRASP-001-PROV **FILING DATE:** September 3, 2025 **SPECIFICATION:** 74 pages **CLAIMS:** 20 **FIGURES:** 8 technical diagrams **ESTIMATED VALUE:** $500-800 Million

REVOLUTIONARY BREAKTHROUGH: First dynamic multi-protocol security orchestration system with real-time quantum threat assessment, automated protocol selection, and seamless interoperability for quantum-safe cryptographic infrastructure.