Experiment No. 7

Title: Challenge-Response Protocol

Batch:A3 **Roll No.: 16010421073** **Experiment No.: 7**

Title: Design and implement a VLab for Challenge-Response protocol.

Resources needed: Windows/Linux OS

Theory:

Pre Lab/ Prior Concepts:

Consider a situation where a server (for example, a base station) wants to authenticate a client (a mobile phone user) by confirming that the client has the correct password (say, a 5-digit password PSWD).

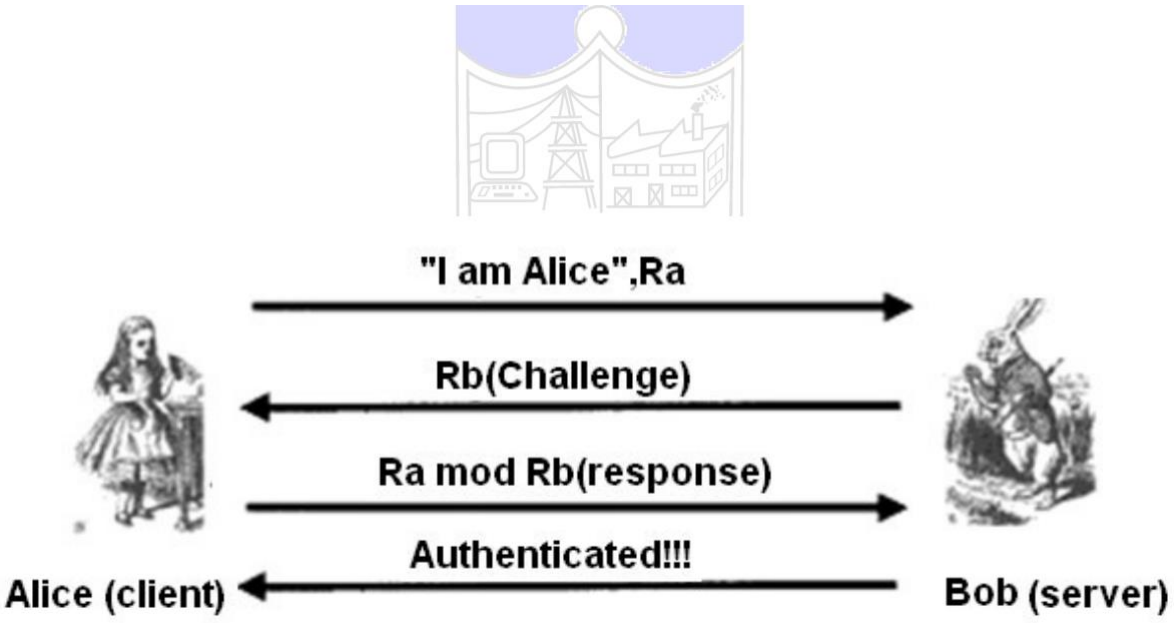


Figure 1 - Challange-Response Protocol

Assume there are malicious eaves-droppers who can hear the communication that is taking place. A simple authentication method is as follows: The server generates a random 3-digit number RAND and sends it to the client. The client computes the remainder(PSWD mod RAND) and sends the result to the server. The server also computes the value (PSWD mod RAND) and if it gets the same result, it concludes that the client has the correct password and authenticates the client as shown in figure 1.

Procedure / Approach /Algorithm / Activity Diagram:

Refere to the VLAB of EXPT NO. 6 simulation (https://cse29-iiith.vlabs.ac.in/exp/diffie-hellman/simulation.html) and implement the above authentication method shown in the figure 1 in the similar way.

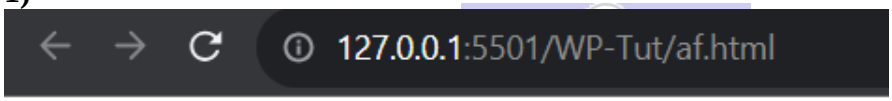
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Authentication</title>
</head>
<body>
    <script>
        function authenticate() {
            // Get the values entered by the user
            var password = document.getElementById('password').value;
            var rand = Math.floor(Math.random() * 900) + 100; // Generate a
random 3-digit number

            // Calculate PSWD mod RAND
            var result = password % rand;

            // Display the random number and the result
            document.getElementById('rand').textContent = "Random Number
(RAND): " + rand;
            document.getElementById('result').textContent = "Result (PSWD mod
RAND): " + result;

            // Simulate server-side verification
            var serverResult = password % rand;
            if (result === serverResult) {
                document.getElementById('message').textContent =
"Authentication Successful!";
            } else {
```

```
                document.getElementById('message').textContent =
"Authentication Failed!";
            }
        }
    </script>
    <h1>Password Authentication</h1>
    <p>Enter your 5-digit password:</p>
    <input type="number" id="password" maxlength="5">
    <button onclick="authenticate()">Authenticate</button>
    <div id="rand"></div>
    <div id="result"></div>
    <div id="message"></div>


