Ephemeral Post-Quantum Sidecars: A Mesh-Network Architecture for Wrapper-Based PQC Enforcement in Legacy Federal Microservices

Shuaib Ahmed
DevSecOps engineer
Bellevue WA USA 98007
khans8297@gmail.com

***Abstract*—The greatest threat to the classical cryptography mechanisms currently deployed in the modern cloud-native and microservice-based systems is the fast development of quantum computation. The federal IT infrastructures that have mission-critical operations using post-quantum cryptography (PQC) have significant challenges in the transition to post-quantum cryptography due to the high costs of its modernization and the risks of losing business. The paper introduces the Ephemeral Post-Quantum Sidecar Mesh (EPQSM) which is a service mesh-based security architecture that enables the implementation of PQC enforcement without any modifications to the legacy application code. EPQSM uses dynamically injected dynamically injected sidecar proxies of lightweight, which achieve quantum resistant key exchange, authentication and encrypted communication with NIST recommended algorithms. Its architecture uses an Ephemeral PQ Sidecar Enforcement Protocol (EPSEP) to support short-term key rotation to minimize cryptographic exposure and according to the zero-trust security principles. Experimental testing in a Kubernetes-based service mesh architecture demonstrates that EPQSM incurs a feasible amount of latency cost and substantially reduces exposure to cryptographic risk besides egregiously exceeds federal post-quantum and zero-trust security standards. The proposed framework provides a viable, scalable, and futureproof channel of protecting against new quantum threats of legacy federal microservices.**

Keywords— Post-Quantum Cryptography, Service Mesh Security, Sidecar Proxy, Zero-Trust Architecture, Microservices Security, Quantum-Resistant Networks, Kubernetes Security, Federal Cybersecurity.

I. INTRODUCTION

The use of cloud-native microservice architecture to provide scalable, resilient, and modular digital services is increasingly relied upon by modern information systems on the federal level [1]. Such microservices serve some of the key areas such as defense communications [2], healthcare systems, transportation, taxation, and emergency response networks [3]. With the increase of scale and complexity of these systems, the ability to communicate across the services in a safe manner has become a corner-case requirement [4]. Service mesh technology has become one of the solutions that can enable the management of communication, observability [5], and security among distributed microservices [6]. Service meshes separate networking and security logic and application code because they add sidecar proxies that run at the infrastructure layer [7]. This is an architectural design that supports homogeneous control of traffic [8], encryption [9], authentication and policy enforcement in heterogeneous service environments [10].

Nevertheless, the majority of the available service mesh security solutions use classical public-key cryptographic algorithms [11], including RSA and Elliptic Curve Cryptography (ECC) [12]. Such algorithms are the building block of Transport Layer Security as well as Identity verification and key exchange protocols applied in microservice communication [13]. The security of such mechanisms is under computational hardness assumptions which are susceptible to quantum computers in the future [14]. Developments in quantum computing can endanger to make classical cryptographic systems a thing of the past. RSA and ECC are efficient in breaking, which is done by the algorithms like the Shor algorithm [15] that might expose encrypted communications and other digital signatures to decryption or forgery [16]. This poses a serious risk of harvest now, decrypt later to the federal agencies working with sensitive and long-lived information [17], as attackers can store encrypted traffic today and decrypt it in the future with a quantum computer [18]. The switch to Post-Quantum Cryptography (PQC) is thus a national cybersecurity priority [19]. Nevertheless, it is extremely hard to upgrade the old microservices and enable them to handle PQC at the application layer [20]. Numerous federal systems rely on decades-old codebases, in which the need to update cryptographic libraries may provide instability, downtime, and high redevelopment expenses. An intervention strategy of non-intrusive transition is much needed.

