iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67704

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Securing Web Applications: An Analysis of Attack Methods and Effective Countermeasures

Sambhav Dudhwewala[1], Nakibar Zaman[2], Aman Alok[3], Ahmad Shakib[4], Shubham Dubey[5]
Department of Computer Science Jain(Deemed-to-be)University Bangaluru, Karnataka, India

Abstract: ***Content Delivery Networks (CDNs) play a crucial role in improving web performance, ensuring scalability, and enhancing security for websites and online services. This research paper delves into the methodologies underpinning CDN architecture, focusing on techniques for optimizing content delivery, reducing latency, and mitigating bandwidth consumption.***
Keywords: CDN, Website, bandwidth, encryption, DNS

I. INTRODUCTION

The primary goal of CDNs is to reduce latency, optimize bandwidth usage, and enhance user experiences by minimizing delays in content delivery. They also play a critical role in improving the scalability and reliability of websites by offloading traffic from the origin server, especially during traffic surges. Beyond performance optimization, CDNs contribute to security by offering protection against Distributed Denial of Service (DDoS) attacks and ensuring secure content delivery through encryption technologies such as SSL/TLS. Quantum communication, a revolutionary approach to secure data transmission, leverages the principles of quantum mechanics to ensure unprecedented levels of security. By utilizing quantum key distribution (QKD), it allows for the exchange of encryption keys that are theoretically immune to interception, as any attempt to eavesdrop on the key alters its quantum state and makes it detectable. In the digital age, where cyberattacks and data breaches are growing more sophisticated, integrating quantum communication into existing infrastructures like Content Delivery Networks (CDNs) presents a promising advancement for secure online interactions. A content delivery network (CDN) is a network of servers spread out geographically in order to store web content at different locations near users, enhancing speed and decreasing delays. Nevertheless, the extensive deployment of CDN nodes can create security risks, especially in areas such as key exchanges and data transmission among nodes. Quantum communication provides a groundbreaking solution here. Through the integration of QKD with CDN networks, sensitive information and encryption keys exchanged between CDN nodes can be safeguarded, maintaining both fast and secure content delivery. Quantum communication enhances the security foundation of CDNs, ensuring they remain secure against interception and data tampering as they expand worldwide. This method ensures a future in which the quick distribution of content through CDNs is combined with the powerful security of quantum encryption, providing users with a smooth and secure online browsing experience. With cyberattacks becoming more advanced, targeting sensitive data and exploiting vulnerabilities in key exchanges, ensuring secure data transmission through CDNs is crucial. This is how quantum communication presents an innovative method for enhancing security. Utilizing the concepts of quantum mechanics allows for quantum communication, especially quantum key distribution (QKD), to offer a level of security that cannot be breached by traditional computational techniques. QKD enables two parties to swap encryption keys using quantum particles such as photons, guaranteeing that any effort to intercept or manipulate the key will be noticed, since measuring disrupts the quantum state.

Incorporating quantum communication into CDN systems represents a substantial advancement in safeguarding data transmission. In a content delivery network setting, many nodes are always in communication to exchange encrypted content and encryption keys. Presently, traditional encryption techniques (such as RSA or Diffie-Hellman) are employed to protect these transactions. However, quantum computers pose a threat as they may be able to decipher this information in the future. By implementing quantum communication in the key exchange procedure, CDNs can add a layer of security that can withstand threats from both classical and quantum computing.

II. RESEARCH OBJECTIVES

1) Identify Common Vulnerabilities and Attack Vectors:

To systematically identify and categorize the most common vulnerabilities and attack vectors that are currently exploited in website hacking, including SQL Injection, Cross-Site Scripting, and other prevalent techniques.

2) Evaluate Effectiveness of Current Security Measures:
To assess the effectiveness of existing security measures and practices in mitigating website hacking risks, including the use of firewalls, encryption, and secure coding standards.

3) Investigate Emerging Hacking Techniques and Trends:
To analyze emerging hacking techniques and trends that pose new challenges to website security, and understand how these advanced threats are evolving beyond traditional defense mechanisms.

4) Develop Recommendations for Improved Security Practices:
To provide actionable recommendations and best practices for website developers and administrators aimed at enhancing security and reducing vulnerabilities in web applications.

5) Enhance User Education and Awareness:
To evaluate the role of user education in preventing website hacking and develop strategies for effectively increasing cybersecurity awareness and training among users.

6) Assess Tools and Technologies for Threat Detection and Response:
To investigate and evaluate the most effective tools and technologies available for detecting, responding to, and mitigating website hacking attempts, including intrusion detection systems and incident response solutions.

7) Balance Security Measures with User Experience:
To explore methods for balancing robust security measures with an optimal user experience, ensuring that protective measures do not unduly compromise the usability and functionality of websites.

III. BACKGROUND OF RESEARCH

The digital landscape has transformed dramatically over the past few decades, offering unprecedented opportunities for communication, commerce, and information sharing. However, this growth has also given rise to significant security challenges, among which website hacking stands as a critical concern. Website hacking refers to unauthorized access or manipulation of websites by individuals or groups, often referred to as hackers, who exploit vulnerabilities to achieve various malicious objectives.

IV. PROBLEM STATEMENT

In the current digital environment, websites are vital platforms for businesses, organizations, and individuals to communicate, transact, and share information. Yet, the increase in cyberattacks against these sites presents substantial risks to the integrity of data, privacy, and overall trust of users. The frequency and complexity of website hacking, which includes methods like SQL injection, cross-site scripting (XSS), session hijacking, and Distributed Denial of Service (DDoS) attacks, phishing attacks [10]have increased.
Many websites are still at risk of these types of attacks even with security tools and practices, because of insufficient security measures, outdated systems, or misconfigurations. This leads to monetary losses, security breaches, and harm to reputations. Dealing with these weaknesses necessitates a thorough grasp of the typical attack paths, the changing tactics of cyber attackers, and successful approaches to safeguarding websites.
This research aims to investigate common website hacking methods, evaluate their effects on current web applications, and suggest strong security measures to reduce these dangers.

A. Attacking Methods To Exploit Vulnerabilites In Website

Weaknesses in the coding, configuration, or infrastructure of a website can be exploited by attackers to compromise the site's security, integrity, or availability - these are known as website vulnerabilities. These weaknesses may result in unauthorized entry, data leaks, vandalism, or service interruptions. There are some major attacking methods to exploit vulnerabilities of websites.[2]

1) Cross-site Scripting (XSS)
It is a type of vulnerability that allows attacker to inject malicious scripts into the original or legitimate website. When the attacker is able to find some input field or URL parameter.

The attacker identifying potential vulnerabilities in a web application by finding user input fields without proper validation and sanitization, URL parameters means a query strings which can be manipulated. [3] When the attacker is able to find vulnerability in the website than attacker use some methods to inject malicious javascript code like:

- Stored XSS

In this attacker injects the script into a persistent location like a database which will be served to user later.

- Reflected XSS

This a type of xss attack where the attacker crafts url parameter to send to the user. The script looks like this (for example): http://example.com/search/?q=<script>alert('xss');</script> [5]

2) DDOS Attack

The attacker uses DDOS to disrupt the normal functioning of the server by overwhelming it with flood of traffic, the flood can be anything like TCP flood, UDP flood, Ping of death and so on.

The attacker performs this activity with the help of botnet, botnet is a software which is used to perform DDOS attack. Attackers send botnet through malicious email/phishing and when the user mistakenly opens the malicious email then botnet immediately installed on the user's device.

Then the attacker uses Command-and-Control server to control botnet remotely. Now the botnet is under control in attacker and attacker will decide which attack should perform according to the situation like flooding, protocol attacks or application layer attack.

3) API Injection Attack

API Injection is a malicious code into API requests to exploit weaknesses in API. For perform API injection attacks the attacker will search for exposed or poorly secured API endpoints by using API enumeration where it try different API URLs and parameters to discover vulnerabilities. When a developer creates server which need API for requesting the data then JavaScript code applied there to make it responsive on that code they have to write endpoint like POST/api/user/login or GET/api/user/register. Once a vulnerable API endpoint discovered the attacker will send crafted API requests containing malicious payload these request may contain SQL, NoSQL or OS commands which lead to expoit vulnerability. Now the API request is send to the server and server accepts it because server think this requests came from legitimate user.

4) DNS Poisoning

In this attack, the attacker sends fake or crafted DNS request to DNS server of the target. When a user types a domain name in their browser, their computer requests the matching IP address from a DNS server. After that, the DNS server provides the accurate IP address, allowing the user to access the desired website.

The attacker corrupts the DNS cache or records by sending false information about the IP address. When a user makes a DNS request then the attacker intercepts the query before it reaches the original DNS server. So this interception happen through various techniques like MITM. Attackers didn't send IP address instead of sending fake DNS requests. So when the server accepts and responds to this fake DNS request then the final ACK packet transfer to the hacker's IP address.

Once the fake DNS server stored in the DNS record then the user will try to get the information from the original server on that time the user get transmitted to the other instead of original one because in the DNS record there is fake request stored which will send user to the IP address of the hacker.

5) Web Application Firewall Bypassing

Attackers uses many method to bypass the web application like encoding and unclear methods. There are some methods to bypassing WAF [4]

- URL Encoding

It is a mechanism of encoding the characters of the URLs there are certain special characters which have other meaning in HTTP requests like '=' used to separate keys from values in query parameters, '?' used to separate the URL path from the query string.

When these characters need to include in the URL content as a part then URL encoding replaces this characters with the '%' sign like '=' will become %3D, ';' will become %3B, '<' will become %3C, '>' will become %3E. For example if a hacker wants to inject XSS by searching the input in the search engine so the hacker have to write JavaScript without encoding which looks like this <script>alert='XSS'</script> now the hacker have to encode this script so that it wouldn't be blocked by WAF, the encoded script will look like this %3Cscript%3Ealert%28%27XSS%27%29%3C%2Fscript%3E. When this script is typed into the URL then the browser will de-code this script after the target server accepts the request send by the browser according to the script.

- Manipulating HTTP headers

Attackers can modify or inject unexpected content into HTTP headers to bypass WAF. Most of the traffic uses standard HTTP methods like GET and POST some WAFs may not acceptably monitor or filter less common methods like PUT, HEAD, OPTIONS and TRACE so that attackers use this methods to bypass WAF because it wouldn't examine closely.

V. IMPORTANCE AND RELEVANCE OF THE RESEARCH

A. RELEVANCE

1) Increasing dependence on internet and cloud-based services:

With the rapid expansion of cloud computing and internet services, safeguarding online assets is more crucial than before. CDNs are essential in providing protection through global-scale distributed security features. This study is very important because it examines the role of CDNs in the broader context of cloud security and safeguarding distributed systems.

2) Reacting to growing cyber security risks:

Cyberattacks, such as brute force, API injection, and JavaScript exploitation, are increasing in frequency and complexity. There is an urgent requirement for efficient prevention strategies. The research emphasizes the significance of modern security solutions in countering evolving threats by examining the role of CDNs in preventing such attacks.

3) Assisting in adhering to regulations:

Numerous sectors must adhere to data protection laws that mandate strong security protocols, such as GDPR, HIPAA, and PCI-DSS. CDNs are able to assist in meeting these regulatory requirements through services such as encryption, traffic monitoring, and attack prevention.

4) Improving Efficiency While Ensuring Safety:

CDNs offer not just security advantages but also boost websites and APIs' speed by distributing content nearer to users and improving delivery.

B. IMPORTANCE

1) Securing Web Infrastructure:

Websites, APIs, and online services play a crucial role in both contemporary businesses and daily activities. Security breaches on these platforms can cause significant disruptions, data breaches, and financial damages.[9]

2) Improving the security of APIs:

APIs play a key role in the operation of contemporary web applications, but they are commonly vulnerable to cyber-attacks such as injection, brute force, and exploitation.

3) Adjusting to changing online dangers:

Cyber threats are always changing, as attackers create advanced methods to evade standard security measures.

4) Encouraging the implementation of a security approach with multiple layers:

The application's security is optimized when it is safeguarded by a combination of different layers. CDNs provide an extra layer of security in addition to server-side input validation, encryption, and authentication methods.

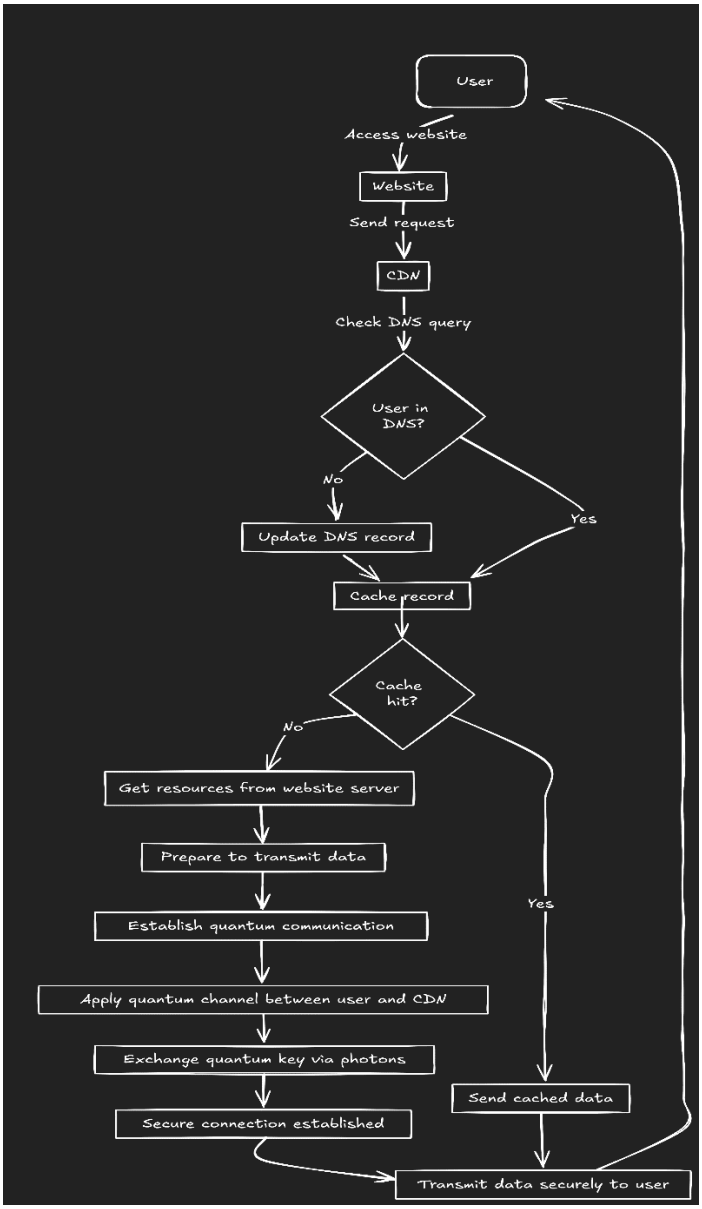
5) Enhancing security in a cost-effective manner:

Small and medium-sized businesses might lack the means to create intricate security infrastructure. CDNs provide cost-effective and scalable options for improving security, making high-level protection available.

VI. LITERATURE REVIEW

This literature review presents a detailed summary of research advancements in web application security, emphasizing protection of third-party resources, identification and prevention of typical attacks, and utilization of advanced technologies such as machine learning and deep learning for spotting vulnerabilities. This review of literature focuses on the advancements and obstacles in protecting web applications. Exploration of key areas includes Subresource Integrity automation, phishing detection using ML, vulnerability analysis with deep learning, and proactive defense using models. Lately, the increasing complexity of cyber threats and the emergence of quantum computing have required improvements in encryption technologies to protect internet communications. With the threat of quantum algorithms like Shor's algorithm compromising traditional cryptographic methods, researchers are now turning to Quantum Key Distribution (QKD) to enhance security in online communications.

VII. METHODOLOGY



The methodology we deliver here is CDN (Content Delivery Network) it is an extra security layer on website to prevent from such attacking methods like DDOS, MITM, XSS, API Attacks and so on. In this methodology we will explain about how CDN works as well as how it will keep safe websites from attacking.

When a user using its browser and types a website URL then the request will be send to the website's server for establishing the connection between user and the server. Now the website is using CDN the DNS will direct the user's browser to nearest CDN server instead of original website's server to check that the user is already been on that website if yes than whatever user went on that website it's all been saved on website cached space. To implement CDN in your website either we register ourselves to use features of CDN or we can just write code for it, and the code goes like this:

<script src="https://cdn.cloudfare.com/"></script>

We use script function in the html page because whatever content we're putting here all are going to save in the CDN server which means all resources like HTML, CSS, JAVASCRIPT codes will be secure by the CDN. [6]

1) PHASE I

Connection Establishment between user and the website's server though browser

When a user try to access some website using any browser than the browser will look into the cache that if the user is already accessed this website if yes than the browser will load the cached data of the website and if not then the browser will look into the DNS server of the website. After the DNS lookup process the browser will query the server for checking the user's IP address is already in there or not if not then the IP address of the user will be recorded on the DNS record.

2) PHASE II

CDN Edge server

Now when the IP address of the user is stored in the record then the browser will send the request to the CDN for establishing the connection between the CDN server and the browser.

a) Establishment Connection

The connection establishment is done by TLS/SSL handshake it secures the connection between the user, browser and the server. The TLS is a cryptographic protocol used to secure the connection between the server and the client, the browser will send a message to server with encryption algorithms, hashing algorithms and key exchange methods set by client. When the server receives the encrypted message from the client then the server decrypt the message with the same encryption algorithm, private key and the hashing algorithm.

