Experiment No. 7

Title: Challenge-Response Protocol

Batch: A3 **Roll No.: 16010421059** **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).

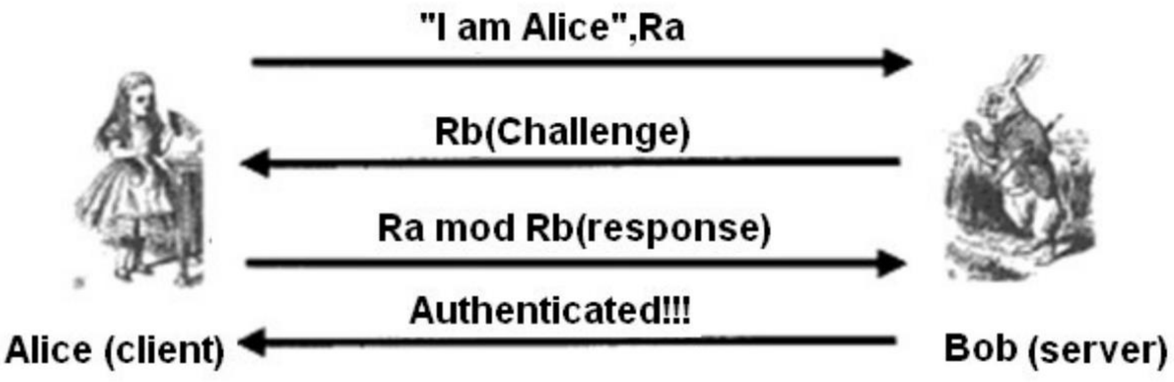
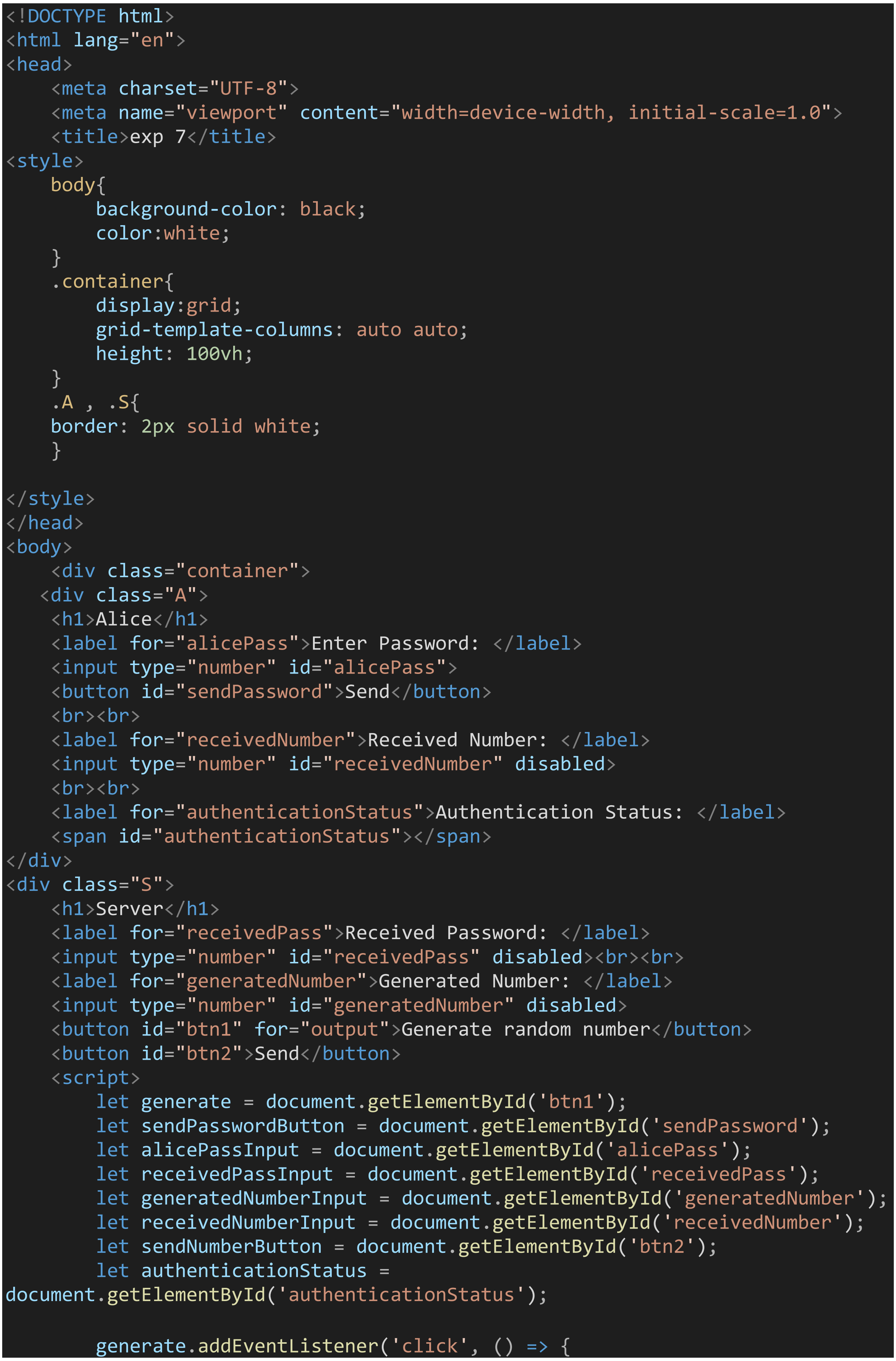


Figure 1 - Challange-Response Protocol

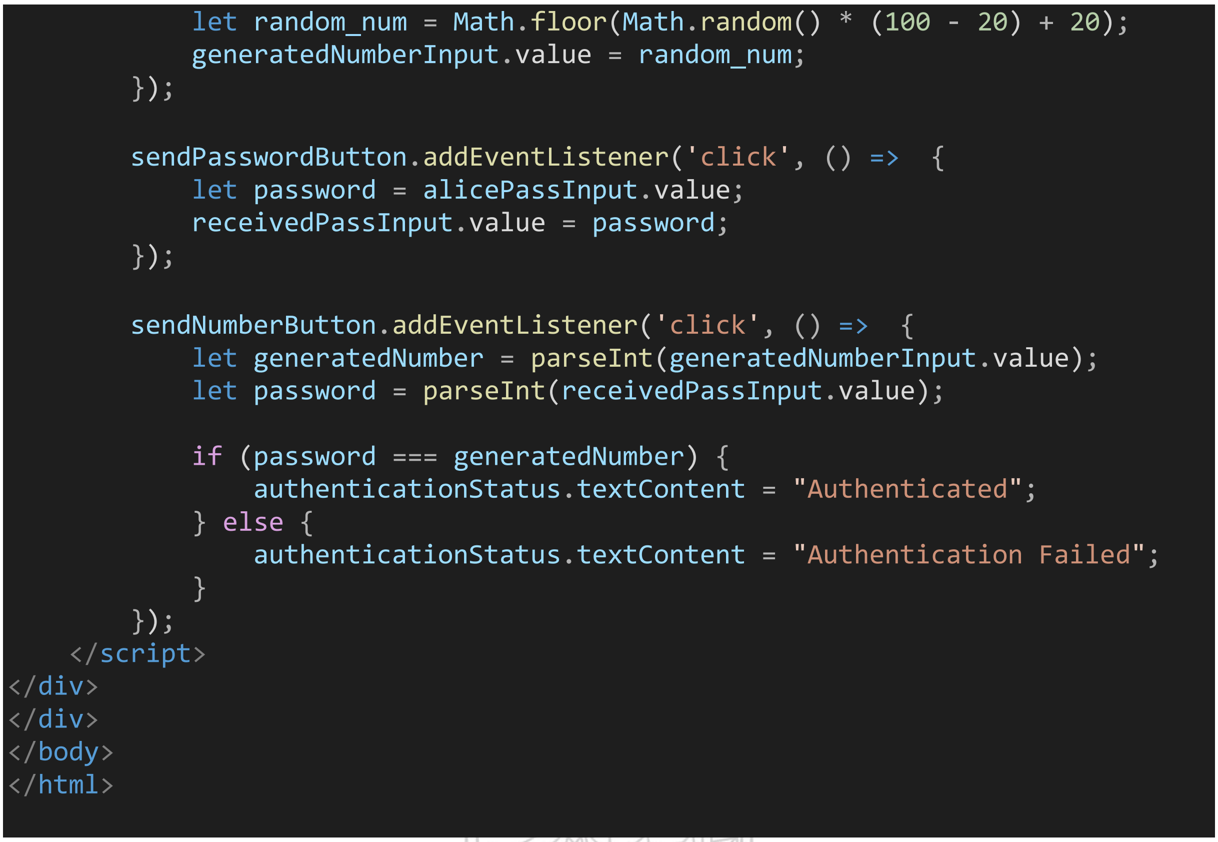
Assume there are malicious eaves-droppers who can hear the communication that is taking place. A simple authentication method is as follows: The server generates a random 3-digit number RAND and sends it to the client. The client computes the remainder(PSWD mod RAND) and sends the result to the server. 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.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>exp 7</title>
<style>
    body{
        background-color: black;
        color:white;
    }
    .container{
        display:grid;
        grid-template-columns: auto auto;
        height: 100vh;
    }
    .A , .S{
    border: 2px solid white;
    }

</style>
</head>
<body>
    <div class="container">
   <div class="A">
    <h1>Alice</h1>
    <label for="alicePass">Enter Password: </label>
    <input type="number" id="alicePass">
    <button id="sendPassword">Send</button>
    <br><br>
    <label for="receivedNumber">Received Number: </label>
    <input type="number" id="receivedNumber" disabled>
    <br><br>
    <label for="authenticationStatus">Authentication Status: </label>
    <span id="authenticationStatus"></span>
</div>
<div class="S">
    <h1>Server</h1>
    <label for="receivedPass">Received Password: </label>
    <input type="number" id="receivedPass" disabled><br><br>
    <label for="generatedNumber">Generated Number: </label>
    <input type="number" id="generatedNumber" disabled>
    <button id="btn1" for="output">Generate random number</button>
    <button id="btn2">Send</button>
    <script>
        let generate = document.getElementById('btn1');
        let sendPasswordButton = document.getElementById('sendPassword');
        let alicePassInput = document.getElementById('alicePass');
        let receivedPassInput = document.getElementById('receivedPass');
        let generatedNumberInput = document.getElementById('generatedNumber');
        let receivedNumberInput = document.getElementById('receivedNumber');
        let sendNumberButton = document.getElementById('btn2');
        let authenticationStatus =
document.getElementById('authenticationStatus');

        generate.addEventListener('click', () => {
```



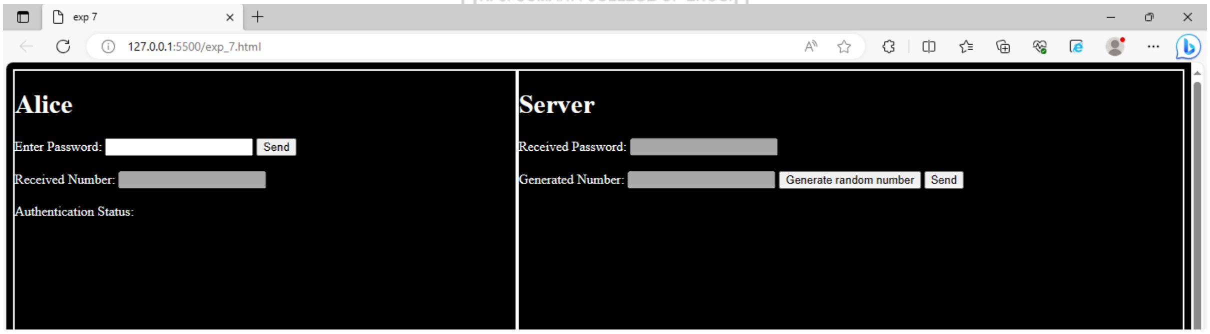
```
            let random_num = Math.floor(Math.random() * (100 - 20) + 20);
            generatedNumberInput.value = random_num;
        });