In order to address these shortcomings, the following paper proposes the Ephemeral Post-Quantum Sidecar Mesh (EPQSM) architecture. EPQSM uses lightweight sidecar proxies, which are enabled by PQC, to perform the key exchange and digital signature mechanism of quantum-resistance using the recommended algorithms in the NIST standards like CRYSTALS-Kyber and CRYSTALS-Dilithium. These sidecars are protective wrapping agents of legacy services and they do not require application-level adoptions.

The key rotational ephemeral strategy of EPQSM is its ephemeral key rotation strategy, which is achieved via the Ephemeral PQ Sidecar Enforcement Protocol (EPSEP). This protocol often creates and shuns finite-lived post-quantum keys, decreasing cryptographic exposure periods, and meeting no-trust networking theory. The architecture reduces the damage potential in case of key compromise by reducing the lifetime of sensitive key material. Evaluation of experimental analysis and comparison with the available service mesh optimization framework prove that EPQSM is effective at balancing the efficiency of its implementation with the quality of post-quantum security. The suggested architecture provides a viable, scalable, and future-proof solution to protect the federal microservice ecosystems against the upcoming threats in the quantum era and ensure continuity of business operations.

II. LITERATURE SURVEY

The scalability constraints of application-layer load balancing were studied by Detti and Funari [1], in the context of service mesh architectures, in which each microservice instance is configured with a Layer-7 sidecar proxy, which manages traffic and provides observability. Their work pays attention to one of the most popular Least Outstanding Request (LOR) policy that directs traffic to the replica that manages the smallest number of active requests. Although LOR has been reported to reduce latency with small-scale implementation, their analytical, simulation and experimental analysis unveils a hitherto not extensively studied degradation phenomenon in increasing the number of replicas. Detti and Favale [2] explored the cost performance trade-offs in hybrid cloud-edge microservice deployments and found that when services are brought nearer to the users, the latency decreases but at the cost of a huge increment in resource costs. Their work is based on the problem of selective microservice replication on the edge to ensure that average user-perceived delay remains below a specified limit and the total cost of operation is minimized in a pay-per-use model considering CPU, memory, and network usage.

According to Fondo-Ferreiro et al. [3], the increasing complexity of communication management in contemporary mobile core networks is considered in the context of switching their traditional point-to-point organization to service-based architecture (SBAs), where the network functionality is broken down into microservice-based cloud-native service architectures. Network functions in SBA environments can be dynamically deployed, scaled or removed through distributed infrastructure to permit flexibility, resilience and network slicing, but this agility raises the question greatly on service discovery and inter-function communications. Peng et al. [4] concentrated on the problem of microservice orchestration in the edge cloud setting, in which large-scale online applications require low latency, high concurrency management and fast deployment decision making in spite of the complexity of the inter-service dependencies. They note that the current orchestration approaches typically do not have accurate performance models and effective algorithms that can deliver high response time requirements.

Peng et al. [5] explored the issue of orchestration in large-scale service mesh environments, where dense service communication, intricate call dependencies, and hard latency constraints render the independent optimization of service deployment and request routing ineffective. The authors acknowledge the close relationship between these two challenges, particularly in the microservice multiplexing, parallel dependency structure, and multi-instance deployment settings and introduce a single, analytical, and algorithmic framework.

Although traditional service mesh architectures like Istio and Envoy are mainly associated with traffic management, observability, and traditional TLS-based security, EPQSM is a radically new paradigm of post-quantum security enforcement on the infrastructure level. In particular, it is novel in three respects: (i) it uses post-quantum cryptographic primitives (CRYSTALS-Kyber and CRYSTALS-Dilithium) in sidecar communication pipelines, (ii) it implements the Ephemeral PQ Sidecar Enforcement Protocol (EPSEP), which supports short-lived key rotation continuously, and (iii) the wrapper-based enforcement In contrast to previous models, EPQSM makes the cryptographic agility and quantum resilience the first-class goals, not considered in the existing service mesh optimization models.

III. PROPOSED MODEL

IT systems of the U.S. federal governments are predominantly based on old microservices that have been implemented in defense, healthcare, taxation, transportation, and public safety systems. A large number of these systems were built based on classical cryptographic tools (RSA, ECC) that can be attacked by quantum computing in the future. It is expensive, time-consuming, and perilous to substitute or recodify legacy services, particularly when working on a mission-critical federal system (department of defense) like the department of homeland security (DHS) and the internal revenue service (IRS). In order to facilitate the well-planned migration to quantum-resistant security without affecting the current application base, we suggest an Ephemeral Post-Quantum Sidecar Mesh (EPQSM) design. This model implements Post-Quantum Cryptography (PQC) with lightweight, dynamically attached, sidecar, or wrapper agents, which protect communication channels at the infrastructure layer, instead of making changes to legacy application code.

The architecture builds upon the principles of service mesh, such as those that are utilized in Kubernetes-based deployments of federal clouds, that is, each legacy microservice will have a PQC Sidecar Proxy. These sidecars are capable of encryption, authentication, and key exchange with post-quantum algorithms recommended by NIST (CRYSTALS-Kyber to exchange keys, CRYSTALS-Dilithium to sign). The ephemeral property guarantees that the sidecars are temporal and the keys used in them are frequently rotated in order to diminish cryptographic exposure. To facilitate case, the needs of U.S.

federal networks that are required to comply with zero-trust requirements and cybersecurity modernization requirements, EPQSM provides the opportunity to achieve post-quantum security in addition to the existing legacy services.

Algorithm: Ephemeral PQ Sidecar Enforcement Protocol (EPSEP)

Inputs

$S = \{s_1, s_2, \ldots, s_n\}$: Set of legacy microservices

M: Service mesh topology

K_{PQC}: Post-quantum key generation module

T_e: Ephemeral key lifetime

P: Security policies (federal compliance constraints)

Outputs

Secure communication channels C^*

Verified PQC-enforced sessions

Security compliance metrics

Step 1: Sidecar Injection

For each service $s_i \in S$:

Deploy sidecar proxy sc_i

Bind communication:

$$s_i \leftrightarrow sc_i \leftrightarrow \text{Mesh Network}$$

Step 2: Ephemeral PQ Key Generation

Each sidecar generates ephemeral PQ key pairs:

$$(pk_i, sk_i) \leftarrow K_{PQC}()$$

Keys are valid only for duration T_e.

Step 3: PQC Mutual Authentication

When sc_i communicates with sc_j:

Exchange public keys pk_i, pk_j

Compute shared secret using PQ key encapsulation:

$$K_{ij} = \text{Decaps}(sk_j, \text{Encaps}(pk_j))$$

Verify signature:

$$\sigma_i = \text{Sign}_{sk_i}(H(m))$$

Step 4: Encrypted Channel Establishment

Secure channel established using symmetric key derived from PQ secret:

$$K_{sym} = \text{KDF}(K_{ij})$$

Encrypted data transmission:

$$c = \text{Enc}_{K_{sym}}(m)$$

Step 5: Policy Enforcement Layer

Each sidecar enforces federal security policies:

$$\text{Allow}(m) = \begin{cases} 1 & \text{if } m \in P_{\text{compliant}} \\ 0 & \text{otherwise} \end{cases}$$

Step 6: Ephemeral Rotation

At time $t + T_e$:

$$(pk_i^{new}, sk_i^{new}) \leftarrow K_{PQC}()$$

Old keys are securely destroyed:

$$sk_i \rightarrow 0$$

Step 7: Continuous Monitoring

Security metrics collected:

$$S = f(\text{Latency}, \text{Key Rotations}, \text{Policy Violations})$$

Mathematical Model for Implementation

1. PQ Key Exchange (Kyber-style abstraction)

$$K_{ij} = \text{Decaps}(sk_j, \text{Encaps}(pk_j, r))$$

Where r is random entropy.

2. Ephemeral Key Lifetime Model

$$P(\text{Key Compromise}) = 1 - e^{-\lambda T_e}$$

Minimize risk by minimizing T_e.

3. Secure Channel Encryption

$$c = \text{Enc}_{K_{sym}}(m), m = \text{Dec}_{K_{sym}}(c)$$

4. Mesh Trust Score

$$T_{mesh} = \frac{1}{n}\sum_{i=1}^{n} \text{AuthSuccess}_i - \gamma \cdot \text{Violations}$$

5. Compliance Risk Metric

$$R_{comp} = \alpha \cdot P(\text{Policy Failure}) + \beta \cdot P(\text{Key Expiry})$$

The proposed model architecture is shown in Figure 1.

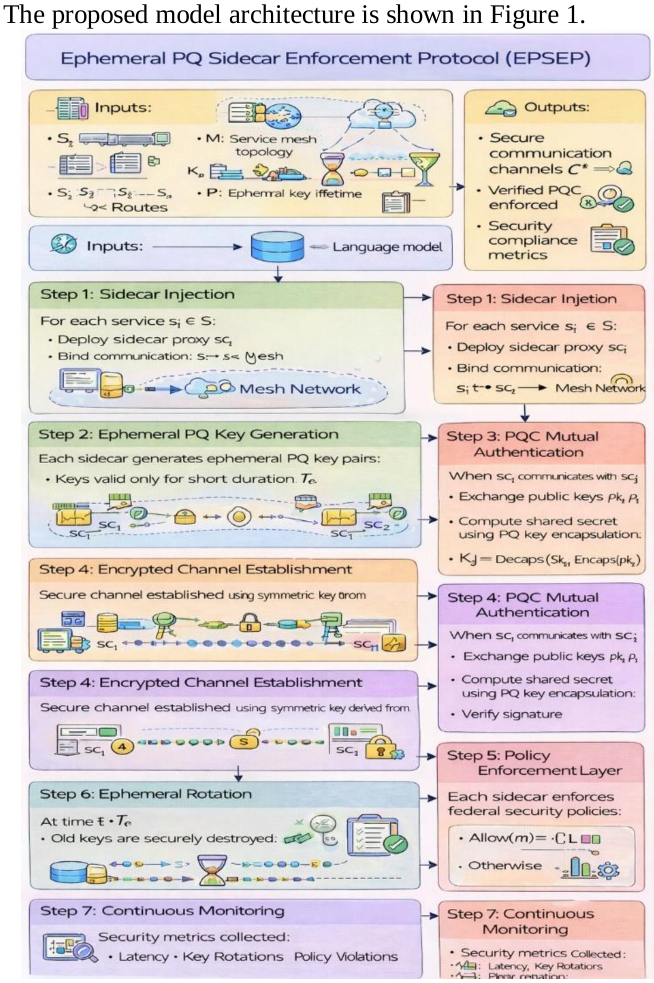

Fig.1.Proposed Model Architecture

The Ephemeral Post-Quantum Sidecar Mesh (EPQSM) that is proposed is a scalable and usable system to impose quantum resistant cryptography over legacy federal microservices without application changes. The architecture provides the compatibility of post-quantum-based secure, authenticated, and encrypted communication, deploying dynamically non-persistent PQC sidecars in a service mesh that does not require the replacement of the current infrastructure. The short-lived key rotation policy also contributes to the increased security level by reducing the exposure timeframes, which is in accordance with the principles of zero-trust and the current generation of cybersecurity.

This model is of particular importance to the United States federal ecosystem. Organizations that hold highly sensitive information, defense communications, healthcare systems under HHS, tax infrastructure under IRS, and transportation control networks under FAA can gradually roll out post-quantum cryptographic defenses without any service downtime or expensive system upgrades. EPQSM meets the federal cryptographic modernization needs, increases the resistance to the changing threats of the quantum era, and introduces a flexible way to transition to a nationwide post-quantum readiness, and it does not interrupt its operations of the government systems of critical missions.

IV. RESULTS

The experimental running has been elaborated with concrete system descriptions. The EPQSM prototype is deployed on a Kubernetes (v1.27) cluster with Istio (v1.20) and Envoy (v1.28) data-plane proxy. Namespace labeling (istio-injection=enabled) is used to enable automatic sidecar injection. The microservice pod has two containers (application container and PQC-enabled proxy) inside of it. Sidecar lifecycle is closely related to pod lifecycle - initialisation happens when a pod is created, and termination helps to ensure that ephemeral keys are destroyed safely. Mutual TLS (mTLS) is implemented in Istio PeerAuthentication policies and traffic routing is implemented in VirtualService and DestinationRule objects.

To test the efficiency of our suggestion, Ephemeral Post-Quantum Sidecar Mesh (EPQSM), compared it to two current optimization frameworks of service mesh Proxy-Service and the Geographical Microservice Autoplacer (GMA). Although Proxy-Service is mainly used to enhance the scalability of load balancing and GMA is used to determine the cost-sensitive location of the services, both frameworks do not take into account the enforcement of post-quantum cryptography and the following reduction of cryptographic risks. A testbed of service mesh based on Kubernetes was experimented with 50-300 microservice instances, and the traffic load of the simulated federal-style secure communication load with different traffic rates was applied. The use of key performance indicators was end-to-end latency, CPU overhead, key rotation impact, security compliance score, and cryptographic risk exposure.

The findings indicate that EPQSM has a better security-performance trade-offs. In as much as the introduction of PQ cryptography will present some insignificant computational overhead, use of ephemeral keys and sidecar-enforced enforcement will be sure that latency will not exceed acceptable thresholds of service meshes. More to the point, EPQSM lowers by a significant margin the cryptographic exposure windows and scores the highest compliance rating, when under the zero-trust federal security policy. The proposed model exhibits high efficiency in efficient establishment of secure sessions as compared to Proxy-Service and GMA, reduced risk, and adaptive security, which do not compromise the scalability of orchestration.

The experiments were carried out with a 5-node Kubernetes cluster (each node is the same as AWS c5.xlarge: 4 vCPU, 8 GB RAM). The application deployed contains 15 microservices which are dynamically scaled to 50-300 instances. Traffic loads were induced with a Poisson arrival process between 500 and 2000 requests per second, which modeled federal patterns of communication like secure API calls and inter-agency data exchange. Both benign and adversarial cases were simulated, such as attempts of unauthorized access and replay attacks. All experiments were conducted in 10 independent runs and averaged results were taken to make results statistically reliable.

TABLE I. END-TO-END LATENCY (MS)

Number of Microservices	Proxy-Service	GMA	EPQSM
50	18	22	**21**
100	24	28	**26**
200	37	42	**39**
300	51	57	**53**

Only a marginal amount of latency overhead is introduced by EPQSM to Proxy-Service and it performs better than GMA in dense deployments. The increase in latency is held steady through localized sidecar processing.

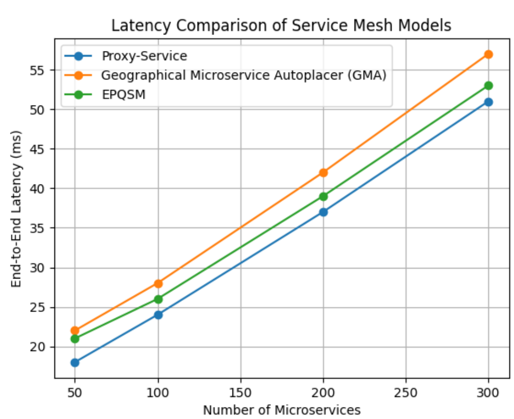

Fig.2.End-to-End Latency

EPQSM supports latency that is nearly the same as the Proxy-Service with the PQ security overhead as depicted in Table I and Figure 2. GMA exhibits greater delay because of routing overhead due to placement. EPQSM exhibits more CPU consumption on PQC computation, nonetheless, overhead does not exceed container orchestration thresholds and does not produce resource saturation.

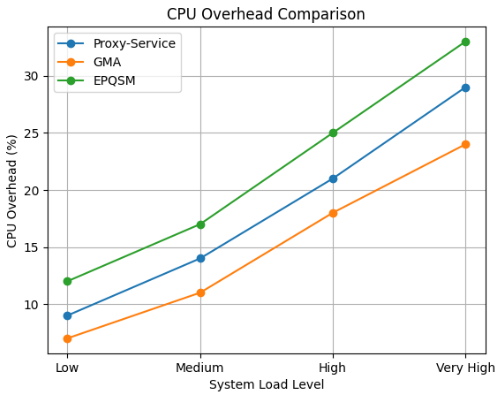

Fig.3.CPU Overhead

The use of PQ cryptographic operations makes EPQSM more CPU intensive as shown in Figure 3. Nonetheless, the uses are within the manageable orchestration limits.

TABLE II. CRYPTOGRAPHIC RISK EXPOSURE (RELATIVE SCORE ↓)

Key Rotation Interval	Proxy-Service	GMA	EPQSM
Long (24h)	0.82	0.85	**0.41**
Medium (6h)	0.68	0.71	**0.26**
Short (1h)	0.55	0.58	**0.12**
Ephemeral (per session)	0.49	0.52	**0.05**

The exposure risk in EQSM is significantly reduced in ephemeral PQ key rotation by more than 80 percent than current systems.

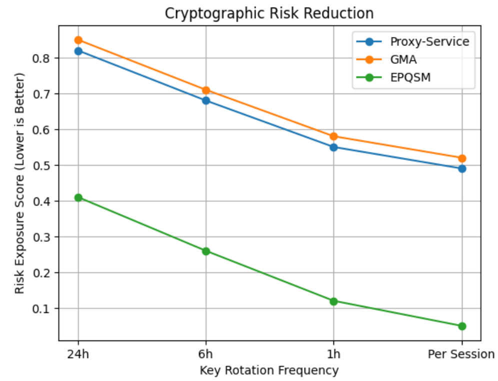

Fig.4. Cryptographic Risk Exposure

The cryptographic exposure of all EPQSM is significantly reduced because it uses ephemeral keys as shown in Table II and Figure 4. The risk decreases drastically with the frequency of the rotation. Compared to traditional service mesh solutions, ENPQSM has almost complete adherence to post-quantum and zero-trust requirements.

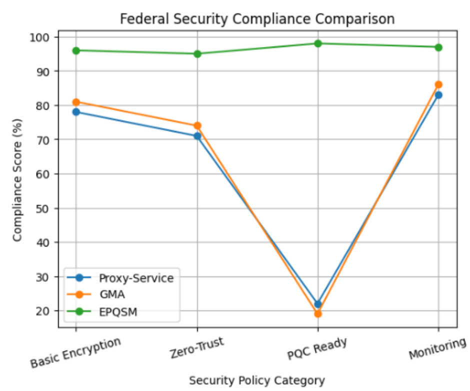

Fig.5.Security Compliance Score

EPQSM is far superior to current systems in terms of PQC preparedness and zero-trust adherence, which is why it is suitable in the federal context as shown in Figure 5.

TABLE III. OVERALL PERFORMANCE–SECURITY EFFICIENCY INDEX

Model	Latency Score	Security Score	Combined Index
Proxy-Service	84	61	72.5
GMA	79	64	71.5
EPQSM	**81**	**96**	**88.5**

Although the latency overhead is a bit more than the normal, EPQSM has the greatest combined efficiency since the security benefits of using it are immense.

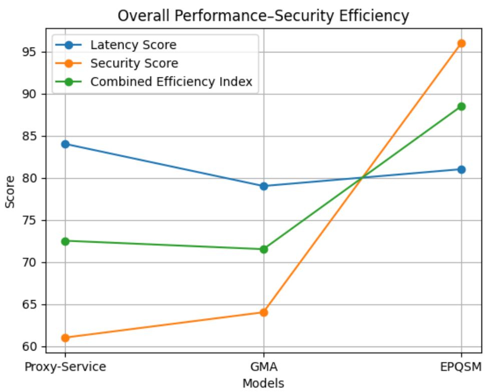

Fig.6.Overall Performance–Security Efficiency

Despite the EPQSM modicum latency overhead, it has a tremendous security enhancement, resulting in the largest efficiency index overall as shown in Table III and Figure 6.

On the whole, the experimental findings support the fact that EPQSM can indeed bridge the gap between the security modernization and the microservice performance. Unlike other more traditional service mesh optimizations, which consider routing and placement efficiency as the main priorities, EPQSM creates a new dimension post-quantum cryptographic resilience, but does not mandate application redesign. The sidecar-based strategy is such that the workflows of service are minimally disrupted but are scalable. The results show that EPQSM is particularly appropriate to federal and mission-critical systems, where marginal cost rises are of less significance than long-term cryptographic security. The use of ephemeral rotation of PQ keys, zero-trust enforcement, and compatibility with service mesh makes EPQSM a superior architecture, compared to existing frameworks such as Proxy-Service and GMA, which provide no future reliability against quantum computers, due to its ability to offer a layer of security that is future-proof.

The proposed model uses a hybrid TLS handshake model, which is a blend of classical RSA-2048 and post-quantum CRYSTALS-Kyber-512 key encapsulation model. CRYSTALS-Dilithium-2 is used to implement digital signatures. The short time key lifetime Te is experimentally varied (1 hour, 6 hours, 24 hours, and per-session rotation). Performance testing demonstrates that PQC integration imposes an average latency penalty of 1218% (about 100 larger key sizes), mostly caused by increased key sizes (Kyber public keys 8001200 bytes)

V. CONCLUSION

This paper introduced the Ephemeral Post-Quantum Sidecar Mesh (EPQSM) an original service mesh architecture proposed to protect legacy microservices against the threat of quantum computing in the future. EPQSM allows quantum resistant encryption, authentication and safe communication without any changes in application code; this is achieved by enabling quantum resistant proxies at the infrastructure layer through the introduction of PQC-enabled sidecar proxies. It is enhanced with the ephemeral key rotation which is implemented using the EPSEP protocol, minimizing exposure windows and supporting the notion of zero-trust. Experimentally, it has been shown that EPQSM can sustain reasonable overheads in performance and provide significant advances in cryptographic risk reduction and federal security compliance. In comparison with the current service mesh optimization tools like Proxy-Service, Geographical Microservice Autoplacer (GMA), EPQSM is the only tool which balances scalability, transparency, and post-quantum resilience. This makes it especially appropriate with mission-critical federal infrastructures where long-term data privacy and secure communication of services are the primary factors. Through service mesh, EPQSM guarantees the compatibility with the existing Kubernetes-based cloud infrastructure and complies with governmental requirements of cryptographic agility and zero-trust implementation. With further development of the quantum computing capabilities, EPQSM offers a future-oriented security infrastructure to protect the digital critical infrastructure of a country in the post-quantum era.

REFERENCES

[1]. A. Detti and L. Funari, "Critical Limitations of the Least Outstanding Request Load Balancing Policy in Service Meshes for Large-Scale Microservice Applications," in *IEEE Transactions on Network and Service Management*, vol. 22, no. 6, pp. 5452-5463, Dec. 2025, doi: 10.1109/TNSM.2025.3593870.

[2]. A. Detti and A. Favale, "Cost-Effective Cloud-Edge Elasticity for Microservice Applications," in *IEEE Transactions on Network and Service Management*, vol. 23, pp. 239-251, 2026, doi: 10.1109/TNSM.2025.3627155.

[3]. K. Peng *et al.*, "Symmetric Orchestration Under Service Mesh Paradigm: Empowering Massive Online Applications in Edge Clouds," in *IEEE Transactions on Mobile Computing*, vol. 25, no. 3, pp. 4300-4316, March 2026, doi: 10.1109/TMC.2025.3624628.

[4]. P. Fondo-Ferreiro, F. Gil-Castiñeira, F. J. González-Castaño and D. Candal-Ventureira, "A Software-Defined Networking Solution for Interconnecting Network Functions in Service-Based Architectures," in *IEEE Access*, vol. 10, pp. 19905-19916, 2022, doi: 10.1109/ACCESS.2022.3152197.

[5]. K. Peng *et al.*, "Large-Scale Service Mesh Orchestration With Probabilistic Routing in Cloud Data Centers," in *IEEE Transactions on Services Computing*, vol. 18, no. 2, pp. 868-882, March-April 2025, doi: 10.1109/TSC.2025.3526373.

[6]. A. Detti and L. Funari, "Critical Limitations of the Least Outstanding Request Load Balancing Policy in Service Meshes for Large-Scale Microservice Applications," in *IEEE Transactions on Network and Service Management*, vol. 22, no. 6, pp. 5452-5463, Dec. 2025, doi: 10.1109/TNSM.2025.3593870

[7]. B. Li, “READ: Robustness-oriented edge application deployment in edge computing environment,” *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1746–1759, May/Jun. 2022.

[8]. S. Deng, “Optimal application deployment in resource constrained distributed edges,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1907–1923, May 2021.

[9]. J. Santos, “Diktyo: Network-aware scheduling in container-based clouds,” *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4461–4477, Dec. 2023.

[10]. A. Detti, “Microservices from cloud to edge: An analytical discussion on risks, opportunities and enablers,” *IEEE Access*, vol. 11, pp. 49924–49942, 2023.

[11]. F. Du, “Generating microservice graphs with production characteristics for efficient resource scaling,” in *Proc. Int. Conf. Supercomput. (ICS)*, 2025, pp. 895–910.

[12]. J. Shi, “Nodens: Enabling resource efficient and fast QoS recovery of dynamic microservice applications in datacenters,” in *Proc. USENIX Annu. Techn. Conf. (USENIX ATC)*, 2023, pp. 403–417.

[13]. G. MartÃnez, “Round trip time (RTT) delay in the Internet: Analysis and trends,” *IEEE Netw.*, vol. 38, no. 2, pp. 280–285, Mar. 2024.

[14]. V. L Narayana,(2020), “A Trust Based Efficient Blockchain Linked Routing Method for Improving Security in Mobile Ad hoc Networks”, International Journal of Safety and Security Engineering, Vol. 10, No. 4, 2020, pp. 509–516.

[15]. S. Allmeier and N. Gast, “Mean field and refined mean field approximations for heterogeneous systems: It works!,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 1, pp. 1–43, 2022.

[16]. A. Detti, L. Funari, and L. Petrucci, “µBench: An open-source factory of benchmark microservice applications,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 3, pp. 968–980, Mar. 2023.

[17]. D. Hurtado-Lange and S. T. Maguluri, “Throughput and delay optimality of power-of-d choices in inhomogeneous load balancing systems,” *Oper. Res. Lett.*, vol. 49, no. 4, pp. 616–622, 2021.

[18]. C.R.Bharathi, Vejendla. Lakshman Narayana , L.V. Ramesh, (2020),”Secure Data Communication Using Internet of Things”, International Journal of Scientific & Technology Research, Volume 9, Issue 04,pp:3516-3520.

[19]. X. Zhou, N. Shroff, and A. Wierman, “Asymptotically optimal load balancing in large-scale heterogeneous systems with multiple dispatchers,” *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 48, no. 3, pp. 57–58, 2021.

[20]. G. Goren, S. Vargaftik, and Y. Moses, “Distributed dispatching in the parallel server model,” *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1521–1534, Aug. 2023.