

8. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).

Code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diffie-Hellman Key Exchange</title>
5 </head>
6 <body>
7   <h1>Diffie-Hellman Key Exchange</h1>
8   <div>
9     <h2>Alice</h2>
10    <label for="alicePrivateKey">Alice's Private Key (a):</label>
11    <input type="text" id="alicePrivateKey" /><br>
12    <button onclick="generateAliceKeys()">Generate Alice's Public Key</button>
13    <div>
14      <label for="alicePublicKey">Alice's Public Key (A):</label>
15      <span id="alicePublicKey"></span>
16    </div>
17  </div>
18  <hr>
19  <div>
20    <h2>Bob</h2>
21    <label for="bobPrivateKey">Bob's Private Key (b):</label>
22    <input type="text" id="bobPrivateKey" /><br>
23    <button onclick="generateBobKeys()">Generate Bob's Public Key</button>
24    <div>
25      <label for="bobPublicKey">Bob's Public Key (B):</label>
26      <span id="bobPublicKey"></span>
27    </div>
28  </div>
29  <hr>
30  <div>
31    <h2>Shared Secret Key</h2>
32    <button onclick="deriveSharedSecret()">Derive Shared Secret Key</button>
33    <div>
34      <label for="sharedSecret">Shared Secret Key:</label>
35      <span id="sharedSecret"></span>
36    </div>
37  </div>
38
```

```
38 <script>
39   // Constants for the Diffie-Hellman calculation (usually prime numbers)
40   const p = BigInt(23); // Prime number
41   const g = BigInt(5); // Generator
42
43   let alicePrivateKey, alicePublicKey, bobPrivateKey, bobPublicKey, sharedSecret;
44   function generateRandomPrivateKey(): bigint {
45     function generateRandomPrivateKey() {
46       return BigInt(Math.floor(Math.random() * 10) + 1); // Generate a random private key (a or b)
47     }
48
49     function generateAliceKeys() {
50       alicePrivateKey = generateRandomPrivateKey();
51       alicePublicKey = (g ** alicePrivateKey) % p;
52       document.getElementById('alicePublicKey').innerText = alicePublicKey;
53     }
54
55     function generateBobKeys() {
56       bobPrivateKey = generateRandomPrivateKey();
57       bobPublicKey = (g ** bobPrivateKey) % p;
58       document.getElementById('bobPublicKey').innerText = bobPublicKey;
59     }
60
61     function deriveSharedSecret() {
62       sharedSecret = (bobPublicKey ** alicePrivateKey) % p;
63       document.getElementById('sharedSecret').innerText = sharedSecret;
64     }
65   }
66 </script>
67 </body>
68 </html>
69
```

Output:

Diffie-Hellman Key Exchange

File | E:/B.Tech/SEM-V/LABS/CISL/Codes/A8.html

Diffie-Hellman Key Exchange

Alice

Alice's Private Key (a):

Generate Alice's Public Key

Alice's Public Key (A):

Bob

Bob's Private Key (b):

Generate Bob's Public Key

Bob's Public Key (B):

Shared Secret Key

Derive Shared Secret Key

Shared Secret Key: