International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Securing Web Applications Against Vulnerabilities Using Quantum Cryptography and Reverse Proxy Authentication for Cookie Protection

Dr. G. Aravind Swaminathan Ph.D*, Ajitha Devadharshini B**
*(Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: aravindswaminathan.g@francisxavier.ac.in)
** (Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: ajithadevadharshinib.ug22.cs@francisxavier.ac.in)

----------**********----------

Abstract:

Web applications are widely used today. A web application may be vulnerable to session hijacking and cookie theft. In this project, we aim to secure web applications against session hijacking and cookie theft. The proposed system uses quantum cryptography and reverse proxy authentication. Upon successful login to the server, the server generates a session key for the session. The session key is transmitted to the client using Quantum Key Distribution (QKD) protocol so that the session key is kept secret from any third party. The AES or Advanced Encryption Standard, is used to encrypt session cookies. This is done with the help of a session key. Reverse proxy servers authenticate the cookies. They verify whether the cookies are modified or not. If the cookies are not modified, the reverse proxy servers send the cookies to the server. If the cookies are modified, the modified cookies cannot be used for accessing the web application. We can also give the auto login facility to the client by using the encrypted session cookies. The client will not have to re-enter the username and password for accessing the web application. Hence, the proposed system is effective in securing web sessions by using quantum cryptography and reverse proxy authentication.

Keywords — **Quantum Cryptography, AES Encryption, Session Key, Cookie Security, Reverse Proxy, Web Application Security, Session Hijacking Prevention, Secure Login, QKD, Cryptographic Techniques**

----------**********----------

I. INTRODUCTION

Web applications are vulnerable to many types of attacks, such as session hijacking and cookie theft. While many applications protect themselves using HTTPS and basic cookie encryption, this is not enough to protect the application from attacks. Our project involves the use of Quantum Cryptography, AES encryption and reverse proxy authentication to secure the web sessions. After a user has logged into the web application, the server will produce a session key for the session and then send this key to the client using the Quantum Key Distribution (QKD). The session key will then be used with the AES encryption to lock and therefore secure the cookies, while the protected cookies will then be verified by the reverse proxy servers before they can proceed and access the web application. Therefore the cookies cannot be hijacked and any malicious users will not be able to access the web application. We can also use the encrypted cookies for an automatic login so users do not have to keep re-entering their username and password. Therefore our system is a highly practical method of securing web applications.

II. OBJECTIVES

Session and cookie security, a crucial part of contemporary web applications, is necessary to protect user data, stop illegal access, and guarantee confidence in online services. Conventional security

techniques, like HTTPS and simple cookie encryption, are often susceptible to replay attacks, cookie theft, and session hijacking. To get around these restrictions, this project develops the Quantum Cryptography and AES-based Cookie Protection System with Reverse Proxy Authentication, which employs secure session key sharing and encryption to safeguard session data.

The primary objective of this project is to create and execute a safe, automated web session management system that safeguards user information, stops illegal access, and guarantees trustworthy authentication. The following particular goals define the system's scope:

A. *Secure Session Key Generation and Sharing:*

Using Quantum Key Distribution (QKD), create a distinct session key for every user session and securely distribute it between the client and server. This keeps attackers from intercepting and guarantees that the session key is only accessible to those who are authorised.

B. *AES-Based Cookie Encryption:*

Use the securely shared session key to encrypt cookies that contain user and session information. This guarantees that cookies cannot be decrypted or abused even if they are accessed by hackers.

C. *Reverse Proxy Verification of Requests:*

Before sending requests to the application server, use a reverse proxy server to verify incoming requests by decrypting cookies and verifying session keys. This stops session hijacking and unwanted access.

D. *Automatic Login with Encrypted Cookies:*

Provide users the option to log in automatically using encrypted cookies without having to repeatedly enter their username and password, all the while maintaining security through session key verification.

E. *Real-Time Detection and Alerts for Tampering Attempts:*

To ensure proactive security, incorporate a monitoring system that notifies the server of any attempts to tamper with or intercept session keys or cookies.

III. MODULES AND ALGORITHMS

A cybersecurity-based system called the Secure Web Application Protection System using Quantum Cryptography and Reverse Proxy Authentication was created to shield web applications from unauthorised access, cookie theft, and session hijacking. The system incorporates Reverse Proxy authentication for request validation, AES encryption for cookie protection, and Quantum Key Distribution (QKD) for safe session key sharing. Secure login, encrypted session management, cookie-based automatic authentication, and real-time tampering detection are all made possible. By guaranteeing user session confidentiality, integrity, and authentication, the system improves web security.

A. *Modules*

1) *User Authentication and Session Management Module :*

In this module, user login is securely achieved by verifying the username and password of the user. Once the user is authenticated, a unique key is generated by the server for the user session. The user session is securely managed, and the time period for the session is set to avoid any kind of misuse of the session.

2) *Quantum key distribution, or QKD Module:*

In this module, the generated key is securely shared between the client and server by applying the Quantum Key Distribution method. If any unauthorized user tries to intercept the key during the time of transmission, the state of the key changes, thus preventing any kind of misuse of the key or any other unauthorized copy of the key being created.

3) AES-Based Cookie Encryption Module :

This module will be used for encrypting the session information stored in the cookie with the AES symmetric encryption algorithm and the securely shared session key. This ensures that the cookie cannot be stolen by the attacker, as it cannot be decrypted without the session key.

4) Reverse Proxy Authentication Module:

This module will be used for providing the reverse proxy, which acts as a security gateway between the client and the application server. This reverse proxy will be responsible for decrypting the cookie, validating the session, and checking for any tampering before allowing the request.

5) Session Validation and Automatic Login Module :

This module will be used for providing the automatic login feature. This feature allows the users to log in automatically, as the encrypted cookie will be sent by the browser whenever the user tries to revisit the application.

6) Intrusion Detection and Logging Module :

This module is responsible for monitoring suspicious activities like incorrect cookie attempts, session mismatches, and unauthorized access. All these activities are logged for auditing purposes. Alerts can be sent out in case of unusual behavior.

B. Algorithms

1) Algorithms for the Advanced Encryption Standard (AES):

Advanced Encryption Standard (AES) is the primary symmetric key algorithm adopted in the project for cookie session encryption. The algorithm works based on blocks of data and requires a secret key for the encryption and decryption process. While performing the encryption process, the data is transformed by performing multiple operations. While performing the decryption process, the same secret key is utilized. Without the secret key, the decryption process is not feasible, which ensures the data is secure against any attacker.

2) Quantum Key Distribution (QKD):

Quantum Key Distribution is adopted for the secure transfer of the secret key between the client and the server. The protocol is based on the application of the principles of quantum mechanics. The protocol is based on the transfer of photons. If any attacker tries to intercept the quantum bits during the transfer process, the state of the quantum bits is changed, which ensures the secure transfer of the key.

3) Reverse Proxy :

The reverse proxy performs the validation logic by decrypting the cookies received and matching them with the session key. If the data received from the decryption process matches the session data and falls within the expected time frame, the request is allowed to proceed to the application server. If not, the request is denied and logged.

C. Functional Modules

1)Secure Login and Session Creation Logic :

Upon the verification of the credentials, the system generates a session key and securely shares it with the help of QKD. This session information is encrypted with the AES algorithm and stored as a cookie in the browser.

2)Cookie Verification and Access Control Logic :

For every incoming request, the encrypted cookie information is verified with the reverse proxy. If the cookie information is valid and has not been tampered with, the user is granted access; otherwise, the request is denied, and the user is redirected to the login page.

IV. METHODOLOGIES

A. User authentication and creation of session keys:

In the first step, the username and password are validated against the server's database. After the authentication process is successful, the session key is generated. This ensures the confidentiality and integrity of the data being sent. The generated key is unique to that particular session. This guarantees that the data being transmitted is secure.

B. Secure Session Key Sharing Using QKD :

In the second step, the generated key is shared securely between the client and the server. Quantum Key Distribution (QKD) is used to share the key. Quantum channels are used to transmit the key. If any attacker tries to intercept the key, the state of the quantum channel is changed. This ensures the integrity and confidentiality of the key, which is being shared.

C. AES-Based Cookie Encryption::

After the session key is shared, the session data is encrypted with the symmetric key encryption method, AES. The encrypted information is saved as a cookie in the user's web browser. Even if an unauthorized individual accesses the cookie, they won't be able to use it to access the session information without permission. This module is responsible for maintaining the confidentiality and integrity of session information.