Now after the server receiving the message it will send a digital certificate to client including the server's private key and the server's identity like domain name, IP address and so on. After validating the server's certificate now the user proceeds with creating a common secret key, to be utilized for symmetric encryption. For example if a user use RSA then the client generates a Pre-Master secret key and encrypts it with the server's public key then client sends the encrypted pre-master secret key to the server.

Each and every connection establishment there is a session keys to secure the connection between the client and the server this secure feature done with the help of session key and this keys are used for symmetric encryption so both the server and the client use the same session keys for encrypting and decrypting the data during the secure session.

After sending message both the client and server finishes the message by sending FIN requests to each other and now both are able to transfer the data.

b) Request Headers

When the user sends request to the server then the browser sends an http/https request to the server along with the Request Headers which may contains information like content type, user details, location, and cookies after that the data's been stored then the CDN will start sending the resources to the client this will depend on when the CDN check its cache record that if the data's already stored or not.

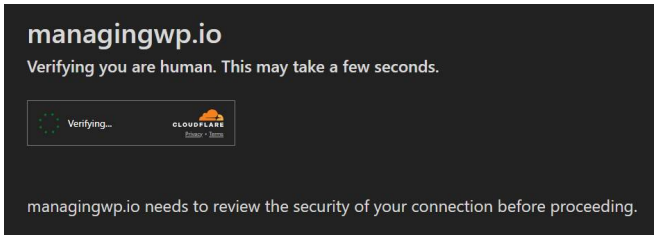
c) CDN cache record

In the CDN cache record it will check its record to find that the user is already went through the website or not then it performs two methods cache hit and cache miss.

In cache hit, the user had already gone through the website and because of that whatever he accessed the resources it's been saved on the cache record of CDN.
In cache miss, the CDN will check the cache record and try to find out that the user is been there or not by query the record when the record didn't find any information about the user on that time CDN will start sending the resources to the user.

3) PHASE III
CDN RESPOSNE
a) Loading Content
When the CDN accepted the request and make the entry of the user in its record then CDN will start loading the content and transmitting it to the user's browser. The content is generated according to the user request For example if a user search an image like "A Puma shoe on amazon" then the browser will provide every possible result for it. Now the browser will provide a link and user will click on it after that the link which is clicked by the user a request will go to the CDN server because amazon contains CDN that's why CDN interferes it and try establish connection between user and the client.

b) Additional Request
Sometimes when we visit website there is a blank comes with asking for verification of the user or just buffering with the loading symbol on the page that means the CDN is just try to verifying that the user is valid or not if the CDN still have some doubt then it starts the process of CAPTCHA for verify the user.



SOLUTION
Quantum Communication
Incorporating quantum communication into a website would be extremely experimental, as it mainly revolves around sending quantum states (such as qubits) through secure channels, which is quite distinct from traditional communication on websites. Yet, I can explain a detailed conceptual method for merging quantum communication principles with a classical web interface through Quantum Key Distribution (QKD). Here is the hypothetical way it could function:

- *Quantum Key Distribution (QKD) Channel*

For set up the QKD we need to establish the channels to securely exchange key using quantum particles like photons instead of relying on traditional key exchange methods. QKD ensures that the keys shared between two parties are completely secure from eavesdroppers.[7]

➢ *Transmission of qubits, which are quantum particles:*
In quantum key distribution, two parties (like a website server and a user) utilize quantum particles, often photons, for information exchange.
These photons have the ability to be polarized in various orientations to symbolize quantum states or "qubits." Every qubit has the possibility of being in a combination of states, unlike classical bits that are limited to 0 or 1.
Different bits can be represented by polarization states such as vertical (|), horizontal (-), diagonal (/), and anti-diagonal ().

➢ *BB84 Protocol (A Basic Quantum Key Distribution Protocol):*
The BB84 protocol, created by Bennett and Brassard in 1984, is a popular quantum key distribution technique. It includes two individuals participating.
Alice (Sender): Alice (the server of the website) creates a random string of qubits (coded with photons) and transmits it to Bob (the recipient).

Bob (Recipient): Bob randomly measures the qubits using different bases such as rectilinear or diagonal. Because of the principles of quantum mechanics, Bob is unable to simultaneously measure both the position and polarization with complete certainty.

- *Web Infrastructure*

The traditional method of website operation on the internet, known as the classical web infrastructure, involves the use of HTTP/HTTPS protocols for communication. It manages typical web operations such as page loading, form submission, user input management, etc. In quantum communication, traditional infrastructure continues to handle these functions, while quantum communication enhances security with key exchange**.**

➢ *Transmission of data over HTTP/HTTPS:*

HTTP, or HyperText Transfer Protocol, is the method used for managing communications between web servers and web clients (such as user browsers). Nonetheless, HTTP is not encrypted, leaving it susceptible to attacks. HTTPS combines HTTP with SSL/TLS encryption. It secures data transmission by encrypting data such as login credentials or payment details, preventing eavesdroppers from accessing it. HTTPS continues to be essential, even with quantum communication, as quantum encryption will mainly focus on protecting key exchanges. The content on the website (including text, images, etc.) will continue to be transmitted over HTTPS.

➢ *Interplay of Quantum and Classical Layers:*

Classical communication focuses on the operations of the website and the transmission of data (such as user forms and product orders) between the client and server in both directions. This includes common tasks such as retrieving a website or submitting user login information. Quantum communication will sit on top of the existing classical infrastructure, mainly for securely exchanging encryption keys (through QKD). After exchanging these keys, the quantum-generated keys will be used for applying traditional classical encryption methods, such as AES.

➢ *Ensuring continued compatibility:*

Despite the added security benefits of quantum communication, websites must continue to utilize the current internet protocols (HTTP/HTTPS) and infrastructure that is widely used. This indicates that users don't require any specific equipment to access the website or view its content. They will access a standard website using regular browsers, with the encryption occurring in the background.

➢ *Encryption and authentication of sessions:*

After exchanging a secure key through quantum communication, it can be utilized to encrypt all confidential communication during the user's session, including login data and personal information. Conventional methods of session management, like securing cookies, keeping authentication tokens, and handling sessions, would remain unchanged, but with increased security using quantum-generated keys.

- *Error Correction and Eavesdropping Detection*

Errors may arise in the transfer of quantum particles (qubits) between the server and the user in quantum key distribution (QKD). Moreover, a significant benefit of QKD is that it enables both parties to identify any attempts of eavesdropping on their communication. Error correction guarantees accurate transmission of the key, while eavesdropping detection prevents interception during key exchange.

➢ *Error correction in Quantum Key Distribution*

Mistakes can occur in Quantum Key Distribution when noise in the quantum channel, like fiber optic cables or satellite links, causes errors while transmitting qubits. For instance, a qubit designated as "0" might be incorrectly read as "1" because of interference or flaws in the transmission phase.

Error Correction Purpose: Error correction aims to guarantee that the sender (server) and receiver (user) possess matching keys following the quantum key exchange. Any discrepancies in the keys may compromise the security.

Error Correction Algorithms: Following the key exchange, the server and user can publicly check a segment of their keys over a standard channel (such as the internet) to detect errors. They utilize error correction methods (like Cascade or LDPC codes) to fix discrepancies without disclosing excessive details about the real key.
The server can help the user fix specific bits where a discrepancy is detected without revealing the entire key. This guarantees that both parties will have the identical ultimate key.

➢ *Eavesdropping Detection*

One essential concept in quantum mechanics is that observing a quantum system causes interference with it. In the scenario of QKD, if an eavesdropper (Eve) attempts to intercept and measure the qubits sent between the server and the user, she will inevitably disrupt the system, altering the state of the qubits. When Eve eavesdrops on a qubit, it changes its polarization or other characteristics. Therefore, when the authorized receiver (the user) observes the qubit, they may receive a value that is not equivalent to the one transmitted by the server. Publicly comparing and sharing error rates: Following the exchange of qubits between the server and user, they are able to communicate (via a classical channel) about the measurement bases used for each qubit (while keeping the actual values undisclosed). Next, they analyze a small segment of their unique code. If the discrepancy between the values exceeds a specific threshold, it suggests the existence of a potential eavesdropper. If Eve attempted to measure the qubits, she would create noticeable errors that would notify the rightful parties. If the error rate exceeds a certain threshold, they will halt the key exchange process and begin anew.

- *User Interaction and Data Encryption*

This stage focuses on the website's interaction with users and ensures encryption of all sensitive data transmitted between the user and the server. [8]The objective is to uphold a secure means of communication, safeguarding important data such as login information, personal details, and payment credentials from unauthorized access or manipulation by malicious individuals.