</body>
</html>
```

Output:

1)

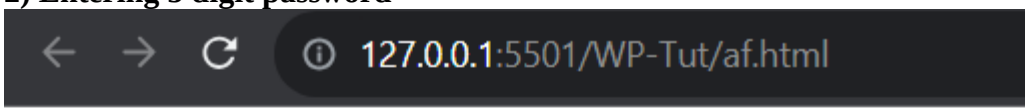


Password Authentication

Enter your 5-digit password:

Authenticate

2) Entering 5 digit password

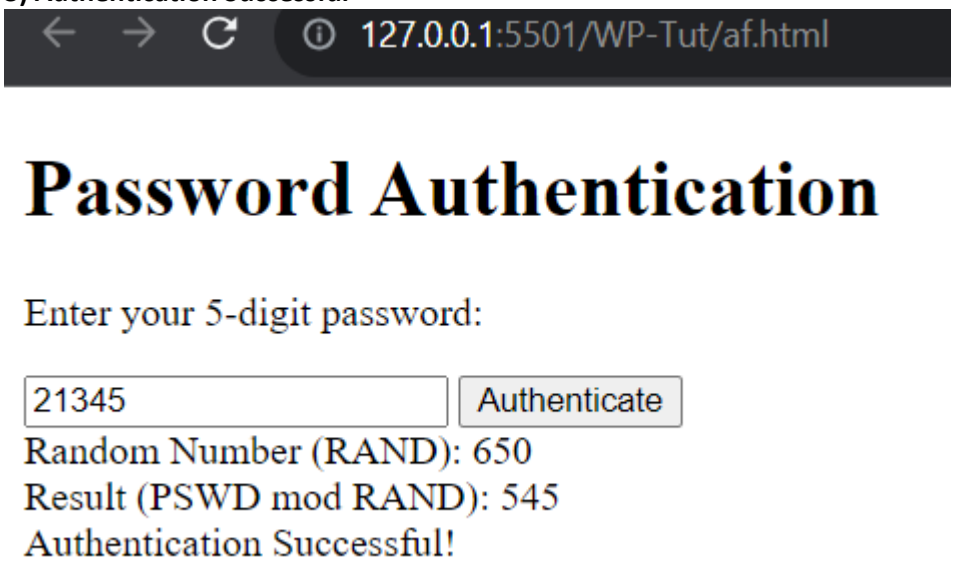


Password Authentication

Enter your 5-digit password:

21345 Authenticate

3) Authentication Successful



Questions:

1. **What are the advantages and disadvantages of the above authentication method?**

- **Advantages of Challenge-Response Protocols:**

1) Enhanced Security: They provide high security by requiring a unique response to a challenge, making it hard for attackers to impersonate users.

2) Resist Replay Attacks: Designed to prevent attackers from reusing captured responses through unique challenges.

3) Reduced Password Exposure: Passwords are not sent over the network, reducing the risk of interception.

4)2FA Support: Can be part of multi-factor authentication, adding another layer of security.

- **Disadvantages of Challenge-Response Protocols:**

1) Complex Implementation: More complex than passwords, requiring careful management of challenges and responses.

2) Key Management: Cryptographic keys can be challenging to manage, especially in large systems.

3) User Experience: May be less user-friendly, requiring extra steps or devices.

4) Lockout Risk: Incorrect implementation can lock out legitimate users if they lose tokens or face issues.

5) Costly: Some methods can be expensive to implement and maintain.

6) Compatibility Issues: Different systems may use incompatible Challenge-Response methods.

7) Vulnerabilities: Poorly implemented protocols can have security flaws that attackers can exploit.

2. Explain replay attack on this protocol?

- A replay attack in a Challenge-Response Protocol involves an attacker intercepting a valid user's response to a unique challenge and later replaying this response to trick the server into granting unauthorized access. It works like this:

1) Challenge: Server sends a unique challenge.

2) Interception: Attacker captures challenge and valid response.

3) Replay: Attacker resends the captured response.

4) Server Verification: Server, unaware it's a replayed response, grants access, thinking it's from the legitimate user.

To prevent replay attacks, use measures like timestamps, nonces, session tokens, or cryptographic techniques to ensure responses are valid only for specific challenges and expire.

1) Timestamps: Use time limits to make responses valid only briefly.

2) Nonce Values: Employ unique random values for challenges, rendering replays ineffective.

3) Session Tokens: Use tokens for single sessions that expire after use.

4) Cryptographic Techniques: Apply cryptographic methods like digital signatures to link challenges and responses securely.

5) One-Time Passwords (OTP): Use OTPs for one-time use and expiration after use.

Outcomes: Describe various access control policies and models

---

Conclusion:

Thus, in this experiment we have learned to design and implement a VIAB for Challenge-Response protocol.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- Mark Stamp, “Information security Principles and Practice” Wiley.