D. Reverse Proxy Validation and Access Control :

All requests to the application server are channeled through the reverse proxy server. The reverse proxy server uses the session key to decrypt the cookie and verify the session information. The session information is checked for validity, timestamp, and integrity. If the cookie is invalid, tampered with, or has an expired timestamp, the request is blocked, thus denying unauthorized access. This module is responsible for secure and controlled access to the application.

E. Automatic Login and Session Continuity :

If an existing user sends an encrypted cookie, the system uses the session key to decrypt the cookie and allows access to the system without the need to re-enter the username and password.

F. System for Intrusion Detection and Warning:

This module detects any suspicious activity, such as invalid cookie attempts, session mismatches, and attempts to tamper with the sessions. An alert message is displayed for the detected anomalies, and all the activities are recorded for auditing and security analysis. This ensures that the integrity of the web application remains high.

G. Session Termination and Cleanup :

Once the user has logged out or the session has expired, the session key is made invalid, and the encrypted cookies are deleted. This ensures that the old sessions are not used again, and the user has to re-authenticate for any new login attempts.

V. EXISTING SYSTEM

A. Traditional Password-Based Authentication System :

The majority of web applications make use of simple authentication using a username and password. However, this type of system is highly vulnerable to attacks like stealing passwords, brute force attacks, etc. A hacker might gain access to a user's account by stealing the credentials. In addition, the session data cannot be protected from being hijacked.

B. Session Management Without Encryption :

The majority of the current applications make use of session management using cookies without encryption. This makes the system vulnerable to attacks by a hacker who accesses the cookies. As a result, the session might be stolen by the hacker.

C. Basic Cookie Encryption System :

In the majority of the applications, a symmetric encryption technique is used to protect the session management using a cookie. However, the session key is sent using a traditional channel. As a result, the session key might be reused by the hacker.

D. Web Applications Lacking Reverse Proxy Security:

The web applications that do not use reverse proxy security face the issue of not being able to validate incoming requests effectively. This makes them vulnerable to attacks such as replay attacks, requests being tampered with, and unauthorized access. Reverse proxy servers help to add an additional layer of validation that is currently lacking.

E. Lack of Quantum Key-Based Security:

The current web security systems do not incorporate quantum key distribution for the sharing of secure session keys. This makes the session keys vulnerable to being intercepted while being sent, which makes the system highly vulnerable to attacks.

F. Limited Real-Time Monitoring and Tamper Detection :

The current web security systems do not incorporate real-time monitoring to detect cases of tampering and unauthorized access. Additionally, there are no alerts for cases of unauthorized activities and session breach.

G. Dependence on Repeated Login Credentials :

In the majority of the traditional systems, the users are forced to repeatedly input the credentials like username and password to initiate a new session.

VI. PROPOSED SYSTEM

The suggested Secure Web Application Protection System via Quantum Cryptography and Reverse Proxy Authentication provides a highly secure web session environment by incorporating AES encryption, Quantum Key-based session management, and reverse proxy authentication. It provides a highly secure login facility, encrypted cookies, automatic session validation, and real-time detection of any attempt to tamper with the sessions.

A. AES and Quantum Key-Based Session Security :

The core of the suggested system is the AES encryption technique, combined with Quantum Key Distribution (QKD) to ensure highly secure sessions. A unique key is generated for each user login, and it is securely shared between the **server** and the client using Quantum Key Distribution. Even in the event of a key being compromised, any attempt to tamper with it will be detected in real time, thereby securing the sessions from any kind of unauthorized access.

B. Module for Reverse Proxy Authentication:

In the reverse proxy authentication module, the reverse proxy is the gateway between the client and the application server. It authenticates the request by decrypting the cookie using the session key and checking the session. It blocks the request if the cookie is tampered with, has expired, or is invalid. This ensures that the web application is not accessed by unauthorized users.

C. Automatic Login with Encrypted Cookies :

The users who have already created an account can log in automatically by using the encrypted cookie. The users do not need to re-enter the username and password. The system decrypts the cookie using the session key and checks the session. It allows the users if the request is valid. This ensures high security while logging in.

D. Tamper Detection and Alert Mechanism :

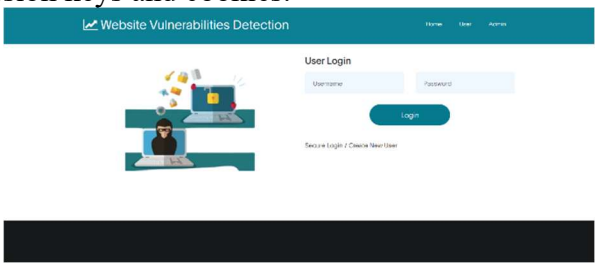
In the tamper detection and alert mechanism, the system is designed in such a way that if an attacker tries to tamper with the cookie, the change is detected immediately. The alerts are sent, and the suspicious activity is logged. This ensures the security of the application.

D. Web-Based Interface :

The system employs a web interface developed using React.js or Flask, which allows users to access the system securely using a web interface. Users can log in to the system, establish a secure session, and manage their account using the web interface. Similarly, the system allows administrators to monitor the session, logs, etc., in real time.

F. Robust Session Lifecycle Management :

The system manages the entire lifecycle of the user session. It invalidates the session keys and removes the cookies when the user logs out. It also employs renewal mechanisms to ensure that the upcoming session is secure. It prevents the reuse of the old session keys and cookies.



VI. OUTPUT

A. Home Page Interface

The home page functions as the web application's primary user interface and offers access to various modules, including the sections for administrative access and user login. It explains the system's goal, which is to identify and stop online threats like cookie attacks and session hijacking. Users can effortlessly access the secure authentication system thanks to the interface's straightforward and user-friendly design.

The developed web application's home page is displayed in Figure 5.1. Home, User Login, and Admin Login are among the navigation options on the page that let users access various system features.

Figure 5.1 Home Page of Website Vulnerability Detection System

B. User Login Page

Registered users can access the system by entering their username and password on the user login page. The system creates a distinct session ID that is saved in the browser as a secure cookie following successful authentication. Throughout the user's interaction with the application, this cookie aids in maintaining the user session.

Before allowing access to protected resources, the login module makes sure that user credentials are validated. To stop hackers from obtaining or altering the session data, the session ID created during login is encrypted.
The system's user login interface is depicted in Figure 5.2.

Figure 5.2 User Login Page

C. Admin Approval Module

Managing user accounts and keeping an eye on system activity fall under the purview of the admin module. The administrator can examine user registration requests in this module and decide whether to accept or reject them in accordance with verification requirements.

This feature guarantees that the web application can only be accessed by authorised users. In addition to offering control over system functions, the admin interface contributes to the platform's security and integrity.

The admin panel, depicted in Figure 5.3, is where user accounts are authorised prior to system access.

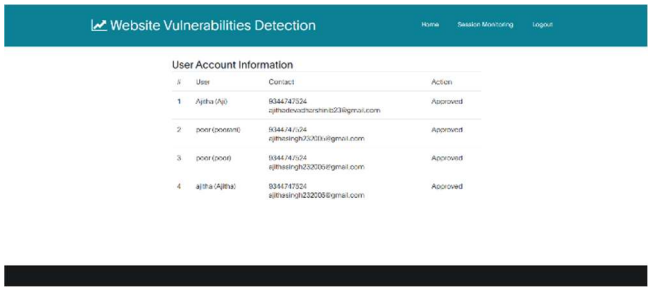


Figure 5.3 Admin User Approval Page

D. Attack by Hijacking a Session:

The system's security is assessed using a simulation of a session hijacking attack. In this module, an attacker tries to obtain a legitimate user's session ID and use it to access the web application without authorisation.

The system keeps an eye on session activity and looks for unusual requests. The system detects suspicious activity and blocks requests when an attacker attempts to use or alter a stolen session ID.

The application's simulated session hijacking attack attempt is shown in Figure 5.4.