➢ *Interaction with the user:*

User interacts with the website's frontend, containing forms, buttons, input fields, and other interactive features. Commonly, technologies such as HTML, CSS, and JavaScript are used to manage these inputs. Whenever a user sends information (such as login details), it is transferred to the backend (server) for further action.

➢ *Data Encryption During Transmission:*

Despite using the regular internet, data is transmitted securely through an encrypted HTTPS connection while the user interacts with the website. HTTPS encrypts sensitive information like usernames, passwords, credit card numbers, and other data using the SSL/TLS protocol. This ensures that no unauthorized person can listen in on the information during its transfer between the user's browser and the web server.
Symmetric encryption involves the use of traditional encryption methods such as AES (Advanced Encryption Standard) for data encryption by both the server and client. AES utilizes a single key, known as a shared secret, for both encrypting data on the client side and decrypting it on the server side.

➢ *Using Quantum-Secured Keys:*

QKD and Data Encryption: When a website employs quantum key distribution (QKD), the encryption key shared between the user and server is very secure due to being created through quantum communication. This makes sure that the key could not have been intercepted or tampered with by anyone. After securely exchanging the key using QKD, it is employed in a classical encryption algorithm (such as AES) to encrypt all sensitive data exchanged during user interactions. Original text: The quick brown fox jumps over the lazy dog.[9]

➢ *End-to-End Encryption:*

Client-Side Encryption: Before being transmitted over the network, the user's data is encrypted on their browser using a key that is quantum-secured. This ensures no unauthorized access or tampering while in transit.
Server-Side Decryption: The data is decrypted by the server upon its reception. Because the quantum key exchange ensures that only the server and user possess the encryption key, it is impossible for any third party (such as an attacker or network administrator) to decrypt the data.

- *Authentication and Session Management:*

  After the user has been verified (for example, after logging in), the server has the ability to create a session key (or token) in order to securely manage the user's session. This session key could be safeguarded with quantum-safe techniques as well.

  Encryption of session: The data shared in the user's session, such as browsing history, purchase orders, and profile changes, will remain encrypted with the quantum-secured key.

VIII. MATHEMATICAL SOLUTION

In this mathematical solution we will take it as an example like we will show this solution to address how it can be solved through mathematically.

So, first the time happens between user and CDN edge server which called as latency we will take it as 20 ms.

Round-trip time = 20 ms X 2 = 40 ms

And the time for CDN to original server it will take more latency if the content is not cached.

Latency = 100 ms

Round-trip time = 100 ms X 2 = 200 ms

Now we will calculate the time of the data transfer

Data size = 5 MB = 40 megabits

Bandwidth = 50mbps

Transfer time = 40 megabits / 50 mbps = 0.8 seconds

Now we have calculated that it takes 0.8 seconds for CDN to reach and get the resources and now we will calculate how much time CDN will take to transfer it to the user.

Data size = 5 MB = 40 megabits

Bandwidth = 100ms

Transfer time = 40 megabits / 100 mbps = 0.4 seconds

As we can see that CDN takes 0.4 seconds to transfer the data to the user.

Now we will calculate the total time for requesting to transferring the data.

Total time = 200 ms + 40 ms + 0.8 seconds + 0.4 seconds

= 2 sc + 0.4 sc + 0.8 sc + 0.4 sc = 3.6 seconds

So the total time to render the resources by CDN is 3.6 seconds.

REFERENCES

[1] N. Albalawi, N. Alamrani, R. Aloufi, M. Albalawi, A. Aljaedi, and A. R. Alharbi, "The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities," Electron., vol. 12, no. 12, 2023, doi: 10.3390/electronics12122664.

[2] R. L. Alaoui and E. H. Nfaoui, "Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review," 2022. doi: 10.3390/fi14040118.

[3] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," 2023. doi: 10.3390/electronics12061333.

[4] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," 2021. doi: 10.3390/make3030034.

[5] P. M. D. Nagarjun and S. S. Ahamad, "Cross-site scripting research: A review," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 4, 2020, doi: 10.14569/IJACSA.2020.0110481.

