



Experiment No. 5

Title: Vlab on Hash Function and its Applications



Batch: B-2

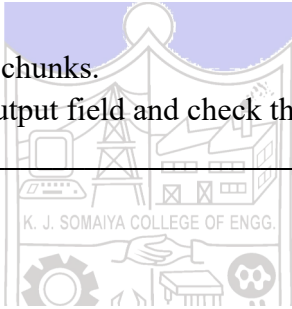
Roll No.: 16010422234

Experiment No.: 8

Activity :

- 1) Perform the Vlab on <https://cse29-iiith.vlabs.ac.in/exp/hash-functions/>
- 2) Implement the similar vlab simulation with a simple block cipher in CBC mode with following details-
 - Select a plaintext for which the HMAC tag is to be computed.(by clicking on NextPlaintext Button)
 - For simplicity fix $l=8$ which is default, but it should be $l < (\text{length of plaintext})/4$.
 - Select an Initialization Vector, IV of length l . by clicking on "Next IV" button
 - Divide generated plaintext 'm' into say 'k' chunks of 8 bits and kth chunk will have bits less than 8, to make it 8-bits by padding zeros at end
 - Compute $z_0 = \text{"IV"} \parallel (\text{k XOR ipad})$ manually where \parallel implies concatenation and enter z_0 in "Your text" field to get z_1 (hash of z_0)
 - Compute $z_1' = \text{"z1"} \parallel m_1$ manually where \parallel implies concatenation and enter z_1' in "Your text" field to get z_2
 - Continue this for all message chunks.
 - Enter Final Z^k into the final output field and check the correctness of the message.

Activity:


HMAC Construction using a "Dummy" Hash Function

HMAC construction

Plaintext:

length of Initialization Vector (IV), l ,

IV:

Key, k :

ipad: 0x5C (01011100)
opad: 0x36 (00110110)

Put your text of size 21 to get the corresponding value of hash of size 1.

Your text:

Hashed value:

Final Output:

Your answer is correct!

Questions:

1) Compare Message Authentication code and cryptographic Hash functions.

Message Authentication Code (MAC):

- **Key-Based:** Requires a secret key to generate the tag.
- **Purpose:** Ensures both message integrity and authentication (verifies the sender's identity).
- **Security:** Depends on the key and the algorithm used.
- **Examples:** HMAC, CMAC.

Cryptographic Hash Functions:

- **Keyless:** Does not require a secret key.
- **Purpose:** Provides message integrity by generating a fixed-size hash.
- **Security:** Depends on resistance to collisions (hard to find two messages with the same hash).
- **Examples:** SHA-256, MD5.

Outcomes: Illustrate different cryptographic algorithms for security

Conclusion:

This experiment provided a hands-on understanding of cryptographic hash functions and Message Authentication Codes (MACs). We learned that while both are used to ensure data integrity, MACs also offer authentication through a secret key, whereas cryptographic hash functions are keyless. Cryptographic hash functions are fundamental in securing digital communications, including password protection, digital signatures, and blockchain verification. The activity demonstrated how these cryptographic techniques are essential for maintaining data security and trust in digital systems.
