Experiment No. 5

Title: Vlab on Message Authentication Codes

Batch: B2 Roll No.: 16010421119
Experiment No.: 5

Title: Illustrate and implement message authentication code.

Resources needed: Windows/Linux OS

Theory:

In cryptography, a cipher block chaining message authentication code (CBC-MAC) is a technique for constructing a message authentication code (MAC) from a block cipher. The message is encrypted with some block cipher algorithm in cipher block chaining (CBC) mode to create a chain of blocks such that each block depends on the proper encryption of the previous block. This interdependence ensures that a change to any of the plaintext bits will cause the final encrypted block to change in a way that cannot be predicted or counteracted without knowing the key to the block cipher.

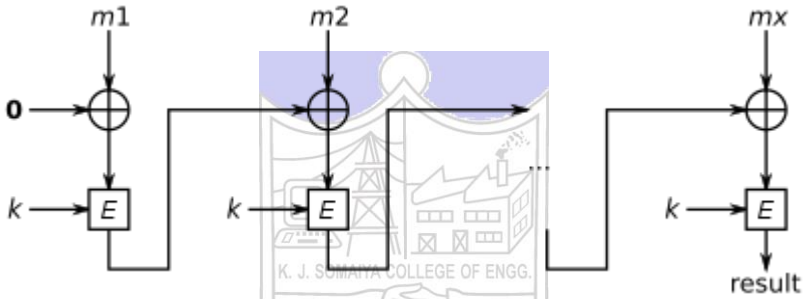


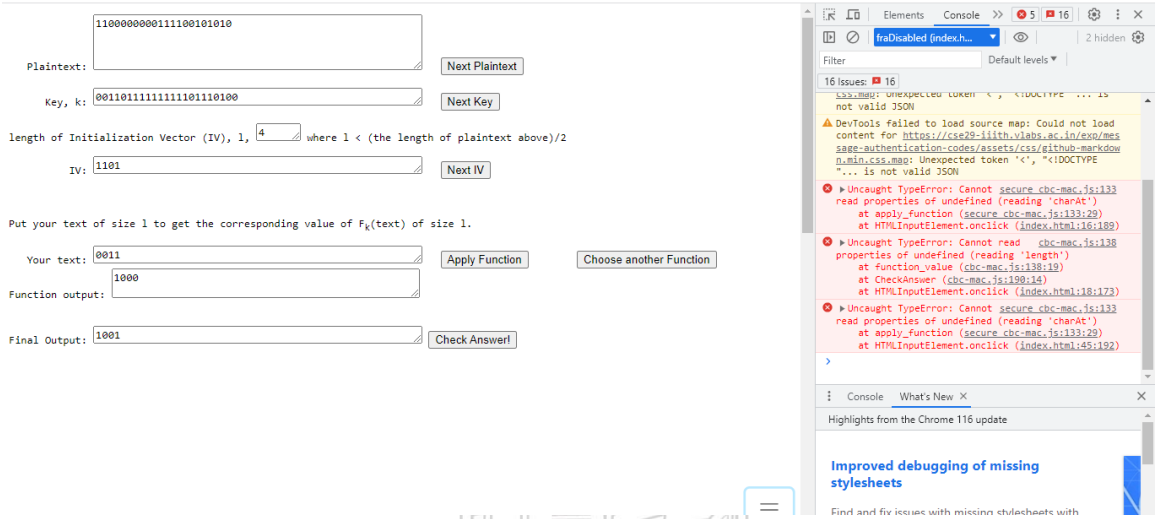
Figure 1 - CBC-MAC construction

To calculate the CBC-MAC of message m, one encrypts m in CBC mode with zero initialization vector(IV) and keeps the last block. The figure 1 sketches the computation of the CBC-MAC of message comprising blocks m1, m2,…mx using a secret key k and a block cipher E.

Activity :

1) Perform the Vlab on MAC - https://cse29-iiith.vlabs.ac.in/exp/message-authentication-codes/index.html

VLAB Outputs:



2) Implement the similar vlab simulation with a simple block cipher in CBC mode with following details-
   - Plain text message M = user's choice (string type)
   - Block Size = user's choice (must be < (length of M_2)/2)
   - Key k = user's choice (length of key is same as block size)
   - IV = user's choice (length of IV is same as block size)
   - E = XOR function

```
<!DOCTYPE html>

<html lang="en">


<head>

  <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Input Data Logger</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1>Input Data Logger</h1>

    <label for="input1">Input 1:</label>
    <input type="text" id="input1"><br>
    <button onclick="logInputData(1)">Log Input 1</button>
    <br><br>

    <label for="input2">Input 2:</label>
    <input type="text" id="input2"><br>
    <button onclick="logInputData(2)">Log Input 2</button>
    <br><br>

    <label for="input3">Input 3:</label>
    <input type="text" id="input3"><br>
    <button onclick="logInputData(3)">Log Input 3</button>
    <br><br>

    <label for="input4">Input 4:</label>
    <input type="text" id="input4"><br>
    <button onclick="logInputData(4)">Log Input 4</button>
    <br><br>

    <label for="input5">Input 5:</label>
    <input type="text" id="input5"><br>
```

```
    <button onclick="logInputData(5)">Log Input 5</button>
    <br><br>

    <label for="input6">Input 6:</label>
    <input type="text" id="input6"><br>
    <button onclick="logInputData(6)">Log Input 6</button>

    <label for="input7">Input 7:</label>
    <input type="text" id="input7"><br>
    <button onclick="logInputData(6)">Log Input 7</button>
    <script src="script.js"></script>
</body>

</html>
```



```
body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
}

h1 {
    text-align: center;
    margin-top: 20px;
}

label {
    display: block;
    margin-top: 10px;
```

```
}

input {
    width: 50%;
    padding: 8px;
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

button {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-top: 10px;
}

button:hover {
    background-color: #45a049;
}

function logInputData(buttonNumber) {
    const inputField = document.getElementById(`input${buttonNumber}`);
    const inputData = inputField.value;
    console.log(`Button ${buttonNumber} clicked. Input data: ${inputData}`);
}
```

Questions:

1) Compare MAC and cryptographic Hash functions.

MAC (Message Authentication Code) and cryptographic hash functions serve distinct purposes in the field of cryptography, but they both play essential roles in ensuring the integrity and authenticity of data. Here's a comparison of the two:

Purpose:

MAC: A MAC is primarily used for message authentication, ensuring that a message has not been tampered with during transit and verifying the identity of the sender. It provides both data integrity and authenticity.
Cryptographic Hash Function: Cryptographic hash functions are designed to take an input (message) and produce a fixed-length output (hash value) that is typically a digest of the message. While they can be used for various purposes, their primary role is data integrity verification. They provide data integrity but not authenticity.
Key Usage:

MAC: MACs require a secret key shared between the sender and receiver. This key is used to compute and verify the MAC value.
Cryptographic Hash Function: Cryptographic hash functions do not use keys. They are deterministic and produce the same hash value for the same input.
Output Size:

MAC: The output size of a MAC depends on the specific algorithm and key length used but is typically longer than hash functions. Common MACs, like HMAC, produce fixed-size outputs.
Cryptographic Hash Function: Cryptographic hash functions produce a fixed-length output, regardless of the input size.

Outcomes:

CO3: Describe various access control policies and models.

Conclusion:
We have learnt about Message Authentication Codes

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

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- William Stallings, “Cryptography and Network Security” by Pearson Education 4th Edition 2014.