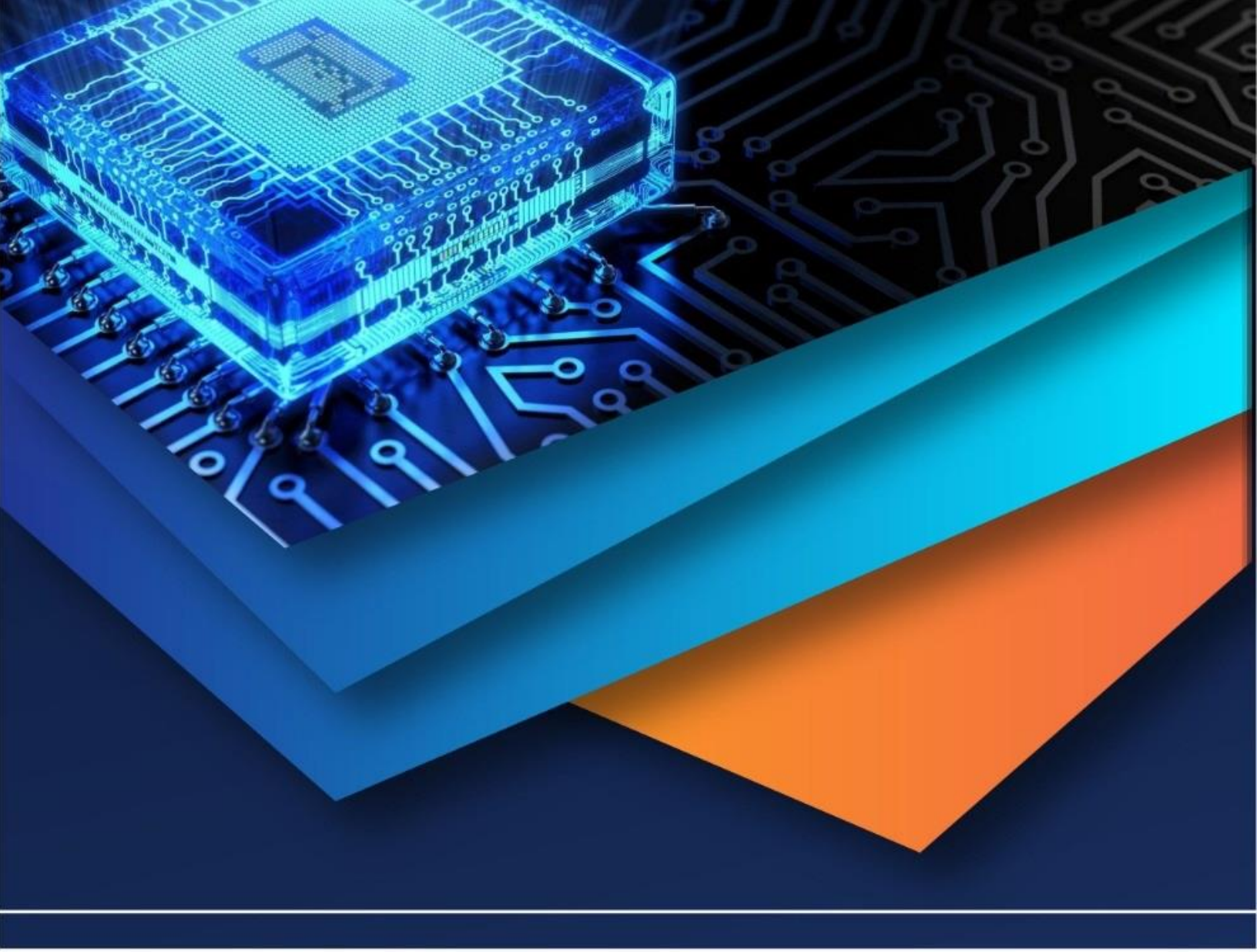
[6] S. Yevseiev et al., "DEVELOPMENT OF AN IMPROVED SSL/TLS PROTOCOL USING POST-QUANTUM ALGORITHMS," Eastern-European J. Enterp. Technol., vol. 3, no. 9(123), 2023, doi: 10.15587/1729-4061.2023.281795.

[7] Editor, "Quantum Technologies Flagship Final Report," High-Level Steer. Comm., no. February, 2017.

[8] T. M. Fernandez-Carames, "From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things," 2020. doi: 10.1109/JIOT.2019.2958788.

[9] L. Erdődi and F. M. Zennaro, "The Agent Web Model: modeling web hacking for reinforcement learning," Int. J. Inf. Secur., vol. 21, no. 2, 2022, doi: 10.1007/s10207-021-00554-7.

[10] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," 2021. doi: 10.3390/make3030034.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY