Mesopotamian journal of Cybersecurity
Vol.5,No.2, **pp**. 361–374
DOI: https://doi.org/10.58496/MJCS/2025/023; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Synergizing Quantum-Safe Signatures with JWT for Unparalleled Security in Web Applications

Kalpana B N [1,*], R Saravana Kumar [2]

[1] *Research Scholar, Dayananda Sagar Academy of Technology and Management, Visvesvaraya Technological University, Belagavi, 590018, India.*

[2] *Research Supervisor, Dayananda Sagar Academy of Technology and Management, Visvesvaraya Technological University, Belagavi, 590018, India.*

ARTICLEINFO

Article history

Received 06 Nov 2024
Revised 09 Jan 2025
Accepted 07 Mar 2025
Published 26 May 2025

Keywords

Authentication

Authorization

JWT

Quantum-safe Signatures

Hybrid Signatures

ABSTRACT

The surge in internet use has made authentication and authorization essential for protecting users' privacy and security in web applications. JSON Web Token (JWT), a token-based authentication mechanism, stands out as a desirable choice for its scalability, ease of use, and interoperability. However, existing JWT signing algorithms, which rely on mathematical problems such as factoring large integers and discrete logarithms, are vulnerable to quantum computing breakthroughs, which poses significant security risks. Addressing this challenge requires evaluating quantum-safe alternatives for JWT authentication. While prior research has focused on limited sets of post-quantum algorithms, a comprehensive evaluation of all standardized algorithms remains unexplored. This study presents the first such evaluation within the JWT authentication framework, analysing algorithms recommended by the National Institute of Standards and Technology (NIST), including Falcon, SPHINCS+, and Dilithium, and their hybrid counterparts. We compare their performance against traditional algorithms such as RS256, ES256, PS256, and HS256. Our experimental results reveal that Falcon is the most efficient quantum-safe algorithm, with a token generation time of 18.68 ms and verification time of 0.65 ms, whereas SuperFalcon outperforms hybrid algorithms, with generation and verification times of 1.19 ms and 1.81 ms, respectively. These findings establish a foundation for transitioning JWT systems to quantum-safe cryptographic standards.

1. INTRODUCTION

In today's digital landscape, web applications play a crucial role in our lives by providing easy access to a wide range of services online. With the increasing use of web-based services, it is important to safeguard user data and ensure smooth interactions. On the basis of the Top Ten 2021 list from the Open Web Application Security Project (OWASP) [1], broken access control is identified as the most serious security risk facing web applications. Therefore, it is crucial to implement strong authentication and authorization mechanisms. Session-based authentication methods [2], while commonly used, are prone to vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and session fixation attacks, as well as scalability issues, owing to their stateful nature. JWT, a stateless alternative, [3, 4] addresses these limitations while offering ease of integration in modern architectures.

JWTs rely on cryptographic algorithms such as the Rivest–Shamir–Adleman (RSA), elliptic curve cryptography (ECC), hash-based message authentication (HMAC), and elliptic curve digital signature algorithm (ECDSA) for signing and verification. While these methods are secure against classical attacks, these algorithms are highly vulnerable to quantum algorithms such as Shor's [5], which factorizes large numbers, and Grover's [6], which accelerates unsorted database searches. As JWTs play a critical role in modern web authentication, addressing their vulnerabilities against quantum threats is vital to ensuring secure online interactions. Adversaries could exploit these vulnerabilities through quantum attacks, leading to compromised data integrity and confidentiality. Moreover, encrypted communications today a-re already at risk from "harvest now, decrypt later" [7, 8] attacks, where malicious actors intercept and store encrypted data to decrypt it once quantum decryption capabilities become available. Cryptographically relevant quantum computers (CRQCs), expected within the next decade, could break current encryption standards, creating an urgent need to transition to quantum-safe cryptography.

Although there are post-quantum cryptography systems that are lightweight and suitable for Internet of Things (IoT) devices with limited resources, [9, 10] web applications typically run-on servers with sufficient computational resources.

*Corresponding author. Email: bnkalpana12@gmail.com

Therefore, this study focuses on standardized post-quantum algorithms that prioritize security strength and quantum resistance for JWT implementations rather than resource optimization.

Securing JWT-based systems against quantum threats [11] is crucial, especially given that current implementations largely rely on classical cryptography, making them vulnerable to future quantum attacks. Despite numerous proposals to include post-quantum algorithms in domains such as blockchain, automotive systems, and wireless networks, JWT implementations have received limited attention[56]. Existing studies often evaluate only one or two post-quantum algorithms, overlooking comprehensive assessments of their feasibility and performance with JWT. This research bridges this gap by systematically evaluating multiple post-quantum and hybrid algorithms, focusing on practical challenges such as signature size, processing overhead, and key size.

Among the algorithms [12] considered are Falcon, Dilithium, and SPHINCS+, which are part of NIST's standardized post-quantum cryptographic suite. These algorithms, which are known for their ability to withstand quantum attacks, are based on challenging mathematical issues such as hash-based and lattice-based cryptography, which are impractical for both classical and quantum computers to resolve. Hybrid algorithms that combine quantum-safe and conventional approaches are also explored to further enhance security[57]y. By analysing key performance metrics such as the token generation time, verification time, and payload size, this work aims to balance robust security with computational efficiency, providing practical solutions for JWT implementations in the quantum era.

The primary objective of our work is to assess the performance of post-quantum signature algorithms with JWT. We implement and test three standardized post-quantum cryptographic algorithms, along with three hybrid algorithms, in combination with the JWT. The findings from our study corroborate that, from a performance perspective, there exist highly promising algorithms, Falcon and SuperFalcon, which can be used in practical applications. Furthermore, this work could act as a comprehensive guide for researchers and developers aiming for ways to improve the security of web-based applications. To summarize, the major contributions of this work are as follows:

- A fully functional web application prototype was developed to demonstrate the practical feasibility of integrating quantum-safe signatures into JWTs for real-world authentication scenarios.
- Evaluation of the feasibility of integrating NIST standardized quantum-safe signature algorithms, namely, Falcon-1024, Dilithium5, and SPHINCS+, and hybrid signature algorithms, such as Super-Falcon, SuperSphincs, and SuperDilithium, with the JWT signing process.
- A thorough performance analysis of quantum-safe and hybrid signatures with JWT is conducted, evaluating critical metrics such as the token generation time, token verification time, response time, request header size, and response body size.
- We compare quantum-safe and hybrid signatures against traditional JWT signatures (RS256, ES256, PS256, and HS256), highlighting the trade-offs in terms of security and computational efficiency.

The remainder of this paper is organized as follows: Section 2 provides background information. Section 3 reviews related work carried out thus far. Section 4 describes the performance analysis and results. Section 5 presents a comparative analysis and consolidates the results obtained. Section 6 discusses the limitations. Section 7 provides a conclusion and outlines potential future improvements.

2. BACKGROUND

2.1 JWT Overview

JWT [13] is an open, stateless, industry-standard method for securely transmitting information between a client and a server. Owing to its stateless nature and reduced server load, JWT is apt to maintain a form of user session state across multiple requests. The JWT structure consists of three parts: a header, payload, and signature separated by dots, as shown in Fig. 1.