        sendPasswordButton.addEventListener('click', () =>  {
            let password = alicePassInput.value;
            receivedPassInput.value = password;
        });

        sendNumberButton.addEventListener('click', () =>  {
            let generatedNumber = parseInt(generatedNumberInput.value);
            let password = parseInt(receivedPassInput.value);

            if (password === generatedNumber) {
                authenticationStatus.textContent = "Authenticated";
            } else {
                authenticationStatus.textContent = "Authentication Failed";
            }
        });
    </script>
</div>
</div>
</body>
</html>
```

OUTPUT:



Alice

Enter Password: Send

Received Number:

Authentication Status:

Server

Received Password:

Generated Number: Generate random number Send

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.

Questions:

1. **Advantages and Disadvantages of Challenge-Response Protocol Authentication:**

Advantages:

a. Resistance to Eavesdropping: Challenge-response protocols are designed to resist eavesdropping attacks. An eavesdropper who intercepts the challenge and response will not have enough information to gain access to the system or authenticate themselves.

b. Enhanced Security: These protocols can provide an additional layer of security by requiring the responder (usually the user or device) to prove their identity through a dynamic response based on a unique challenge.

c. Protection Against Stolen Credentials: Challenge-response authentication is effective against attacks involving stolen credentials (e.g., passwords) since an attacker would also need to respond to the challenge correctly.

d. Resistance to Replay Attacks: When implemented correctly, challenge-response protocols can resist replay attacks by including nonces (unique numbers used only once) in the challenge.

Disadvantages:

a. Complexity: Implementing and managing challenge-response protocols can be complex and resource-intensive. Both the challenger and responder need to have access to shared secrets or keys, and key management can be challenging.

b. User Experience: Challenge-response can sometimes be less user-friendly than traditional username and password authentication methods, as users are required to respond to dynamic challenges.

c. Dependency on Secure Channels: These protocols often rely on secure communication channels to transmit challenges and responses. If the channel is compromised, the security of the protocol can be undermined.

d. Replay Attack Vulnerabilities: If not implemented properly, challenge-response protocols can be vulnerable to replay attacks. A replay attack occurs when an attacker intercepts and retransmits a valid response to a challenge. To mitigate this, nonces must be used to ensure that responses cannot be reused.

2. **Replay Attack on Challenge-Response Protocol:**

A replay attack is a form of network attack in which an attacker intercepts valid data transmission and retransmits it, typically to gain unauthorized access or deceive a system. In the context of a challenge-response authentication protocol, a replay attack occurs when an attacker captures a legitimate response to a challenge and then sends the same response again at a later time to gain unauthorized access.

To explain further:

1. The challenge-response protocol involves a challenge being sent by the challenger to the responder. The responder generates a unique response based on the challenge and a secret key.

2. If the protocol does not incorporate safeguards against replay attacks, an attacker can intercept this valid response during transmission.

3. The attacker stores the valid response and, at a later time, sends the same response to the system, mimicking the legitimate responder.

4. The system, unaware of the replay attack, accepts the response as valid and grants access to the attacker.

To mitigate replay attacks in a challenge-response protocol, the following measures are often used:

- Nonces (Number Used Once): Both the challenger and the responder use nonces, which are random or unique numbers, in the challenge and response. These nonces should be used only once and are included in the communication. This ensures that a response cannot be replayed because the nonce value will have changed in the next challenge.

- Timestamps: Timestamps can also be used to ensure that responses are not replayed. The system checks the timestamp to ensure that the response is valid within a reasonable time frame.

- Session Management: Implement session management and session-specific challenges and responses to prevent replay attacks within the context of a specific session.

By incorporating these safeguards, a challenge-response protocol can effectively resist replay attacks and enhance security.

Outcomes:

CO 3: Describe various access control policies and models.

Conclusion:

Designed and implemented a VLab for Challenge-Response protocol successfully.

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.