**Experiment No. 5**

**Title:** Vlab on Message Authentication Codes

**Batch: A3 Roll No.: 16010421073 Experiment No.: 5**

**Title:** Illustrate andimplement message authentication code.

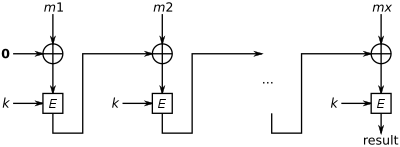
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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](https://en.wikipedia.org/wiki/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.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

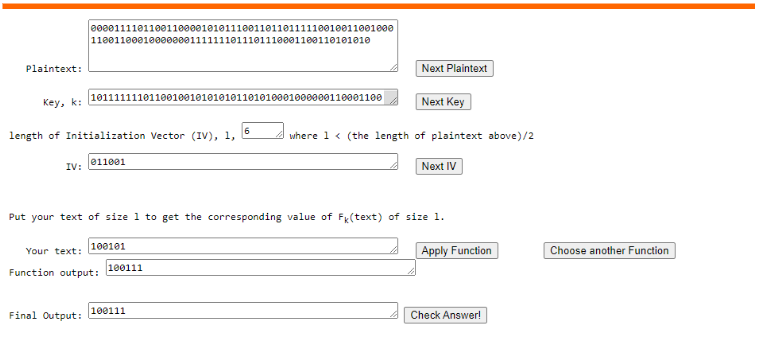
**Activity :**

1. Perform the Vlab on MAC - <https://cse29-iiith.vlabs.ac.in/exp/message-authentication-codes/index.html>
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 M2)/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

**Results:**

**Vlab Output:**

****

**HTML Code:**

body {

font-family: Arial, sans-serif;

text-align: center;

}

h1 {

color: #333;

}

.input-section {

margin-top: 20px;

border: 1px solid #ddd;

padding: 20px;

max-width: 400px;

margin: 0 auto;

background-color: #f9f9f9;

}

input[type="text"], input[type="number"] {

width: 100%;

padding: 10px;

margin: 5px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

background-color: #007bff;

color: #fff;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.output-section {

margin-top: 20px;

}

#encrypted-message {

font-weight: bold;

}

CSS CODE

body {

    font-family: Arial, sans-serif;

    text-align: center;

}

h1 {

    color: #333;

}

.input-section {

    margin-top: 20px;

    border: 1px solid #ddd;

    padding: 20px;

    max-width: 400px;

    margin: 0 auto;

    background-color: #f9f9f9;

}

input[type="text"], input[type="number"] {

    width: 100%;

    padding: 10px;

    margin: 5px 0;

    border: 1px solid #ccc;

    border-radius: 4px;

}

button {

    background-color: #007bff;

    color: #fff;

    padding: 10px 20px;

    border: none;

    border-radius: 4px;

    cursor: pointer;

}

button:hover {

    background-color: #0056b3;

}

.output-section {

    margin-top: 20px;

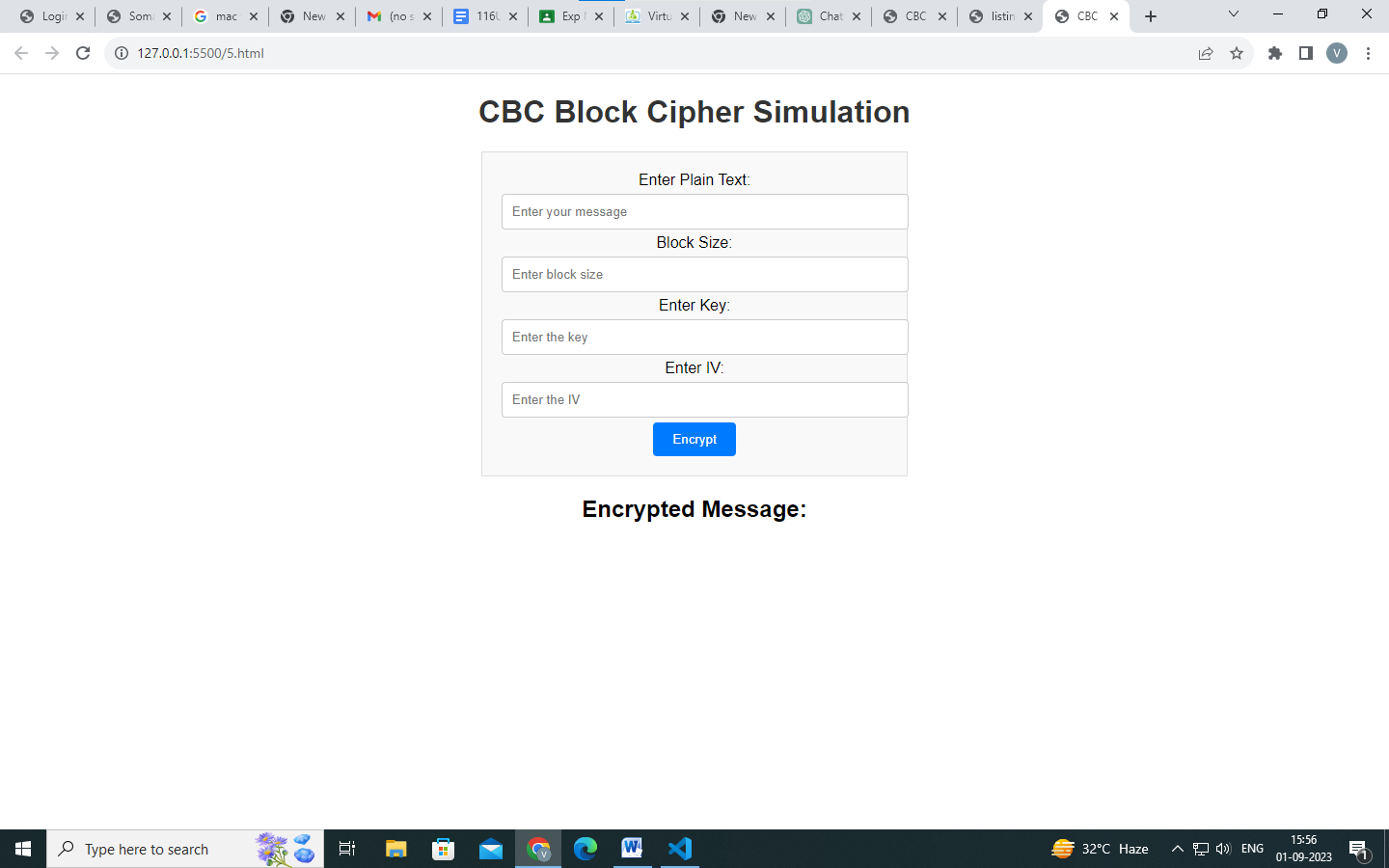
}

#encrypted-message {

    font-weight: bold;

}

**Output:**



\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A white background with black text

Description automatically generated**

**Questions:**

1. **Compare MAC and cryptographic Hash functions.**

**Ans:**

MAC (Message Authentication Code) and cryptographic hash functions are both used in cryptography, but they serve different purposes and have distinct characteristics. Here's a comparison of the two:

**1. Purpose:**

**- MAC (Message Authentication Code):** MAC is primarily used for ensuring the integrity and authenticity of a message. It is used to verify that a message has not been tampered with and that it was indeed generated by a specific sender.

**- Cryptographic Hash Function:** Cryptographic hash functions are used to transform data into a fixed-size output (hash) in a way that is difficult to reverse. They are commonly used for data integrity checks, digital signatures, and password storage.

**2. Key Usage:**

**- MAC:** MAC requires a secret key to generate and verify the authentication code. Both the sender and the receiver must possess the same secret key.

**- Cryptographic Hash Function:** Hash functions do not require a key; they produce a hash based solely on the input data. Hash functions are one-way functions, meaning you can't reverse them to obtain the original data.

**3. Verification:**

**- MAC:** Verification of a MAC involves re-computing the MAC on the received data using the same key and comparing it to the received MAC. If they match, the message is considered authentic.

**- Cryptographic Hash Function:** Hashes can't be directly verified in the same way as MACs. They are typically used for comparing the hash of received data to a known, trusted hash value to check for tampering.

**4. Collision Resistance:**

**- MAC:** MACs are designed to be collision-resistant, which means it should be computationally infeasible for an attacker to find two different messages that produce the same MAC with the same key.

**- Cryptographic Hash Function:** Cryptographic hash functions are also designed to be collision-resistant, meaning it should be computationally difficult to find two different inputs that produce the same hash value.

**5. Output Length:**

**- MAC:** The length of a MAC depends on the specific algorithm and key size. It can be variable.

**- Cryptographic Hash Function:** The output length of a cryptographic hash function is fixed. For example, SHA-256 always produces a 256-bit hash.

**6. Examples:**

**- MAC:** HMAC (Hash-based Message Authentication Code) is a widely used MAC algorithm.

**- Cryptographic Hash Function:** Examples include SHA-256, SHA-3, and MD5 (though MD5 is not recommended for security-sensitive applications due to vulnerabilities).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:**

**CO 2** Illustrate different cryptographic algorithms for security.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion:**

Thus, we have successfully completed the implementation of MAC and also made a simple GUI to describe it.

**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.