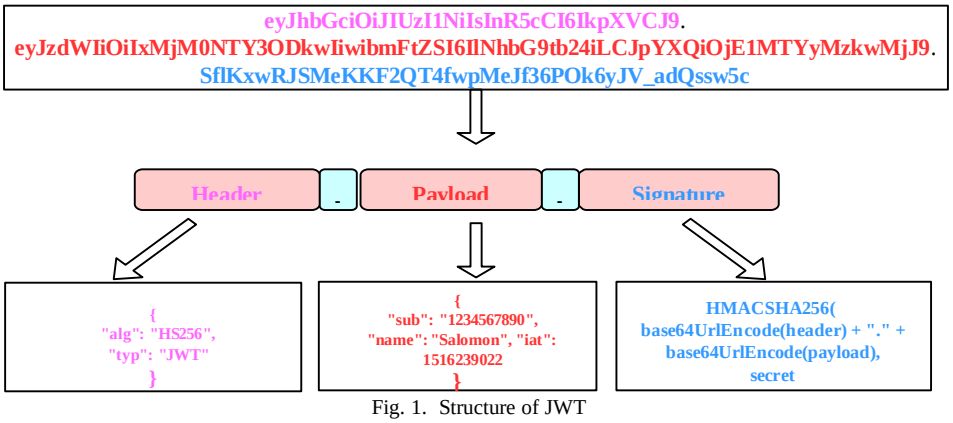


Fig. 1. Structure of JWT

This credential is used to grant access to resources. The header contains 2 parts: the type and algorithm being used. The payload contains claims about the user, and both the header and payload are encoded in the base64. The signature is created via an encoded header, an encoded payload, a secret, and the algorithm mentioned in the header. If an attacker tries to tamper with the data, the signature verification fails. Since the contents of the token are base64 encoded and not encrypted, it should only be used to transmit non-sensitive information within the claim or token payload. The basic steps involved in the JWT mechanism are as follows:

1. The client authenticates by sending a username and password to the server.
2. The server verifies the credentials, generates a token, and sends it back to the user.
3. The client stores this token and uses for subsequent interaction with the server. The overall process involved in JWT authentication is shown in Fig. 2.

2.2. Classical Cryptographic Signatures

Cryptography is the practice of securing information by transforming it into a format that only authorized parties can understand and use. It has evolved significantly and encompasses two primary categories: symmetric and asymmetric cryptography.

- Symmetric cryptography, also known as secret-key cryptography, uses a single shared key for both encryption and decryption. This method, exemplified by algorithms such as the advanced encryption standard (AES), offers fast processing but faces challenges in secure key distribution.
- Asymmetric cryptography, or public-key cryptography, addresses the limitation of key distribution by using a pair of mathematically related keys—a public key for encryption and a private key for decryption. The RSA and ECC are prominent examples of asymmetric algorithms.

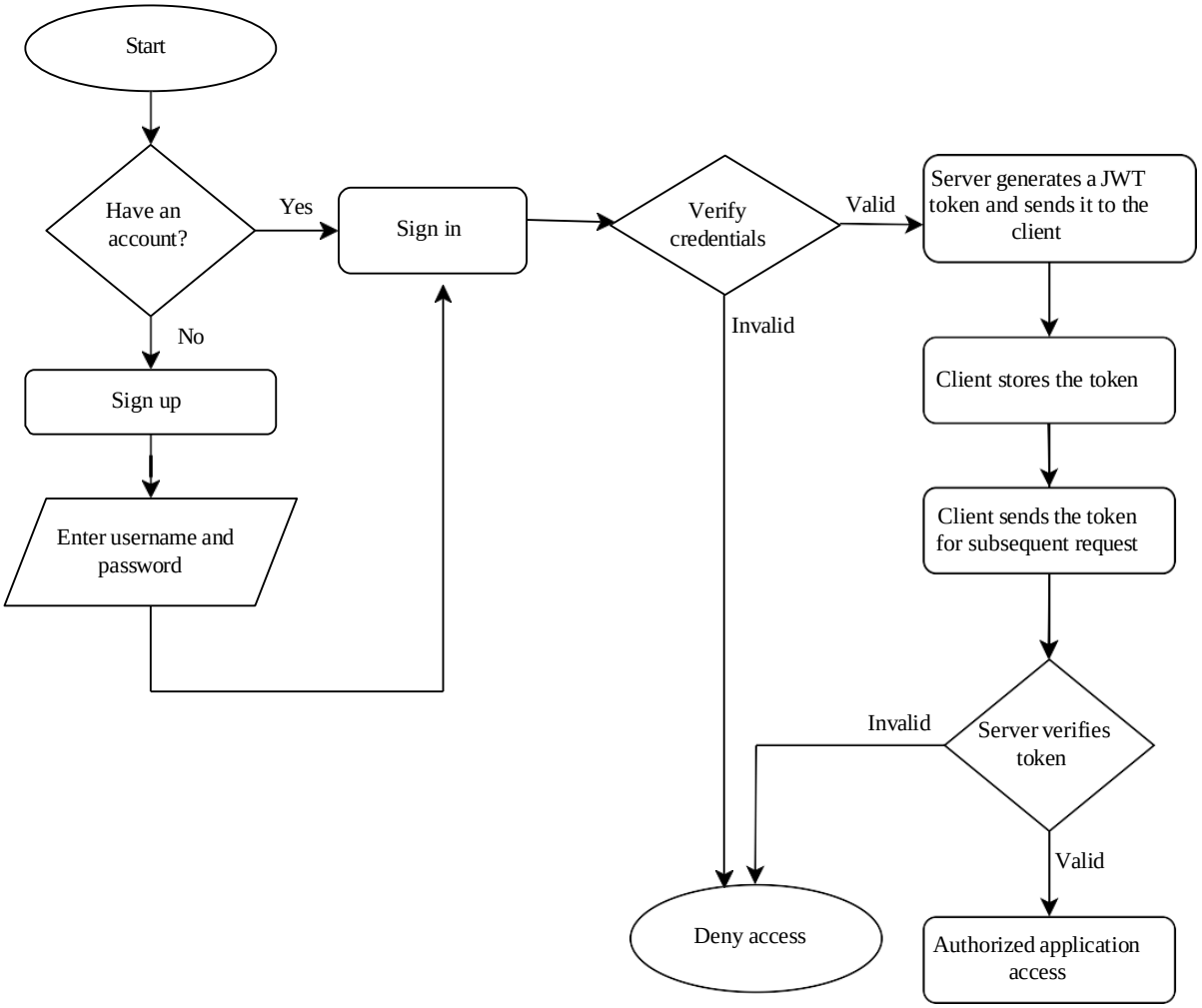


Fig. 2. JWT authentication flow

Building upon these foundations, digital signatures emerged as a critical application of asymmetric cryptography. Digital signatures provide a means to verify the authenticity and integrity of digital messages or documents, playing a crucial role in non-repudiation and secure communication in our increasingly digital world. Conventional cryptographic systems rely on the computational challenges posed by specific mathematical problems. These problems are designed to be difficult to solve, even with significant computing power, thereby forming the foundation of the security these systems provide. The classical signature algorithms [14, 15] commonly used in JWTs include the following:

1. The RSA signature with SHA-256 (RS256) uses the RSA asymmetric encryption algorithm with the secure hash algorithm 256 (SHA-256) hashing algorithm for digital signature generation and verification.
2. The ECDSA Signature Algorithm with SHA-256 (ES256) uses the ECDSA with the SHA-256 hashing algorithm for digital signature operations based on ECC.
3. The probabilistic signature scheme with SHA-256 (PS256) incorporates probabilistic padding with the SHA-256 hashing algorithm for RSA-based digital signature operations.
4. HMAC with SHA-256 (HS256) employs the HMAC construction with the SHA-256 hashing algorithm for symmetric key-based digital signature generation and verification.