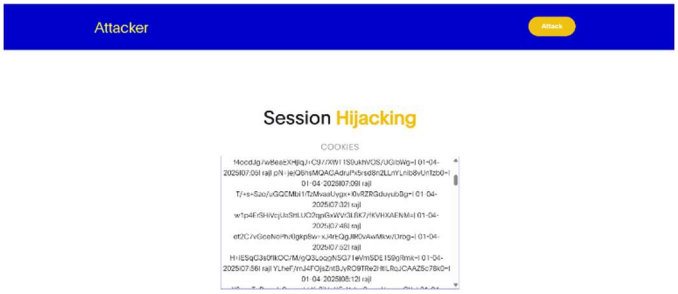
Figure 5.4 Session Hijacking Attack

E. Cookie Protection Mechanism

Before the application uses session cookies, the cookie protection module makes sure they are safely encrypted and verified. This stops hackers from using stolen cookies to pretend to be authentic users.

To safeguard session data, the system makes use of reverse proxy validation and secure cookie attributes. Before being handled by the application server, every request made by the client browser is checked by the reverse proxy server.

The system recognises an anomaly and prevents



access if a cookie is altered, reused, or accessed from an unauthorised source.

The system's cookie protection mechanism is depicted in Figure 5.5.

Figure 5.5 Secure Cookie Protection Page

VII. CONCLUSIONS

For the purpose of protecting web applications from session hijacking and cookie stealing, the Quantum Cryptography and AES-Based Cookie Protection System with Reverse Proxy Authentication can be implemented. This ensures that the sessions of the users remain secure and tamper-proof, providing them with a seamless and secure experience through the automatic login feature with encrypted cookies, session validation, and real-time tamper detection.

A. Enhanced Web Security :

The system ensures that the web application remains secure from any kind of unauthorized access, replay attacks, and cookie stealing, allowing only valid users to access the web application.

B. Automated Session Management :

The entire session management process, including session creation, validation, and termination,

becomes completely automated, reducing the chances of human error.

C. Real-Time Tampering Detection:

The inclusion of monitoring and alerting systems helps in the early detection of any suspicious activity, which improves the reliability and integrity of the system.

D. Scalability and Adaptability :

The modularity of the system helps it to be scalable, i.e., it can be extended with features such as multi-level authentication, mobile integration, and scaling up the system without compromising the integrity.

ACKNOWLEDGMENT

Therefore, my sincere and heartfelt thanks go to my mentor, the esteemed Dr. G. Aravind Swaminathan, Ph.D., for his invaluable mentorship, encouragement, and expert knowledge that have greatly helped shape the course and quality of this research. Without his mentorship, it is unlikely that the research would have achieved its maximum potential.

Further, I would like to extend my sincere and heartfelt thanks to the esteemed members of the faculty, industry experts, and professionals for their invaluable recommendations and expert suggestions that greatly enriched the research methodology and analysis.

Lastly, I would like to extend my sincere and heartfelt appreciation to my friends, colleagues, family, and peer group for their encouragement, support, and assistance. Their trust and confidence in me have inspired me and motivated me throughout the course of this research.

REFERENCES

[1] P. Szalachowski, “Password-authenticated decentralized identities,” IEEE Trans. dependable, secure computer.., vol. 19, no. 2, pp. 734–746, 2022.

[2] H.-K. Lo, M. Curty, and K. Tamaki, “Secure quantum key distribution,” Nature Photon., vol. 8, pp. 595–604, 2014.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, 2nd ed. Boca Raton, FL, USA: CRC Press, 1997.

[4] C. H. Bennett and G. Brassard, "Quantum cryptography: Coin tossing and public key distribution," in Proc. IEEE Int. Conf. Computers, Systems and Signal Processing, Bangalore, India, 1984, pp. 175–179.

[5] M. Wegmüller, N. Gisin, O. Guinnard, and H. Zbinden, “High resolution fiber distributed measurements with coherent OFDR,” in Proc. ECOC’00, Munich, Germany, 2000, paper 11.3.4, p. 109.

[6] S. Singh, S. Chatterjee, and P. Singh, “Enhancing web security using AES and session key management,”

[7] R. Kaur and A. Kaur, “Reverse proxy based web application security framework,” *J. Inf. Secur. Appl.*, vol. 58, p. 102748, 2021.

[8] IEEE, “IEEE Standard for Local and metropolitan area networks—Secure web application protocols,” *IEEE Std. 802.11i*, 2004.

[9] D. J. Bernstein and T. Lange, *Post-Quantum Cryptography*, Springer, 2017.

[10] H. Lo, X. Ma, and K. Tamaki, “Secure quantum key distribution: From theory to practice,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 21, no. 3, pp

