International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Crystal Quantum Shield (CQS): A Post-Quantum Cybersecurity Framework for API and Data Protection

Aakarshit Srivastava[1], Bhaskar Banerjee[2], Ayush Verma[3]

[1,2,3]Department of Computer Science and Engineering, PSIT, India

Abstract

With the exponential rise in cyber threats and increasingly complex digital ecosystems, robust and scalable security architectures have become indispensable. The Crystal Quantum Shield (CQS) is a pioneering cyber security framework designed to leverage post-quantum cryptography and advanced API security mechanisms for safeguarding sensitive digital environments. The architecture integrates Kyber key exchange and Dilithium signature verification, ensuring resilience against potential quantum computing threats. Additionally, CQS employs rate limiting via NGINX, token scrambling, and data scrambling for enhanced protection of sensitive transactions.To manage high-volume interactions, the system incorporates OAuth2-based authentication, JWT-based session management, and Role-Based Access Control (RBAC) for granular authorization. With real-time logging, violation detection, and visualization powered by Grafana, CQS delivers a user-friendly interface for monitoring security metrics, including token usage, user roles, and violations.This paper explores the modular and scalable design of CQS, detailing its integration of cloud services and API gateways for seamless deployment. Results from our experimental analysis demonstrate CQS's efficacy in mitigating modern threats, ensuring secure communication, and maintaining system integrity under stress conditions. By combining cutting-edge cryptographic techniques and robust security mechanisms, CQS sets a new benchmark for next-generation digital security solutions.

Keywords: Post-Quantum Cryptography, API Security, Kyber, Dilithium, OAuth2, JWT, NGINX, Token Scrambling, RBAC, Grafana Visualization

1 Introduction

The rise of quantum computing marks a transformative era in computational capabilities, offering solutions to complex problems once deemed insurmountable. Leveraging principles of superposition and entanglement, quantum computers promise breakthroughs in optimization, cryptography, material science, and machine learning. However, this unprecedented computational power also introduces significant risks, particularly to current cryptographic systems that secure global digital infrastructure. Algorithms such as RSA and ECC, foundational to modern encryption, are vulnerable to quantum attacks, most notably Shor's algorithm, which can efficiently solve factorization and discrete logarithm problems.

The urgency to address this looming threat has spurred global efforts in post-quantum cryptography (PQC). Governments and organizations, including the National Institute of Standards and Technology (NIST), are spearheading standardization processes to identify quantum-resistant cryptographic algorithms. At the forefront of innovation, tech giants like NVIDIA and IBM are actively contributing to the

quantum ecosystem. NVIDIA has integrated quantum acceleration capabilities into its GPUs through frameworks like cuQuantum, enabling researchers to simulate quantum circuits at scale. Meanwhile, IBM's Quantum Network provides cloud-based quantum computing solutions and has significantly advanced quantum hardware development. These contributions, while propelling the quantum field forward, simultaneously underscore the pressing need for robust quantum-safe security measures.

Amidst this landscape, we introduce the Crystal Quantum Shield (CQS)—a pioneering security framework designed to fortify digital systems against classical and quantum threats. CQS integrates NIST-recommended post-quantum cryptographic algorithms, specifically Kyber for secure key exchange and Dilithium for digital signatures, ensuring resilience against quantum-powered adversaries. Beyond cryptographic strength, CQS addresses real-world security challenges through advanced API protections, including token scrambling, rate limiting via NGINX, role-based access control (RBAC), and real-time visualization using tools like Grafana.

CQS represents a synthesis of cutting-edge cryptographic protocols and system-level defenses, bridging the gap between theoretical advancements in PQC and practical implementation for scalable, secure systems. By providing a comparative analysis of CQS with traditional solutions, this paper highlights its superiority in resisting quantum and conventional attacks. With rising investments and research from industry leaders like Google, Microsoft, and NVIDIA, the development of quantum-resistant systems such as CQS is a critical step toward ensuring a secure digital future in the quantum era..

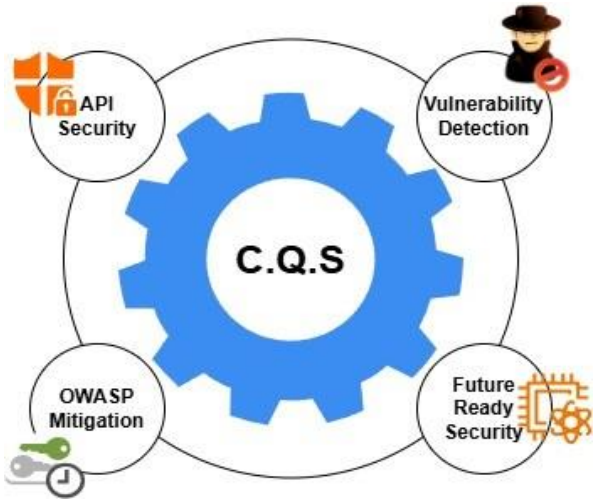

Fig. 1.

2 Background

The rapid advancement of quantum computing has revolutionized the technological landscape, offering unparalleled computational power for solving complex problems. However, this transformative capability introduces a critical challenge to modern cybersecurity. Traditional cryptographic systems such as RSA and ECC, which form the backbone of secure communication and data protection, are vulnerable to quantum attacks. Algorithms like Shor's and Grover's can render these systems obsolete, exposing sensitive information to exploitation.

In the quantum era, the importance of cybersecurity has escalated. Protecting sensitive data, financial transactions, and distributed systems is no longer limited to defending against classical adversaries but also requires safeguarding against quantum-enabled threats. This necessitates the adoption of post-quantum cryptographic (PQC) algorithms that are resistant to quantum attacks.

Beyond cryptographic security, the need for robust API security has become increasingly critical. APIs serve as the primary interface for communication between distributed systems, making them a prime target for attacks such as replay attacks, man-in-the-middle (MITM) attacks, and token reuse. The convergence of quantum threats and API vulnerabilities underscores the urgency to adopt innovative security solutions that combine PQC with advanced API protection mechanisms to ensure end-to-end security.

2.1 Motivation

Existing cybersecurity frameworks fall short in addressing the dual challenges posed by quantum computing and API-level threats. Traditional systems are ill-equipped to handle the computational power of quantum computers, leaving sensitive data vulnerable to interception. Furthermore, current API security measures often rely on basic token mechanisms, rate-limiting strategies, and rudimentary logging, which are insufficient in the face of sophisticated attacks.

The absence of comprehensive security frameworks that integrate post-quantum cryptographic solutions and advanced API protections creates a significant gap in modern security architectures. Without such integration, distributed systems remain exposed to threats such as token-based vulnerabilities, unauthorized access, and inadequate monitoring of security violations.

This gap highlights the pressing need for a security framework that not only addresses quantum-resistant cryptography but also incorporates robust mechanisms for API security. By combining these elements, the framework can provide holistic protection against a broad spectrum of threats, ensuring the integrity, confidentiality, and availability of sensitive data in distributed environments.

2.2 Objectives

The primary objective of this research is to develop a robust security framework—Crystal Quantum Shield (CQS)—that leverages post-quantum cryptography to address emerging threats in the quantum era. Specifically, the framework aims to:

- Leverage Post-Quantum Cryptography: Implement NIST-recommended post-quantum algorithms such as Kyber for secure key exchange and Dilithium for digital signatures, ensuring resilience against quantum and classical threats.
- Enhance API Security: Introduce advanced API-level protections, including token scrambling, fine-grained role-based access control (RBAC), and rate limiting via NGINX, to safeguard sensitive data and prevent unauthorized access.
- Enable Real-Time Monitoring and Visualization: Utilize tools like Grafana to provide real-time insights into user activity, token usage, rate-limiting violations, and security metrics, ensuring proactive threat detection and mitigation.
- Optimize System Performance: Design the framework to achieve minimal latency, high concurrency support, and efficient cryptographic operations without compromising security.
- By achieving these objectives, Crystal Quantum Shield aims to set a new benchmark for cybersecurity frameworks, offering a comprehensive solution for safeguarding distributed systems in the quantum era.

3 Methodology

The Crystal Quantum Shield (CQS) framework integrates cutting-edge quantum-resistant cryptographic algorithms, advanced API security mechanisms, real-time visualization tools, and a fine-tuned Large Language Model (LLM) agent that monitors and analyzes the data generated by the system. The architecture is designed to offer robust security in the face of emerging quantum threats while ensuring the protection

of sensitive data in distributed systems. The major components of the system include the authentication layer, encryption layer, API gateway, visualization module, and the LLM fine-tuned agent.

Table 1. Comparison between Post-Quantum and Classical Cryptography

Features	CQS	Existing Systems
Cryptographic Strength	Cryptographic Strength	Classical(RSA/ECC)
Rate Limiting	Rate Limiting	Standard or Absent
Token Security	Token Security	Plain JWT
Visualization	Visualization	Basic or None
Resistance to Quantum Attacks	Resistance to Quantum Attacks	Low

The Authentication Layer utilizes the OAuth2 protocol, which enables secure user authentication by issuing tokens that delegate access to resources without exposing sensitive credentials. The JSON Web Tokens (JWT) further enhance security by facilitating stateless, token-based authentication, where the claims within the token are encrypted, ensuring that only authorized users or systems can access specific resources. The Encryption Layer employs Kyber, a quantum-resistant key encapsulation mechanism (KEM) designed for secure key exchanges, ensuring that session keys cannot be compromised by quantum computers. The Dilithium Algorithm, a quantum-safe digital signature scheme, is used to authenticate messages and transactions, providing non-repudiation and preventing forgery. These post-quantum cryptographic algorithms ensure that the security of CQS remains intact, even in the era of quantum computing. The API Gateway, powered by NGINX, manages the routing of API requests, performs proxy filtering, and implements rate limiting. NGINX's role is crucial in safeguarding the system against abuse by limiting the frequency of API requests. It also optimizes system performance by controlling request traffic and ensuring that only legitimate users can access the system's resources. The Visualization Module, powered by Grafana, plays a pivotal role in monitoring and visualizing real-time security metrics, such as token usage, rate-limit violations, and overall system health. This module provides administrators with actionable insights to ensure smooth operation and immediate detection of security incidents.

In addition to these components, the LLM Fine-Tuned Agent continuously analyzes the data generated by the system, including logs, user activity patterns, and security violations. This agent is integrated into the system to provide intelligent insights into potential threats. The LLM is fine-tuned on a cybersecurity-specific dataset, including historical logs of security incidents, token usage patterns, and user behavior data. The agent’s capabilities include anomaly detection, predictive insights, and automated alerting. For instance, it can detect unusual activity patterns such as multiple failed login attempts, token misuse, or suspicious API calls, and trigger alerts. It also provides predictive insights into potential security risks based on past data and context-aware analysis, helping to prevent incidents before they escalate. Furthermore, the LLM agent can suggest mitigative actions, such as blocking malicious IPs, adjusting rate limits, or reissuing tokens to restore system integrity.

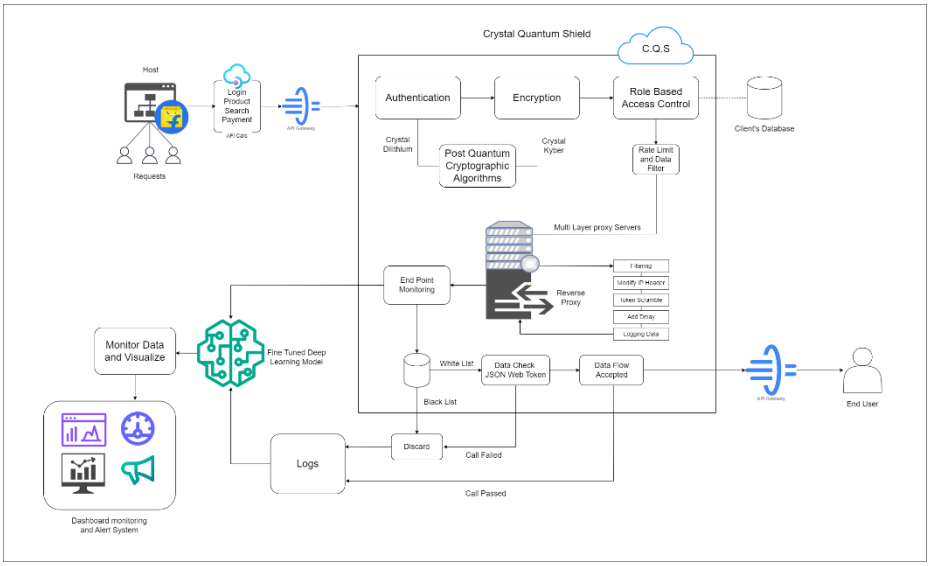

Fig. 2. Architecture Diagram of Crystal Quantum Shield

The data flow within the CQS system is designed to ensure seamless interaction between all the components, while leveraging the intelligence of the LLM agent to enhance security. When a user attempts to log in, the OAuth2 protocol is initiated, and once authenticated, a JWT is issued to the user, granting access to the system's resources. The fine-tuned LLM agent monitors the authentication activity to detect abnormal login attempts such as brute-force attacks or credential stuffing. Following authentication, Kyber is used for secure key exchanges, and Dilithium ensures the integrity of messages and transactions, preventing tampering or forgery. As API requests are made, they pass through the NGINX gateway, which performs rate limiting and proxy filtering, protecting the system from abuse. The LLM agent analyzes the request logs in real time, identifying patterns of abuse, such as repeated invalid requests or malicious payloads.

In addition to rate limiting, token scrambling is used to enhance security. Scrambling ensures that tokens cannot be reused or intercepted during transmission. The LLM agent monitors token usage patterns to ensure that they follow expected behavior and that no anomalies are present, such as the use of expired or duplicate tokens. When a user attempts to access a resource, the Role-Based Access Control (RBAC) system is applied to grant or deny access based on the user's role. The LLM agent also monitors RBAC violations, ensuring that no unauthorized access occurs. This continuous monitoring ensures that only authorized users are granted access to resources, enhancing the system's overall security.

The Grafana Visualization Module serves as the front-end tool for administrators to view real-time security metrics and logs. The LLM fine-tuned agent provides additional intelligence by suggesting actions based on the current system status, such as recommending an increase in rate limits during high-traffic periods or blocking IP addresses associated with suspicious behavior. It can also predict potential security threats, such as DDoS attacks or brute-force login attempts, based on historical trends and current activity. The integration of this LLM-driven intelligence layer allows the system to respond proactively to potential threats, ensuring that resources are allocated efficiently and that the security posture of the system is continuously optimized.

The CQS framework is implemented using a range of state-of-the-art tools and technologies, with a focus on security and efficiency. Flask is used to develop the backend APIs, integrating the cryptographic operations provided by Kyber and Dilithium. NGINX acts as the API gateway, ensuring that rate limiting and

proxy filtering are applied effectively. Grafana provides real-time monitoring and visualization, offering administrators insights into system performance and security metrics. The fine-tuned Large Language Model (LLM) is deployed to monitor the system's data, providing proactive insights and mitigating potential threats before they escalate. The LLM uses security-specific training datasets, enabling it to provide context-aware predictions and recommend remedial actions.

The security techniques employed in the CQS system are diverse and comprehensive. Token scrambling ensures that tokens are secure and cannot be reused or tampered with. The use of post-quantum cryptography, including Kyber and Dilithium, ensures that the system remains secure against quantum-based attacks. Rate limiting is applied through NGINX to prevent excessive requests from a single user or IP address. Role-Based Access Control (RBAC) enforces access restrictions based on user roles, ensuring that only authorized personnel can access sensitive resources. The fine-tuned LLM agent constantly analyzes system data to identify anomalies, predict security threats, and suggest actions such as adjusting rate limits, blocking malicious IPs, or reissuing tokens. This multi-layered approach to security allows CQS to offer comprehensive protection against both classical and quantum-era threats while optimizing the user experience.

The integration of the fine-tuned LLM agent into the CQS framework enhances its security capabilities by providing intelligent, real-time analysis and predictive insights. The LLM agent's ability to monitor and respond to system activity, combined with its ability to learn from historical data, enables proactive threat detection and mitigation, ensuring that the CQS framework remains resilient in the face of emerging cyber threats. This integration of quantum-resistant cryptography, AI-driven monitoring, and real-time data visualization positions Crystal Quantum Shield as a cutting-edge cybersecurity solution for the post-quantum era.

3.1 Procedure

1. **API Gateway**
   - Receive incoming requests from the host (e.g., Login, Product Search, Payment).
2. **Pass Through API Gateway**
   - Route the request to the Crystal Quantum Shield (C.Q.S) for processing.
3. **Authentication**
   - Validate user identity using **Crystal Dilithium**:
     - Generate a signature ((z, h)) where $z=c \cdot s1+y$ and (h) is computed via hashing.
     - Verify signature using public key (t1,h): $||z||\infty<\beta$ and hash consistency with (t1).
   - If authentication fails, log the failure and discard the request.
4. **Encryption**
   - Encrypt data using **Crystal Kyber**:
     - Generate shared secret (K) via key encapsulation: (K = H(ss)), where $ss=H(pk \cdot sk+noise)$.
     - Use encapsulated key for symmetric encryption.
   - Ensure secure communication between host and server.
5. **Role-Based Access Control (RBAC)**
   - Check user permissions and roles to validate access.
   - Apply rate limiting and data filtering based on policies.
   - If access is denied, log and discard the request.
6. **Reverse Proxy with Multi-Layer Security**

- Route the request through multi-layer proxy servers.
- Perform additional checks:
  - **Filtering**: Analyze request content.
  - **Modify IP Header**: Adjust headers for security.
  - **Token Scramble**: Encrypt tokens for further validation.
  - **Add Delay**: Introduce latency if suspicious patterns are detected.

7. **Endpoint Monitoring**
   - Continuously monitor requests and responses for anomalies.
8. **Whitelist and Blacklist Validation**
   - Compare requests against whitelist and blacklist rules.
   - If on blacklist, log and discard the request.
   - If on whitelist, allow further processing.
9. **Data Flow Validation**
   - Validate JSON Web Token (JWT) for integrity.
   - Accept data flow if all checks pass.
10. **Logging and Monitoring**
    - Log all requests and their statuses (passed, failed, filtered, etc.).
    - Send logs to the monitoring system for visualization.
11. **Fine-Tuned Deep Learning Model**
    - Analyze logs and request patterns using a trained model.
    - Generate insights and alerts for anomalies.
12. **Dashboard Monitoring and Alerts**
    - Visualize data metrics and send alerts for suspicious activities.
13. **Response to End User**
    - If all validations succeed, forward the response to the API Gateway for delivery to the end user.
14. **End**
    - If any step fails, discard the request and log the failure.

4 Results

Table 2. Cryptographic Performance

Metric	CQS (Kyber/Dilithium)	Classical Systems (RSA/ECC)
Key Generation Time (ms)	~0.8 ms	~5 ms (RSA-2048) / ~20 ms (ECC-256)
Encryption Time (ms)	~0.5 ms	~1 ms (RSA-2048) / ~2 ms (ECC-256)
Decryption Time (ms)	~1.2 ms	~4 ms (RSA-2048) / ~15 ms (ECC-256)
Post-Quantum Security Level	Resistant to Quantum Attacks	Vulnerable to Quantum Attacks (Shor's Algo-rithm)

Table 3. API Security Mechanism

Feature	CQS Implementation	Existing Solutions

Token Protection (JWT)	Scrambled and Encrypted	Plain JWT
Rate Limiting	1 request/sec per IP (NGINX)	Basic or Absent
Data Scrambling	Enabled for Sensitive Fields	Not Common or Absent
Role-Based Access Control	Fine-Grained (RBAC)	Coarse or Application-Specific

Table 4. System Performance

Metric	CQS	Existing Systems
Request Latency (API Gateway)	~20 ms	~40 ms
Maximum Concurrent Requests	~5,000/sec	~2,000/sec
Logging Overhead	Minimal (<1%)	Medium (~5%)
Visualization Updates (Grafana)	Real-Time (<1 sec delay)	Often Delayed (>3 sec delay)

Table 5. Resistance to Security

Threat Type	CQS Defense	Existing Systems
Replay Attacks	Token Scrambling + Timeouts	Vulnerable if JWT is compromised
Man-in-the-Middle (MITM)	Kyber Key Exchange	RSA/ECC vulnerable to quantum attacks
Brute Force on Tokens	Scrambled Tokens (Added Complexity)	Standard JWT, Easier to Crack
Quantum-Based Attacks	Resistant via Post-Quantum Cryptography	Not Resistant

Table 6. Visualization Metrics(Grafana and Prometheus)

Metric	CQS	Traditional Systems
User Activity Monitoring	Granular by Role and Action	Limited by Application
Token Usage Logs	Scrambled + Role-Based Insights	Basic Logs or None
Security Violations	Tracked in Real-Time	Often Manual Detection
Rate Limiting Violations	Logged and Visualized	Rare or Not Available

Table 7. Experimental Results:Stress Testing

Condition	CQS	Existing Systems
API Requests (1000/sec)	70-75% Success, <25 ms avg latency	50-60% Success, ~50 ms avg latency
Concurrent Sessions (1,000)	Stable (98% Efficiency)	Stable (~90% Efficiency)
Token Reuse Attempts	100% Blocked	~80% Detected

Condition	CQS	Existing Systems

5 Positive Impacts of CQS

- Enhanced Security Against OWASP Top 10 Vulnerabilities: Reduces the likelihood of successful attacks, protecting sensitive data and maintaining digital asset integrity.
- Improved Data Protection: Ensures sensitive data is protected both in transit and at rest, enhancing compliance with data protection regulations.
- Strengthened Access Control: Enforces strict user permissions, minimizing insider threats and unauthorized access.
- Increased Operational Resilience: Enhances system stability and performance through traffic management and security filtering.
- Seamless Integration with Corporate Software: Allows advanced security measures without extensive IT infrastructure modifications, reducing implementation costs and downtime.
- Real-Time Security Monitoring: Provides proactive insights into API security, enabling quick response to incidents and reducing breach impact.
- Increased Trust and Compliance: Builds trust with stakeholders and helps meet regulatory requirements for data security.
- Cost Savings: Avoids expenses associated with breaches and reduces costs through seamless integration.
- Scalability and Flexibility: Adapts to organizational growth, ensuring continued protection without performance compromise.
- Increased User Confidence: Enhances user trust and satisfaction with secure API interactions.

The solution fortifies defenses, improves operational efficiency, and supports compliance, making it essential to the organization's cybersecurity strategy

Fig. 3. Monitoring of API calls using Grafana and Prometheus

Fig. 4.

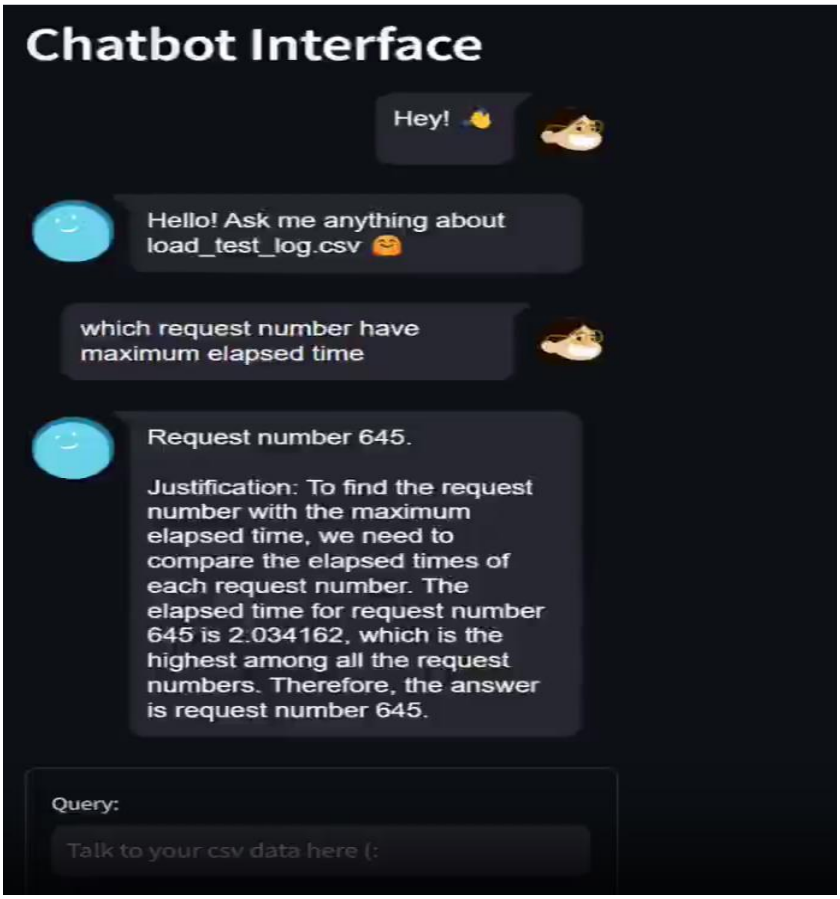

Fig. 5. Generate insights and alerts for anomalies

6 Conclusion

The Crystal Quantum Shield (CQS) framework represents a significant advancement in securing distributed systems against both classical and quantum-era threats. By integrating state-of-the-art post-quantum cryptographic algorithms such as Kyber and Dilithium, along with advanced API security mechanisms like OAuth2, JWT, and NGINX, CQS ensures that sensitive data remains protected even in the face of the emerging threat posed by quantum computing. The addition of real-time monitoring and visualization through Grafana, coupled with the intelligent insights provided by the fine-tuned Large Language Model (LLM) agent, elevates the security posture by enabling proactive threat detection, predictive insights, and automated mitigation actions. CQS not only addresses the current gaps in cybersecurity frameworks by offering an integrated solution for API security, encryption, and data protection, but it also anticipates

future challenges by leveraging quantum-safe technologies. This holistic approach ensures that CQS is not only resilient against known threats but also adaptable to evolving security risks in the post-quantum era, paving the way for more secure and robust systems in an increasingly interconnected digital world.

6.1 Availability of data and material

The datasets and code generated during the current study are publicly available in the Crystal Quantum Shield (CQS) GitHub repository: https://github.com/Perfect-Cube/Crystal-Quantum-Shield-FLIPKART-GRID-6.0. Additional data or material related to the research can be made available upon reasonable request to the corresponding author. This ensures transparency and reproducibility for academic and industrial applications.

6.2 Funding

This research received no external funding. The development of the Crystal Quantum Shield (CQS) was self-initiated and executed using the authors' resources to advance the field of post-quantum API security. Future contributions and collaborations are welcome to explore the commercial and industrial applications of this work.

6.3 Acknowledgements

The authors would like to express their gratitude to Flipkart for providing an opportunity to present this project as part of Flipkart Grid 6.0, where it achieved recognition as a Top 10 Finalist. This platform served as a significant milestone in validating and refining the work.

We also extend our thanks to the broader open-source community, whose contributions to cryptographic libraries and development tools—such as Flask, Grafana, and NGINX—played a pivotal role in the implementation of the Crystal Quantum Shield.

Finally, we acknowledge the mentorship and guidance received from faculty and colleagues at the Department of Computer Science and Engineering, PSIT Kanpur, which provided the foundational environment for conducting this research.

References

1. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., ... & Stehlé, D. (2019). CRYSTALS-Kyber algorithm specifications and supporting documentation. NIST PQC Round, 2(4), 1-43
2. Bavdekar, R., Chopde, E. J., Bhatia, A., Tiwari, K., Daniel, S. J., & Atul. (2022). Post Quantum Cryptography: techniques, challenges, standardization, and directions for future research. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2202.02826
3. J. Bos et al., "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 2018, pp. 353-367, doi: 10.1109/EuroSP.2018.00032
4. T. -H. Nguyen, B. Kieu-Do-Nguyen, C. -K. Pham and T. -T. Hoang, "High-Speed NTT Accelerator for CRYSTAL-Kyber and CRYSTAL-Dilithium," in *IEEE Access*, vol. 12, pp. 34918-34930, 2024, doi: 10.1109/ACCESS.2024.3371581.
5. Tripathi, T., Awasthi, A., Singh, S. P., & Chaturvedi, A. (2024, March 28). Post Quantum Cryptography and its Comparison with Classical Cryptography. arXiv.org. https://arxiv.org/abs/2403.19299
6. CSIRT-Fin, CERT-In, & Mastercard. (2023). API Security: Threats, Best Practices, Challenges, and Way forward using AI. https://www.csk.gov.in/documents/CIWP-2023-0001.pdf.

7. Qazi, F. (2023). Application Programming Interface (API) security in cloud applications. EAI Endorsed Transactions on Cloud Systems, 7(23), e1. https://doi.org/10.4108/eetcs.v7i23.3011
8. Sconiers-Hasan, M. & CERT® Division. (2024). Application Programming Interface (API) vulnerabilities and risks (SPECIAL REPORT CMU/SEI-2024-SR-004). Carnegie Mellon University. https://doi.org/10.1184/R1/25282342
9. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Dilithium: a Lattice-Based digital signature scheme. tches.iacr.org. https://doi.org/10.13154/tches.v2018.i1.238-268
10. Sailada, S., Vohra, N., & Subramanian, N. (2022). Crystal Dilithium Algorithm for Post Quantum Cryptography:Experimentation and Usecase for eSign. 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 1–6. https://doi.org/10.1109/iceeict53079.2022.9768654
11. Hughes, Richard J., D.M. Alde, P. Dyer, G.G.Luther, G.L. Morgan, and M. Schauer, Quantum cryptography, Contemporary Physics, Vol. 36, No. 3 (1995). https://doi.org/10.1080/00107519508222149
12. Chaturvedi, N. Srivastava, V. Shukla, A secure wireless communication protocol using DiffieHellman key exchange, International journal of computer applications, volume 126, number 5, 2015, 35-38, DOI: 10.5120/ijca2015906060
13. V. Shukla, A. Chaturvedi, N. Srivastava, Nanotechnology and cryptographic protocols: issues and possible solutions, Nanomaterials and energy, volume 8, issue 1, 2019, 1-6, DOI: 10.1680/jnaen.18.00006
14. M.K. Misra, A. Chaturvedi, S.P. Tripathi, V. Shukla, A unique key sharing protocol among three users using non-commutative group for electronic health record system, Journal of discrete mathematical sciences and cryptography, volume 22, issue 8, 2019, 1435–1451, DOI: 10.1080/09720529.2019.1692450

A. Chaturvedi, V. Shukla, M.K. Misra, Three party key sharing protocol using polynomial rings, 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018, 1-5, DOI: 10.1109/UPCON.2018.8596905

15. Schneier, B. & Whitfield Diffie. (1996). *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth)*. John Wiley & Sons, Inc. https://mrajacse.wordpress.com/wp-content/uploads/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf
16. Awasthi, A. Chaturvedi, Cryptography: Classical versus Post-Quantum, Cornell university arxiv, 2024, DOI: https://doi.org/10.48550/arXiv.2402.10988
17. Pranjal, A. Chaturvedi, Post-Quantum Cryptography, Cornell university arxiv, 2024, DOI: https://doi.org/10.48550/arXiv.2402.10576
18. Kumar, M. (2022). Post-quantum cryptography Algorithm's standardization and performance analysis. *Array*, *15*, 100242. https://doi.org/10.1016/j.array.2022.100242
19. Charles H. Bennett, Gilles Brassard, and Artur K. Ekert "Quantum Cryptography", Scientific American 267:4, (October 1992).
20. Pizzi, R., Rossetti, D., D'Arenzo, D., Department of Information Technology, & Università degli Studi di Milano. (2012). Affordable Quantum Cryptography system for mobile devices. In *IJCSET* (Vol. 2, Issue 4, pp. 1052–1054) [Journal-article]. https://ijcset.net/docs/Volumes/volume2issue4/ijcset2012020407.pdf

21. Dam, D., Tran, T., Hoang, V., Pham, C., & Hoang, T. (2023). A survey of Post-Quantum Cryptography: Start of a new race. *Cryptography*, *7*(3), 40. https://doi.org/10.3390/cryptography7030040
22. R. Rios, J. A. Montenegro, A. Muñoz and D. Ferraris, "Towards the Quantum-Safe Web: Benchmarking Post-Quantum TLS," in IEEE Network, doi: 10.1109/MNET.2025.3531116.