2.3. Post-quantum Cryptographic Signatures

In the realm of cryptography, post-quantum cryptography (PQC) is also known as quantum-safe cryptography. The main goal of PQC [16] is to build a cryptographic system that is resistant to both classical and quantum computers and is compatible with existing communication protocols and networks. The most popular strategies for post-quantum algorithms [17, 18] include the following:

- Lattice-based cryptography is based on the computational difficulty of solving problems in lattice theory, such as the shortest vector problem (SVP) and the closest vector problem (CVP).
- Code-based cryptography is based on the computational hardness of decoding random linear codes.
- Hash-based cryptography is based on the security of hash functions and their properties, such as collision resistance and preimage resistance.
- Multivariate-based cryptography is based on the computational difficulty of solving systems of multivariate polynomial equations over finite fields.
- Isogeny-based cryptography is based on the computational difficulty of finding isogenies between elliptic curves.

The adoption of post-quantum cryptography is driven by concerns about the potential development of a CRQC, a device theoretically capable of breaking current encryption methods. Theoretically, breaking a 160-bit elliptic curve cryptographic key would require a quantum computer with 1,000 logical qubits, whereas factoring a 1,024-bit RSA modulus would necessitate a quantum computer with 2,000 logical qubits [19]. Hence, to proactively circumvent this issue in 2016, NIST initiated a process to find the best quantum-safe schemes with the aim of establishing new cryptographic standards. Six years later, out of 69 submissions and three rounds of evaluation, NIST selected four quantum-resistant algorithms for standardization: Cryptographic Suite for Algebraic Lattices (CRYSTALS) Kyber for public-key encryption and CRYSTALS-Dilithium, Falcon, and SPHINCS+ for digital signatures.

2.3.1. Falcon

Fast-Fourier lattice-based compact signatures over NTRU (FALCON) is a lattice-based signature scheme specifically created for effective and compact execution over N-th degree truncated polynomial ring unit (NTRU) lattices. Falcon's architectural simplicity stems from its use of the theoretical design for lattice-based hash-and-sign algorithms proposed by Gentry, Peikert, and Vaikuntanathan (GPV) [20] in their 2008 study. The two main elements required by this theoretical model are NTRU lattices and trapdoor sampling. The integration of the GPV framework, NTRU lattices, and fast Fourier sampling creates Falcon's signature system.

2.3.2. Dilithium

Dilithium is a lattice-based digital signature scheme renowned for its resilience against chosen message attacks. It leverages the intricacies of lattice problems across module lattices [21, 22]. Its design draws upon Lyubashevsky's "Fiat-Shamir with Aborts" approach, ensuring both compactness and security in lattice-based Fiat-Shamir schemes through the utilization of rejection sampling. Employing uniform sampling, Dilithium has the smallest public key signature size among lattice-based systems reliant on a uniform distribution. This cryptographic scheme achieved significant recognition by advancing as one of the leading contenders in prestigious NIST competition.

2.3.3. SPHINCS+

SPHINCS is a stateless hash-based signature scheme that employs HORST, a few-time (hash-based) signature scheme. HORST represents an enhancement over the previous few-time signature scheme, hashing to obtain a random subset (HORS). SPHINCS+ [23, 24] is an enhancement of SPHINCS and includes multitarget attack protection, tree-less

Winternitz one-time signature (WOTS+) public key compression, forest of random subsets (FORS), and verifiable index selection. The flexibility of using various hash functions, enabling customization on the basis of specific security needs, is a key benefit of this scheme. However, a notable drawback is the considerable size of the resulting signatures. Table 1 outlines the sizes of the bytes corresponding to the public key, secret key, and signature for the post-quantum standardized algorithms. The data reveal that SPHINCS+ has the largest signature size, whereas Falcon has a more compact signature size.

TABLE I. POST-QUANTUM ALGORITHM SIGNATURE AND KEY SIZES IN BYTES

Algorithm	Public Key	Private Key	Signature
Dilithium5	2592	4864	4595
SPHINCS+-256 s	64	128	29792
Falcon1024	1793	2305	1280

2.4. Hybrid Signatures

A hybrid signature scheme combines a classical signature algorithm with a quantum-safe signature algorithm. Super-SPHINCS, SuperFalcon, and SuperDilithium merge post-quantum SPHINCS+, Falcon, and Dilithium, respectively, with the conventional elliptic-curve Ed25519 into single-screening schemes. Ed25519 is an ECC-based cryptographic algorithm used for digital signature generation and verification.

2.5. Light Weight Cryptography

Lightweight cryptography refers to cryptographic algorithms and protocols that are designed to function well in settings with few resources. The processing power, memory, energy, and bandwidth constraints of embedded systems, IoT devices, and mobile devices are common features of these contexts. Recent work by Hazzaa et al. [25] introduced a new lightweight cryptographic method for IoT applications in smart city environments, demonstrating a 33% reduction in energy consumption and execution time compared with AES through a dual S-box approach in the SubByte process. Similarly, Abbas et al. [26] proposed a lightweight encryption system for IoT devices that uses DNA-based keys to enhance data protection through a strong avalanche effect. Our work focuses on quantum-safe algorithms suitable for web applications where computational resources are less constrained.

3. LITERATURE REVIEW

This section provides an overview of research on JWT authentication mechanisms using both classical and post-quantum signature algorithms. The first part shown in Table 2 reviews existing works on JWT authentication mechanisms, whereas the second part examines the utilization of post-quantum signature algorithms in various applications.

In the pursuit of transitioning to quantum-safe cryptography, several research efforts have been undertaken to analyse and evaluate the performance, security, and feasibility of various post-quantum cryptographic algorithms. Raavi et al. [43] examined the security of various post-quantum signature algorithms by considering the depth–width cost for quantum circuits to measure the security strengths via the Universal Quantum Gate Model and Quantum Annealing. The performance analyses included computational loads of the algorithms during execution, together with the implementation overheads and communication costs, when integrated with the transmission control protocol (TCP)/Internet protocol (IP) and transport layer security (TLS).

On the feasibility front, Tan et al. [44] conducted a detailed analysis of the obstacles associated with adopting post-quantum digital signatures in 14 real-world applications across diverse sectors, including finance, internet, critical infrastructure, and corporate sectors. The signing requirements of the applications were assessed and mapped against each of the six PQC standardization round 3 methods in a feasibility matrix. Similarly, Ye Yuan et al. [45] implemented and tested the performance of various post-quantum cryptography algorithms on JavaScript-enabled platforms. The results demonstrated the feasibility of deploying lattice-based cryptography on JavaScript-capable devices while simultaneously providing portability. Ghinea et al. [46] introduced a practical way of upgrading hardware security keys to a hybrid digital signature scheme that combines two components: a classically secure scheme, ECDSA, and a post-quantum secure scheme, Dilithium. To demonstrate the feasibility of the scheme, the implementation was carried out with open-source security-key firmware, and performance was measured.

With respect to performance evaluation in TLS, Tzinos et al. [47] conducted three experiments to assess the performance of the post-quantum secure algorithms Kyber, Saber, and NTRU in TLS. The experimental results showed that the Kyber method performed best in the handshake phase of the TLS for key exchange. In the context of the Internet of Things

(IoT), Ashraaf [48] analysed various post-quantum cryptographic algorithms to identify a set of algorithms suitable for IoT devices. These findings suggest that the use of Kyber-512 is the best choice.
Hekkala et al. [49] focused on challenges related to the integration of four lattice-based algorithms, such as Kyber, Dilithium, Saber, and Frodo, in the C++ crypto library, Crypto++, to benefit developers. In comparative studies, Henry et al. [50] introduced a variant of the McEliece cryptosystem to secure access control data and a variant of the NTRU cryptosystem to secure cloud user data. The study indicated that adopting the proposed variants enhanced security, but time constraints remain a concern. Gan and Yokubov [51] conducted performance analyses of the ECDSA in comparison with lattice-based quantum-safe algorithms, namely, Falcon, Dilithium, and multivariate-based Rainbow, in the context of blockchain usage. The outcome revealed the need to refine key sizes and efficiently implement post-quantum signature algorithms.

TABLE II. SUMMARY OF THE EXISTING WORKS ON JWT

Ref	Summary	Pros	Cons
[27]	A novel solution using user behavior history was proposed to enhance JWT authentication reliability in web applications.	Enhances authentication reliability.	Users updating devices, switching devices, or auto-updating browsers are penalized.
[28]	To prevent JWT attacks during client-server interactions a random timestamp value technique incorporating client request time and server response time was proposed.	Strengthens protection against replay attacks.	May lead to synchronization issues in distributed systems with varying network latencies.
[29]	Performance analysis of signed algorithms RS256, ES256, and HS256 with JWT in terms of data transfer, token size, and creation time revealed that HS256 outperformed the others.	Provides performance insights.	All the evaluated classical signature algorithms are vulnerable to quantum computing attacks.
[30]	The proposed method enhances JWT security by splitting the token into smaller parts, encrypting each with RSA (1024-bit), and reassembling them into a single token.	Improves security by adding an additional encryption layer.	RSA is vulnerable to quantum attacks, inefficient, and complex.
[31]	A method was suggested to login without making repeated visits to a secured server by utilizing the jumble render technique for JWT-based authentication.	Reduces server load and login latency.	Susceptible to advanced crypto-graphic attacks.
[32]	A token-based authentication mechanism on the Message Queuing Telemetry Transport (MQTT) protocol was presented to enhance security by preventing unauthorized nodes.	Improves network security.	Use of classical vulnerable signature algorithm to validate signatures.
[33]	Comparative performance and security analysis between JWT and Platform Agnostic Security Tokens (PASETO) on the Representational State Transfer (REST) architecture were carried out.	PASETO offers better security than JWT.	PASETO uses traditional cryptographic algorithms, has longer token generation and transfer times than JWT.
[34]	To address the issues of load balancing and communication between agents in distributed systems, a model based on multi-agents with JWT was proposed.	Improves scalability and efficiency.	Asymmetric algorithms used are exposed to threat by quantum computers.
[35]	A token revocation scheme based on Role-Based Access Control (RBAC), was suggested to prevent unauthorized users from reusing tokens.	Effective token revocation.	Signature algorithms used are vulnerable to quantum attacks.
[36]	A JSON Web Signature (JWS) based authentication method was suggested for integration with an adaptive authentication system in healthcare applications.	Facilitates adaptive authentication.	Widely used signature algorithms are prone to quantum threats.
[37]	A scheme based on JWT for authentication and access control was proposed to access protected resources in cloud applications.	Enhances cloud security.	Signature algorithms are prone to exploitation by quantum computers.
[38]	The analysis of JWT with SHA-256 and SHA-512 indicated that JWT with SHA-512 was the suitable option for achieving a faster authentication process.	Improves authentication speed.	Hashing algorithms considered may be exposed to quantum threats.
[39]	A token-based authentication framework was introduced for 5th Generation (5G) Multi-Access Edge Computing (MEC).	Enhances 5G MEC authentication.	Requires higher security levels; use of traditional cryptographic algorithms.
[40]	An efficient approach for implementing JWT authentication using dynamic secret key was proposed for applications in Software Defined Networking (SDN).	Improves key management.	Scheme relies on traditional crypto-graphic mechanisms.
[41]	A token-based authentication mechanism using JWT was proposed for use in RESTful web services.	Simplifies authentication.	Depends on conventional crypto-graphic algorithms.
[42]	A JWT authentication mechanism with HS256 was proposed and tested on Windows Server 2019 and Windows 10.	Demonstrates performance across environments.	Based on symmetric cryptographic algorithms.

The work of Alkhulaifi and El-Alfy [52] evaluated the effectiveness of two lattice-based signature algorithms, quantum tight-secure efficient signatures from standard lattices (qTESLA) and dilithium, against RSA for JWT authentication. The findings demonstrated that the lattice-based algorithms exhibited superior performance, surpassing RSA in terms of both security and efficiency. The results further highlighted the potential of a lattice-based post-quantum signature scheme as a viable alternative to RSA in the quantum computing era. The evaluation was limited to two algorithms, and no hash-based schemes were considered.

Unlike previous studies that focused predominantly on classical cryptographic algorithms for JWT, our approach explores quantum-safe and hybrid cryptographic solutions specifically designed to withstand quantum attacks. Through our review, we identified several key research gaps:

1. While most research on post-quantum cryptography has concentrated on public-key infrastructure (PKI), blockchain, or TLS protocols, our work is the first to systematically evaluate quantum-safe signature algorithms and hybrid algorithms in the context of JWT for secure web authentication.
2. We go beyond merely adopting quantum-safe algorithms and conduct an in-depth performance comparison, revealing the practical impact of integrating these algorithms with JWT across multiple metrics.
3. Additionally, our study progresses from theoretical analysis to practical implementation, providing insights into the deployment of quantum-safe JWT in real-world environments—a gap that has not been fully addressed in the literature.

Current JWT implementations are fundamentally vulnerable to quantum attacks because they depend on cryptographic algorithms such as RSA and ECC, which are based on mathematical problems that can be solved efficiently via quantum algorithms. As quantum computing progresses, the security of JWT tokens deteriorates, making it easier for adversaries to forge signatures, intercept communications, or execute man-in-the-middle attacks. This creates a critical security gap in modern web applications that rely on JWT for authentication.

Our work directly addresses these gaps by providing practical implementation guidance, performance benchmarks, and real-world feasibility analysis for quantum-safe JWT authentication. The results demonstrate that algorithms such as Falcon and SuperFalcon offer viable quantum-resistant alternatives to traditional JWT signing methods while highlighting important performance trade-offs that organizations must consider in their migration to quantum-safe authentication systems.

4. ANALYSIS AND RESULTS

4.1. Experimental Setup

While the post-quantum era [53] anticipates adversaries with quantum computing capabilities, PQC cipher schemes are tailored for implementation on classical computers to safeguard ordinary users. NIST, as part of its standardization process, selects Dilithium, Falcon, and SPHINCS+ as the approved algorithms for post-quantum digital signatures. These algorithms represent a significant security level on the basis of the computational effort required to breach them. The security levels span from level 1, which offers the minimum protection, to level 5, which provides the maximum security. The cryptographic strength of Dilithium and SPHINCS+ used in this work corresponds to the classical equivalent of AES with a 256-bit key size, whereas Falcon offers a security level equivalent to AES with a 192-bit key size. All three standardized algorithms meet the requirements of NIST security level 5.

The experiments were conducted on these three standardized algorithms via a web application built with the MongoDB, Express, React, and Node.js (MERN) stack and deployed on a laptop with an Intel i5 Dual-Core 1.8 GHz processor and 8GB DDR3-1600 RAM, running macOS 12.6.2. We employed algorithm implementations sourced from an open-source JavaScript library, pqcrypto.js, [54, 55], which is a suite of quantum-resistant cryptographic libraries for JavaScript/Web Assembly, along with a few related utilities.

4.2. Test Scenarios and Performance Metrics

The test scenario included 15 automated scripts written in JavaScript to evaluate the performance of each algorithm. A total of 50 responses were collected in common for all the algorithms by utilizing the 15 test scripts that are given below:

1. Registering a new user
2. Should not create a new user if email is already registered
3. Should login if the user's credentials are correct.
4. Should not login with an incorrect user password.
5. Should create a task
6. Should not create a task if the user is not authenticated
7. Should return task of the authenticated user
8. Should not return tasks if the user is not authenticated

9. Should update the completed task
10. Should not update the task if the user is not authenticated
11. Should not delete the task if the user is not authenticated
12. Should delete the task
13. Should register users with low loads, high loads, delays and slow networks
14. Should login users with low load, high load and high latency
15. Should create tasks with high payloads

The selection of performance metrics is crucial for evaluating the practicality of integrating post-quantum signature algorithms with JWT in web applications. We evaluated the performance of the post-quantum and hybrid signature algorithms with the JWT in comparison with that of classical algorithms via the following metrics:

- Token generation time: The amount of time required to generate a JWT token following the login process. This metric quantifies the computational cost of post-quantum algorithms on classical hardware, which is critical for ensuring rapid authentication processes.
- Token verification time: The amount of time required to validate a JWT token for subsequent requests. This measure, like token generation time, assesses computational overhead to ensure that the verification process is efficient and does not cause substantial delays.
- Request header size: The overall size of the Java script object notification (JSON) representation of the hypertext transfer protocol (HTTP) request headers after converting them to a string and calculating byte length. This metric measures the impact of larger post-quantum signatures on the network payload, which influences bandwidth utilization and data transfer efficiency.
- Response body size: The overall size of the HTTP response body, including JWT tokens returned to the client. Like request header size, this metric assesses the impact of post-quantum signatures on payload size and network performance.
- Response time: The time that the server takes to process the request, generate a response, and return it to the client. This metric combines computational and network impacts to provide insights into the overall user experience.

Together, these metrics analyse the real-world usability of post-quantum algorithms with JWT for online application security. Furthermore, in our investigation, we used JavaScript's console.time and console.timeEnd functions to evaluate the timing. These functions are part of the standard console object in JavaScript environments such as web browsers and the Node.js runtime.

4.3. Security analysis

The security of our proposed approach is rooted in the mathematical foundations of the implemented algorithms. Falcon's security is based on the hardness of the short integer solution (SIS) problem over NTRU lattices, which is believed to be resistant to quantum attacks. Dilithium security relies on the module learning with errors (MLWE) and module short integer solution (MSIS) problems, whereas SPHINCS+ derives its security from the properties of hash functions and their quantum resistance. Through our implementation, we maintain the security properties of these algorithms while adapting them for JWT authentication. Additionally, the hybrid approaches provide a defense-in-depth strategy by combining the security benefits of both classical and quantum-resistant algorithms.

4.4. Results

The initial testing aimed at examining the performance of numerous post-quantum cryptography and hybrid algorithms. The stages below detail the encoding and decoding operations involved in the token generation and verification process:

- Encode Key: The base64-encoded key is first transformed to a Buffer and then to a Uint8 array.
- Decode Key: The Uint8Array representation of the key is transformed back to a base64-encoded string via the base64-js library.
- Encode the Token: The token's string representation is transformed to a Uint8 array.
- Decode the Token: The Uint8Array representation of the token is altered back to a string.

The token generation process encompasses several steps: encoding the private key, encoding the token (including both the header and payload), signing the token with the private key, and decoding the token. The token generation process can be represented as follows:

$$\text{Signature} = \text{DecodeKey}\ (\text{Sign}\ (\text{EncodeToken}\ (\text{Header+Payload}), \text{EncodeKey}\ (\text{PrivateKey}))) \quad (1)$$

$$\text{Token generation time} = \text{TEncodeprivate key} + \text{TEncodetoken} + \text{TSigntoken} + \text{TDecodetoken} \quad (2)$$

In terms of the average token generation time with post-quantum signature algorithms, as shown in Fig. 3a, dilithium has a minimum of 2.23 ms, and SPHINCS+ has a maximum of 6.1 s. With hybrid signature algorithms, SuperDilithium has a

minimum time of 2.51 ms, and SuperSphincs has a maximum time of 5.9 s. Subsequently, the time required for token verification was assessed. The token verification time includes the time taken for encoding the public key, encoding the signature, retrieving the token via the public key, and finally verifying the token.
The token verification time can be represented as:

$$\text{Signature Verification} = \text{DecodeToken (Verify (EncodeKey (Signature), EncodeKey (PublicKey)))} \quad (3)$$

$$\text{Token Verification Time} = \text{TEncodepublic key} + \text{TEncodesignature} + \text{TRetrievetoken} + \text{TVerifytoken} \quad (4)$$

As illustrated in Fig. 3b, among the quantum-safe signature algorithms tested, Falcon demonstrated the fastest average token verification time at 0.65 ms. For hybrid signature algorithms, SuperFalcon proved most efficient, with a time of 1.81 ms, whereas SuperSphincs took the longest at 18.48 ms. Following token verification, the sizes of request headers and response bodies were measured. The request header size is calculated by converting the request headers to a JSON string and measuring its byte length. The request header sizes for various HTTP requests, such as the GET, POST, PUT, and DELETE methods, were considered to evaluate the performance of the algorithms. However, for SPHINCS+, only registrations and signs in log results were considered because of the larger signature size. To log the HTTP request and response information, the Morgan logging library in a Node.js application was used.

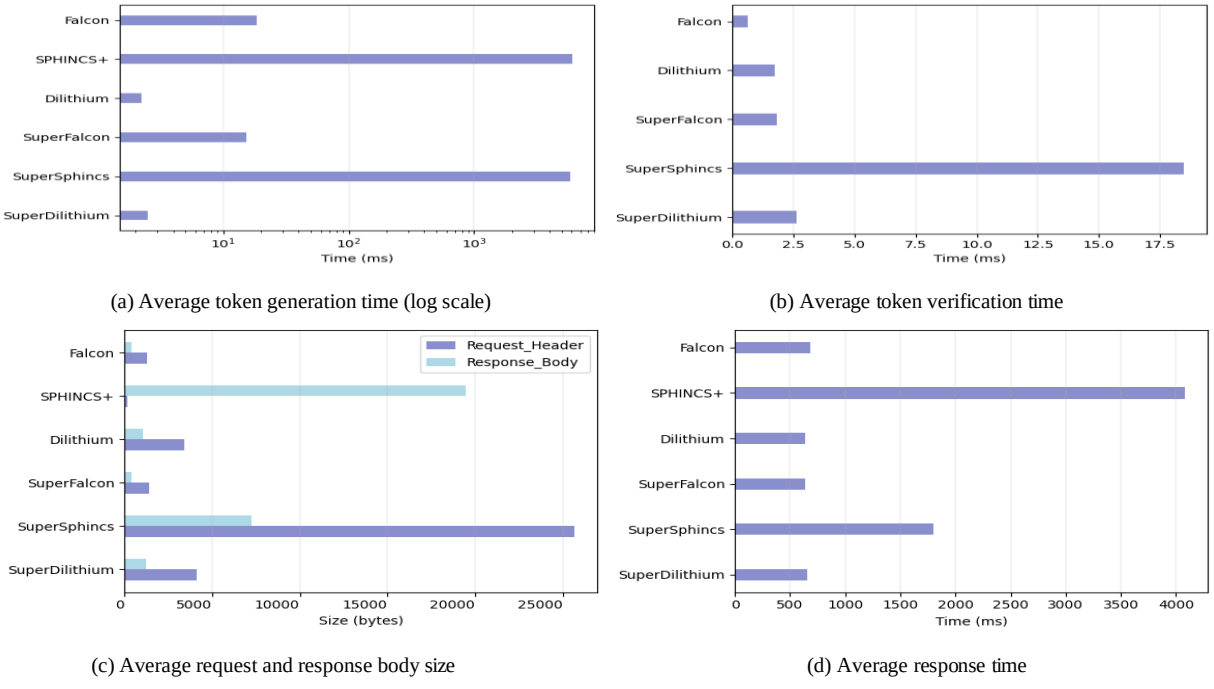
$$\text{LogData} = [\text{HTTP Method, URL, Status, RequestHeaderSize, ResponseBodySize, ResponseTime}]$$

With respect to the PQC algorithms shown in Fig. 3c, Falcon has the minimum size of 1317.24 bytes for the request header and 424.44 bytes for the response body. Similarly, in hybrid algorithms, SuperFalcon features a minimum size of 1393.34 bytes for the request header and 436.02 bytes for the response body.
Finally, the response time was measured. The response time represents the total time taken for sending a request from the client and receiving a response from the server.

$$\text{Response Time} = \text{Time when the response is sent} - \text{Time when the request is received}$$

TABLE III. AVERAGE TIMES AND SIZE COMPARISON OF POST-QUANTUM AND HYBRID SIGNATURE ALGORITHMS

Algorithms	Token gen	Token ver	Request header	Response body	Response time
Dilithium	2.23	1.74	3428.54	1030.18	640.13
SPHINCS+	6084.51	–	193.55	19477.93	4082.55
Falcon	18.68	0.65	1317.24	424.44	681.65
SuperDilithium	2.51	2.62	4139.53	1244.41	651.79
SuperSphincs	5879.18	18.48	25686.24	7267.20	1804.25
SuperFalcon	1.19	1.81	1393.34	436.02	636.72



(a) Average token generation time (log scale)

(b) Average token verification time

(c) Average request and response body size

(d) Average response time

Fig. 3. Efficiency metrics for quantum-safe and hybrid signatures

Fig. 3d presents the results of the average response time. Among the postquantum signature algorithms, Dilithium exhibited the shortest response time of 676.96 ms, whereas among the hybrid signature algorithms, SuperFalcon achieved the minimum response time of 636.7 ms. Table 3 displays the mean execution time for both token generation and token verification across 50 requests, alongside the request header size, response body size, and response time for a range of post-quantum and hybrid signature algorithms. The findings reveal that Falcon and SuperFalcon perform competitively across most metrics, offering a strong balance between security and efficiency.

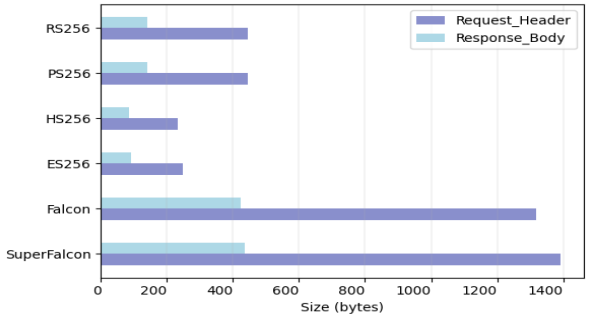
5. COMPARATIVE ANALYSIS

To ensure comprehensive security in the quantum era, it is crucial to compare traditional signature algorithms with post-quantum and hybrid algorithms. To facilitate this comparison, we generated traditional RSA, ECDSA, and RSA-PSS keys via industry-standard tools such as OpenSSL. For RS256, we created 4096-bit RSA key pairs via OpenSSL, as 4096 bits provide enhanced security for RSA keys in this context. ES256 keys were generated with OpenSSL via the prime256v1 curve (equivalent to P256). For PS256, we again used OpenSSL to create 4096-bit RSA key pairs. By utilizing these widely recognized cryptographic tools, we ensured that our traditional keys were generated via robust, standardized methods, allowing for an accurate and relevant comparison between traditional algorithms and the post-quantum alternatives examined in this study.

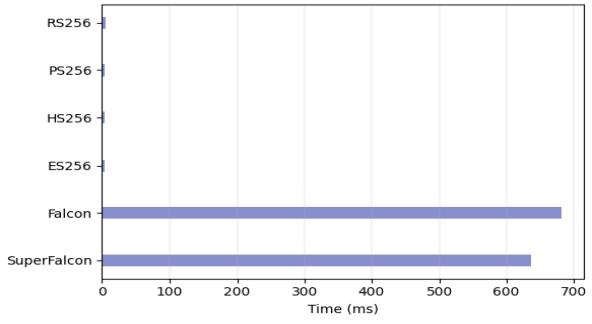
We initially conducted a comparative analysis of traditional signature algorithms to establish a performance baseline. Our results in Table 4 present a critical baseline for comparing post-quantum solutions against current industry standards. The performance metrics of classical algorithms demonstrate why they remain popular in current implementations, with consistently low response times and minimal payload sizes. Among traditional algorithms, HS256 demonstrated the lowest response time of 3.45 ms, outperforming even the best quantum-resistant option. Among the other traditional algorithms evaluated, ES256 was the fastest at 4.03 ms, followed closely by PS256 at 4.10 ms, whereas RS256 was notably slower

TABLE IV. AVERAGE TIMES AND SIZES OF CLASSICAL ALGORITHMS

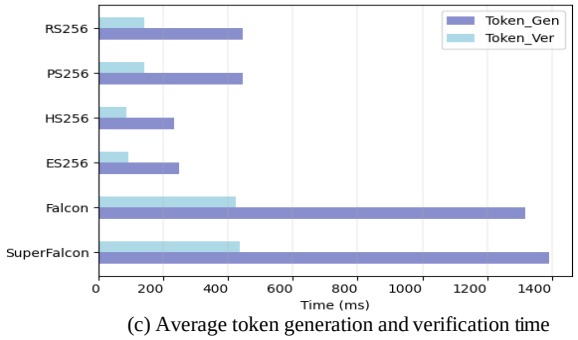
Algorithms	Token gen	Token ver	Request header	Response body	Response time	Security Level
RS256	7.06	6.10	445.51	139.51	4.89	Not Quantum-safe
ES256	0.67	6.13	246.54	89.75	4.03	Not Quantum-safe
PS256	6.90	5.34	445.52	139.53	4.10	Not Quantum-safe
HS256	0.38	5.66	232.17	86.16	3.45	Not Quantum-safe



(a) Average request and response body size



(b) Average response time



(c) Average token generation and verification time

Fig. 4. Comparison of efficiency metrics for classical, hybrid and quantum-safe signatures

at 4.89 ms. While HS256 shows superior performance in terms of response time, it is crucial to consider the security implications. HS256 uses a single secret key for both signing and verification, which can be a security risk in certain

scenarios. In contrast, RS256, ES256, and PS256 use public-key cryptography, offering better security but at the cost of slightly increased response times.

Following our baseline analysis, we expanded our investigation to compare traditional algorithms with the leading post-quantum and hybrid options. Fig. 4a compares the average request header and response body sizes of traditional signature algorithms with those of the best-performing post-quantum and hybrid signature algorithms. This comparison illustrates that the post-quantum and hybrid algorithms yield larger sizes for both headers and bodies. Fig. 4b contrasts the average response times of conventional signature algorithms with those achieved by the top-performing post-quantum and hybrid signature algorithms. The best-performing quantum-resistant algorithm, Falcon (0.68 s), and the hybrid algorithm SuperFalcon (0.64 s), while slower than HS256, still offer competitive performance. These results suggest that for applications requiring long-term security or those dealing with highly sensitive data, the slight performance trade-off of Falcon or SuperFalcon may be well justified given their quantum resistance. When considering the feasible bearer token size for inclusion in the authorization header, employing SPHINCS+ poses a challenge because of its larger signature sizes, as cookie sizes are usually limited to 4096 bytes. Fig. 4c shows the average token generation and verification times of classical, quantum-safe, and hybrid algorithms. The results indicate that the post-quantum and hybrid algorithms require much less time than traditional algorithms do. These findings have direct implications for the real-world adoption of quantum-safe JWT in web applications.

The fast token generation and verification times exhibited by Falcon and hybrid algorithms such as SuperFalcon make them ideal candidates for secure web services that require high-throughput and low-latency authentication. Moreover, despite its strong quantum resistance, SPHINCS+ may be less suited for applications where the network bandwidth and token size are critical due to its enormous signature size. This study provides a detailed performance comparison across multiple quantum-safe and hybrid algorithms, allowing developers and researchers to make informed decisions about the trade-offs between security, performance, and scalability when integrating quantum-safe cryptography into JWT-based systems. Given the importance of cryptographic implementations, we built a code quality evaluation for our research process. This step was critical for ensuring the accurate implementation of our algorithms and minimizing potential flaws that could affect our performance comparisons. The SonarQube static code analyser toolkit 10.3.0.82913 was used to validate the code structure, demonstrating that it was written securely and without flaws or vulnerabilities.

6. LIMITATIONS

The analysis focuses on selected NIST-standardized post-quantum algorithms—Falcon, Dilithium, and SPHINCS+—and their hybrid variations. The incorporation of new cryptographic algorithms could provide further insight into the trade-offs between performance and security. The study was carried out in a controlled software testing environment, which may not accurately reflect the complexity of real-world deployment settings. Challenges such as the increased token size of post-quantum algorithms may have an impact on bandwidth and latency in restricted network contexts, particularly in IoT and mobile applications. Furthermore, while NIST's validation supports the resilience of post-quantum algorithms, this work did not address adversarial testing unique to JWT implementations.

7. CONCLUSION AND FUTURE WORK

This work critically examines the effectiveness and feasibility of using quantum-safe and hybrid signature techniques with JWT to secure web applications from quantum threats. Traditional algorithms, such as HS256, excel in terms of response time and data efficiency, but they are vulnerable to quantum computing breakthroughs, forcing a strategic change to quantum-resistant alternatives. Post-quantum algorithms such as Falcon, Dilithium, and SPHINCS+ provide strong security since they rely on difficult mathematical issues such as lattice-based and hash-based cryptography, which quantum computers cannot solve. Among these, Falcon achieved the best balance of security and performance. Hybrid algorithms, such as SuperFalcon, improve this balance by integrating classical and quantum-safe techniques, providing useful alternatives for future-proofing JWT systems. This study emphasizes the necessity for a strategic shift towards quantum-safe cryptographic techniques to safeguard web applications in the quantum era. As NIST continues its standardization efforts, future work should examine the performance of these algorithms on various hardware platforms and their integration into commercial applications.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

Not applicable.

Acknowledgement

The authors would like to thank the anonymous reviewers for their efforts.

References

[1] OWASP, "OWASP Top Ten," OWASP Foundation. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed: January 08, 2025].

[2] Akanksha and A. Chaturvedi, "Comparison of different authentication techniques and steps to implement robust JWT authentication," in 7th International Conference on Communication and Electronics Systems (ICCES), 2022.

[3] M. Jones, B. Campbell, and C. Mortimore, "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants," 2015.

[4] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," 2015.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Review, vol. 41, pp. 303–332, 1999.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," Proc. 28th Annu. ACM Symp. Theory of Comput. - STOC, 1996.

[7] C. Balamurugan, K. Singh, G. Ganesan, and M. Rajarajan, "Post-quantum and code-based cryptography—some prospective research directions,"Cryptography, vol. 5, p. 38, 2021.

[8] S. Li, Y. Chen, L. Chen, J. Liao, C. Kuang, K. Li, W. Liang, and N. Xiong, "Post-quantum security: Opportunities and challenges," Sensors, vol. 23, p. 8744, 2023.

[9] H. Faria and J. M. Valença, "Post-Quantum Authentication with Lightweight Cryptographic Primitives," Cryptology ePrint Archive, Paper 2021/1298, 2021.

[10] P. Tandel and J. Nasriwala, "Secure authentication framework for IoT applications using a hash-based post-quantum signature scheme," Service- Oriented Computing and Applications (SOCA), 2024.

[11] Y. Ostrianska and Y. Gorbenko, "Current state of standardization of post-quantum cryptography," Physico-Mathematical Modelling and Informational Technologies, pp. 52–56, 2023.

[12] A. A. Giron, "Migrating applications to post-quantum cryptography: Beyond algorithm replacement," in Proc. 20th Int. Conf. Security and Cryptography, 2023.

[13] JWT.io, "JSON Web Tokens," [Online]. Available: https://jwt.io/. [Accessed: January 08, 2025].

[14] IETF, "JSON Web Signature (JWS)," IETF RFC 7518. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7518. [Accessed: January 08, 2025].

[15] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Signature (JWS)," 2015.

[16] A. Karakaya and A. Ulu, "A survey on post-quantum based approaches for edge computing security," WIREs Computational Statistics, vol. 16, 2024.

[17] G. Yalamuri, P. Honnavalli, and S. Eswaran, "A review of the present cryptographic arsenal to deal with post-quantum threats," Procedia Computer Science, vol. 215, pp. 834–845, 2022.

[18] C. Chen, "The performance analysis of post-quantum cryptography for vehicular communications," 2022.

[19] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," Quantum Information and Computation, vol. 3, pp. 317–344, 2003.

[20] D. Moody, "Fast Fourier sampling over NTRU lattices digital signature standard," 2023.

[21] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Dilithium: A lattice-based digital signature scheme," IACR Trans. Cryptographic Hardware and Embedded Systems, pp. 238–268, 2018.

[22] F. Liu, Z. Zheng, Z. Gong, K. Tian, Y. Zhang, Z. Hu, J. Li, and Q. Xu, "A survey on lattice-based digital signatures," Cybersecurity, vol. 7, 2024.

[23] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn, "SPHINCS: Practical stateless hash-based signatures," in Advances in Cryptology – EUROCRYPT 2015, Lecture Notes in Computer Science, pp. 368–397, 2015.

[24] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS+ signature framework," in Proc. 2019 ACM SIGSAC Conf. Computer and Communications Security, 2019.

[25] F. Hazzaa, M. M. Hasan, A. Qashou, and S. Yousef, "A new lightweight cryptosystem for IoT in smart city environments", Mesopotamian Journal of CyberSecurity, vol. 4, no. 3, pp. 46–58, 2024.

[26] Mohammed Abbas Fadhil Al-Husainy, Bassam Al-Shargabi, Shadi Aljawarneh, "Lightweight cryptography system for IoT devices using DNA,"Computers and Electrical Engineering, vol. 95, 2021.

[27] A. Bucko, K. Vishi, B. Krasniqi, and B. Rexha, "Enhancing JWT authentication and authorization in web applications based on user behavior history," Computers, vol. 12, p. 78, 2023.

[28] S. Ahmed and Q. Mahmood, "An authentication-based scheme for applications using JSON Web Token," in Proc. 22nd Int. Multitopic Conf. (INMIC), 2019.

[29] A. Rahmatulloh, R. Gunawan, and F. M. S. Nursuwars, "Performance comparison of signed algorithms on JSON Web Token," IOP Conf. Series: Materials Science and Engineering, vol. 550, 2019, p. 012023.
[30] M. Malvin and C. Safitri, "JSON Web Token leakage avoidance using token split and concatenate in RSA256," Indonesian J. Computing, Engineering and Design (IJoCED), vol. 5, pp. 43, 2023.
[31] E. S. Chopra, A. Singh, and A. Singh, "JSON Web Token jumble render technique based authentication scheme for Android applications," Recent Advances in Computing Sciences, pp. 129–133, 2023.
[32] F. A. Shodiq, R. R. Pahlevi, and P. Sukarno, "Secure MQTT authentication and message exchange methods for IoT constrained device," in Proc. Int. Conf. Intelligent Cybernetics Technology and Applications (ICICyTA), 2021.
[33] A. F. Nugraha, H. Kabetta, I. K. S. Buana, and R. B. Hadiprakoso, "Performance and security comparison of JSON Web Tokens (JWT) and platform agnostic security tokens (PASeTo) on RESTful APIs," in Proc. IEEE Int. Conf. Cryptography, Informatics, and Cybersecurity (ICoCICs), 2023.
[34] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, "Authentication and load balancing scheme based on JSON Token for multiagent systems,"Procedia Computer Science, vol. 148, pp. 562–570, 2019.
[35] R. G. K. Babu, A. Badirova, F. F. Moghaddam, P. Wieder, and R. Yahyapour, "Authentication and access control in cloud-based systems," in Proc. 14th Int. Conf. Ubiquitous Future Networks (ICUFN), 2023.
[36] V. Krishnan, C. S. Sreeja, S. Binu, and M. Misbahuddin, "A JSON Web Signature based adaptive authentication modality for healthcare applications," in Proc. IEEE Int. Conf. Public Key Infrastructure and its Applications (PKIA), 2022.
[37] O. Ethelbert, F. F. Moghaddam, P. Wieder, and R. Yahyapour, "A JSON Token-based authentication and access management schema for cloud SaaS applications," in Proc. IEEE 5th Int. Conf. Future Internet of Things and Cloud (FiCloud), 2017.
[38] N. Rasyada, "SHA-512 algorithm on JSON Web Token for RESTful web service-based authentication," J. Applied Data Sciences, vol. 3, pp. 33–43, 2022.
[39] W. Niewolski, T. W. Nowak, M. Sepczuk, and Z. Kotulski, "Token-based authentication framework for 5G MEC mobile networks," Electronics, vol. 10, p. 1724, 2021.
[40] P. Varalakshmi, G. B., V. S. P., and D. T., S. K., "Improvising JSON Web Token authentication in SDN," in Proc. Int. Conf. Communication, Computing and Internet of Things (IC3IoT), 2022.
[41] M. Haekal and Eliyani, "Token-based authentication using JSON Web Token on SIKASIR RESTful web service," in Proc. Int. Conf. Informatics and Computing (ICIC), 2016.
[42] A. R. Ficry Cahya Ramdani and R. N. Shofa, "Implementation of JSON Web Token on authentication with HMAC SHA-256 algorithm,"SISTEMASI, vol. 12, pp. 194, 2023.
[43] M. Raavi, S. Wuthier, P. Chandramouli, Y. Balytskyi, X. Zhou, and S.-Y. Chang, "Security comparisons and performance analyses of post-quantum signature algorithms," Applied Cryptography and Network Security, Lecture Notes in Computer Science, pp. 424–447, 2021.
[44] T. G. Tan, P. Szalachowski, and J. Zhou, "Challenges of post-quantum digital signing in real-world applications: a survey," Int. J. Inf. Secur., vol. 21, pp. 937–952, 2022.
[45] Y. Yuan, J. Xiao, K. Fukushima, S. Kiyomoto, and T. Takagi, "Portable implementation of post-quantum encryption schemes and key exchange protocols on JavaScript-enabled platforms," Secur. Commun. Netw., vol. 2018, pp. 1–14, 2018.
[46] D. Ghinea, F. Kaczmarczyck, J. Pullman, J. Cretin, S. Kölbl, R. Misoczki, J.-M. Picod, L. Invernizzi, and E. Bursztein, "Hybrid post-quantum signatures in hardware security keys," in Lecture Notes Comput. Sci., Appl. Cryptogr. Netw. Secur. Workshops, 2023, pp. 480–499.
[47] I. Tzinos, K. Limniotis, and N. Kolokotronis, "Evaluating the performance of post-quantum secure algorithms in the TLS protocol," J. Surveillance Secur. Safety, vol. 3, pp. 101–127, 2022.
[48] A. Ashraaf, "Analysis of post quantum cryptography algorithms concerning their applicability to IoT devices," 2024.
[49] J. Hekkala, K. Halunen, and V. Vallivaara, "Implementing post-quantum cryptography for developers," in Proc. 8th Int. Conf. Inf. Syst. Secur. Privacy, 2022.
[50] H. C. Ukwuoma, G. Arome, A. Thompson, and B. K. Alese, "Post-quantum cryptography-driven security framework for cloud computing," Open Comput. Sci., vol. 12, pp. 142–153, 2022.
[51] L. Gan and B. Yokubov, "A performance comparison of post-quantum algorithms in blockchain," J. British Blockchain Assoc., vol. 6, pp. 1–10, 2022.
[52] A. Alkhulaifi and E.-S. M. El-Alfy, "Exploring lattice-based post-quantum signature for JWT authentication: Review and case study," in Proc. IEEE 91st Veh. Technol. Conf. (VTC2020-Spring), 2020.
[53] M. G. Y. Albahri, H. A. Aljanabi, and A. K. Ali, "Securing Tomorrow: Navigating the evolving cybersecurity landscape," Mesopotamian Journal of CyberSecurity, vol. 4, no. 1, pp. 1–3, Mar. 2024.
[54] "https://github.com/cyph/pqcrypto.js/," Accessed: January 08, 2025.

[55] "https://www.npmjs.com/," Accessed: January 08, 2025.
[56] O. Aljumaiah, W. Jiang, S. R. Addula, and M. A. Almaiah, "Analyzing cybersecurity risks and threats in IT infrastructure based on NIST framework," *J. Cyber Secur. Risk Audit.*, vol. 2025, no. 2, pp. 12–26, 2025.
[57] A. K. Abed , Tran., “Utilizing Artificial Intelligence in Cybersecurity: A Study of Neural Networks and Support Vector Machines”, BJN, vol. 2025, pp. 14–24, Feb. 2025, doi: 10.58496/BJN/2025/002.