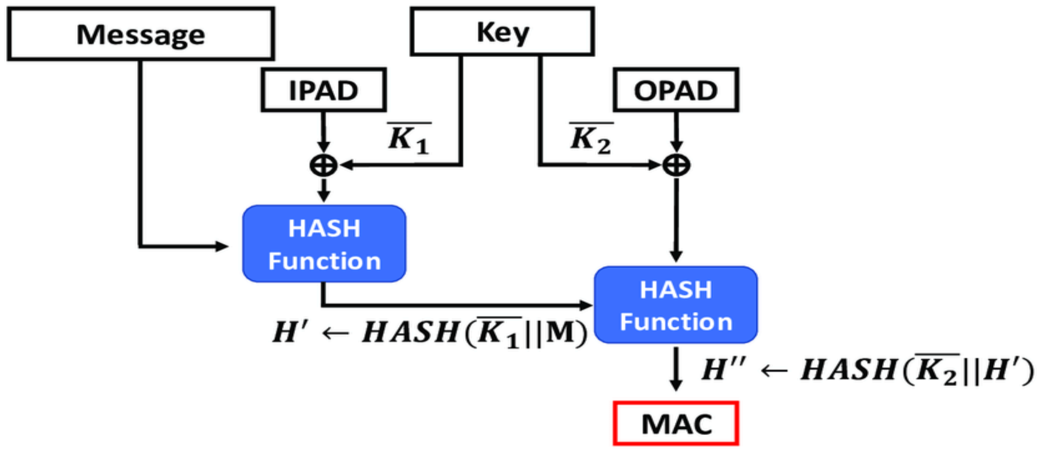
EXPERIMENT NO: -5

Aim: - Cryptographic Hash Functions and Applications (HMAC): to understand the need, design and applications of collision resistant hash functions.

Theory: -

- **Hash Authentication**: - Hash-based Message Authentication Code (HMAC) is a message authentication code that uses a cryptographic key in conjunction with a hash function. Hash-based message authentication code (HMAC) provides the server and the client each with a private key that is known only to that specific server and that specific client. In authentication systems, when users create a new account and input their chosen password, the application code passes that password through a hashing function and stores the result in the database.

- **HMAC Authentication**: -HMAC stands for Keyed-Hashing for Message Authentication. It's a message authentication code obtained by running a cryptographic hash function (like MD5, SHA1, and SHA256) over the data (to be authenticated) and a shared secret key. HMACs are almost similar to digital signature. They both enforce integrity and authenticity. They both use cryptography keys. And they both employ hash functions. The main difference is that digital signatures use asymmetric keys, while HMACs use symmetric keys.

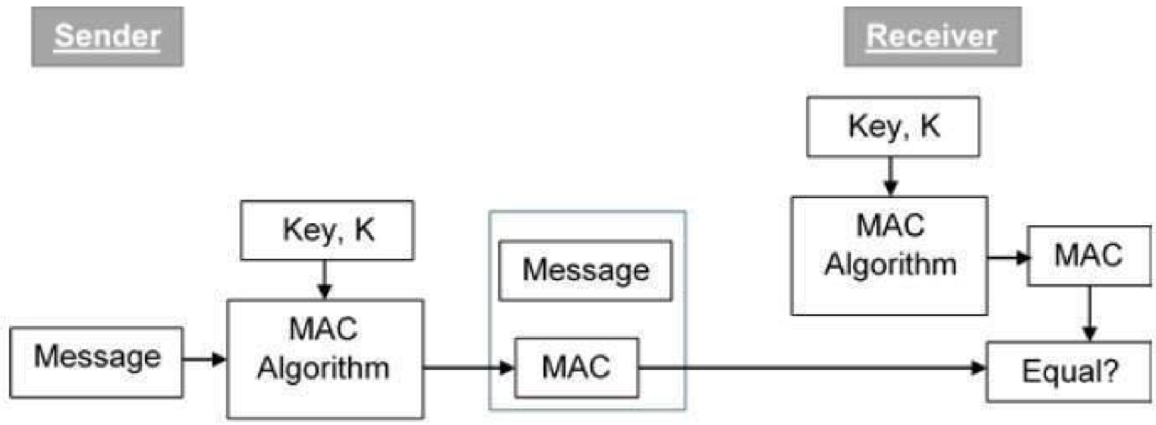


Message
Key
IPAD
OPAD
$\overline{K_1}$
$\overline{K_2}$
HASH Function
HASH Function
$H' \leftarrow HASH(\overline{K_1}||M)$
$H'' \leftarrow HASH(\overline{K_2}||H')$
MAC

Application Of HMAC: - The HMAC can be applied in a number of scenarios, for example:

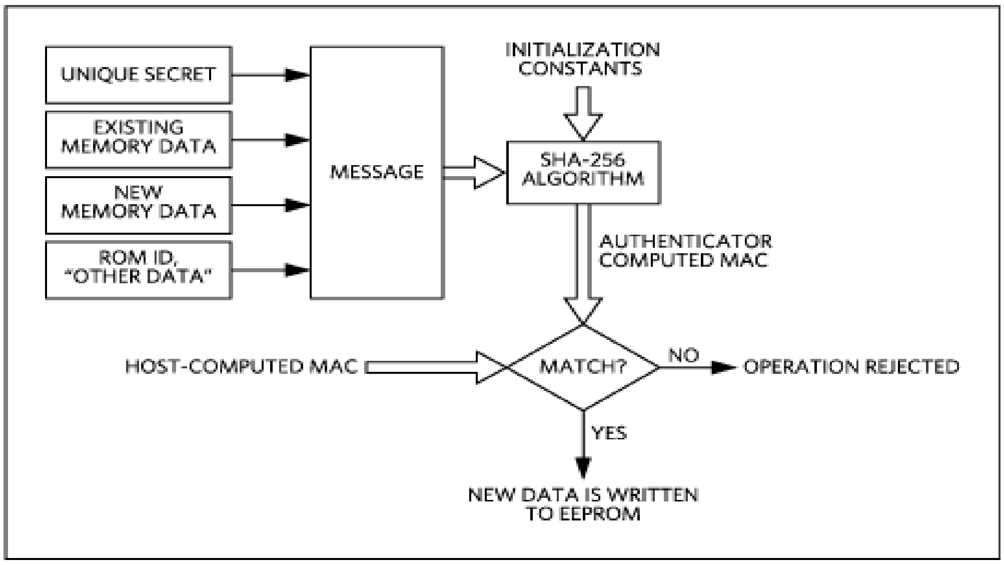
- Sending an e-mail with a password reset link that is valid only for a certain time and can only be used once. The HMAC allows for this without any added server state.
- Verifying e-mail address in order to create or activate an account
- Authenticating form data that has been sent to the users’ web browser and then posted back
- **MAC Authentication and its Application**: -Essentially, a MAC is an **encrypted checksum** generated on the underlying message that is sent along with a message to ensure message authentication. The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value. In cryptography, a message authentication code (MAC), sometimes known as a tag, is **a short piece of information used to authenticate a message**—in other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed. In this model of MAC, sender encrypts the content before sending it through network for confidentiality. Thus this model provides confidentiality as well as authentication. For cases when there is an alteration in message, we decrypt it for waste, to overcome that problem, we opt for external error code.

- **SHA Authentication**: -In cryptography, SHA-1 (Secure Hash Algorithm 1) is a



cryptographic hash function which takes an input and produces a 160-bit (20- byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. **SHA-1 hash** is used to authenticate messages sent between the client and server during the TLS handshake.SHA-512 is the current strongest encryption algorithm.
SHA-256, SHA-384, and SHA-512 (sometimes grouped under the name SHA-
2) are variants of SHA-1 and use longer message digests. The Junos OS supports the SHA- 256 version of SHA-2, which can process all versions of Advanced

Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) encryption.



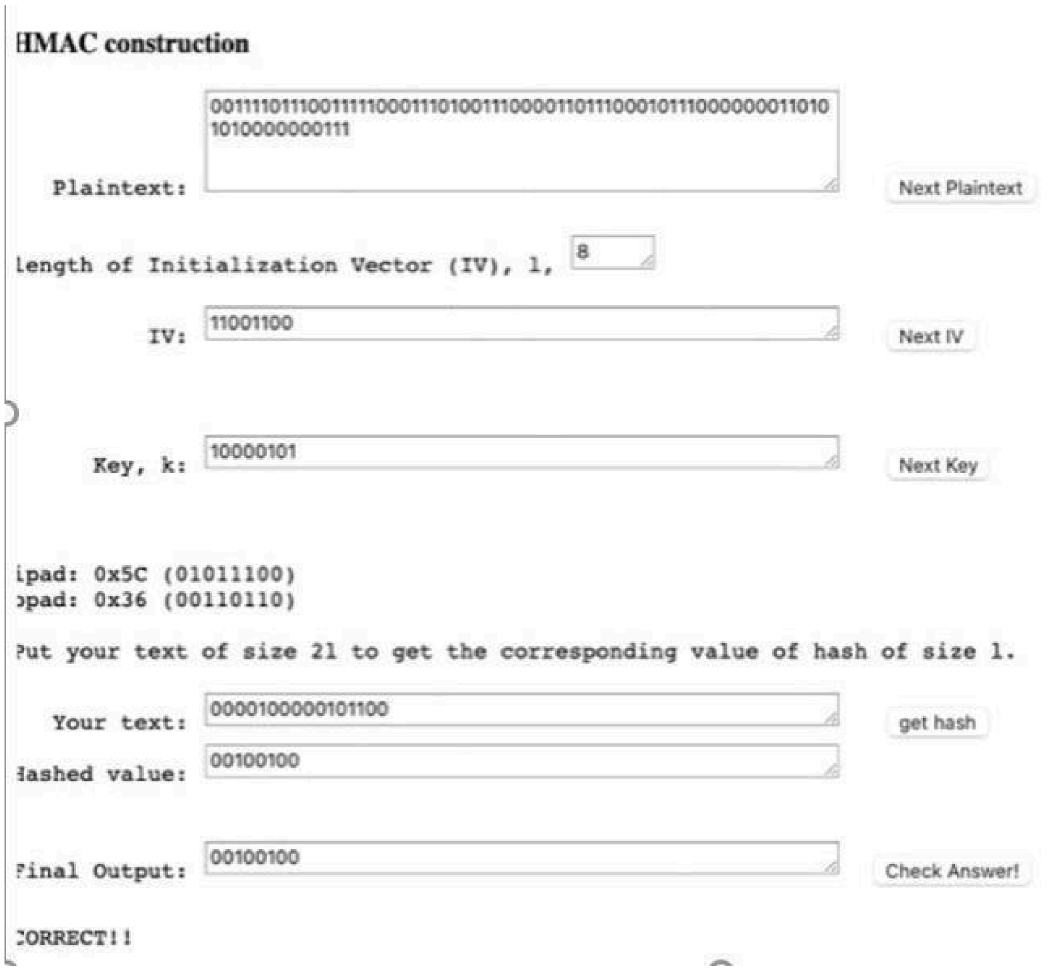
- **MD File:** -md file stands for "**Markdown documentation**". md file is a plain- text document that contains no other elements. The text can be formatted using special inline text symbols.
  Markdown can be used for everything. People use it to create websites, documents, notes, books, presentations, email messages, and technical documentation. Markdown is portable. Files containing Markdown-formatted text can be opened using virtually any application. Markdown is a lightweight markup language, and each MD file is written in a particular "flavor" of Markdown. It can be helpful to think of the different flavors as being like different dialects of one language.
- **Application Of MD File: -**md file you can use **any text editor**. There are plugins for many editors (for example Atom, Emacs, Sublime Text, Vim, and Visual Studio Code) that allow you to preview Markdown while you are editing it. You can also use a dedicated Markdown editor like Stack Edit or Dillinger.

STEP:-

1. Familiarize yourself with the working of SHA-1. Though we would be using a dummy hash in the sequel for simplicity, in general, you could be using SHA-1 instead
2. Select a plaintext for which the HMAC tag is to be computed.(by clicking on NextPalintext Button)
3. For simplicity fix l=8 which is default,but it should be l < (length of plaintext)/4.
4. Select an Initialization Vector, IV of length l.by clicking on "Next IV" button)
5. Use the ipad and opad as described in theory part to compute the ciphertext with the help of the hash function provided to you.
6. 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

7. Compute z0="IV||(k XOR ipad)" manually where || impies concatenation and enter z0 in "Your text" field to get z1
8. Compute z1="z0||m1" manually where || impies concatenation and enter z1 in "Your text" field to get z2
9. Repeat above step and finally compute z(k+1)="zk||L" where L=|m|,make L 8-bits by padding zeros to left of it
10. Compute p="IV||(k XOR opad)" manually where || impies concatenation and enter p in "Your text" field to get q
11. Compute r="q||z(k+1)" manually where || impies concatenation and enter 'r' in "Your text" field to get final HMAC tag 't'
12. Notice that z0,z1,z2,............zk,z(k+1),p,r are all of size '2l'(=16 in our case as l=8).
13. Write the final cipher text 't' in 'Final Output' field and check your answer

EXAMPLE:



HMAC construction

Plaintext: 0011110111001111100011101001110000110111000101110000000011010
101000000111

Length of Initialization Vector (IV), l, 8

IV: 11001100

Key, k: 10000101

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

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

Your text: 0000100000101100

Hashed value: 00100100

Final Output: 00100100

CORRECT!!

Conclusion: -In the above experiment we learnt about the application and usage of HASH, HMAC and MAC Authentication. Basically, a cryptographic hash function is an algorithm that can be run on data such as an individual file or a password to produce a value called a checksum. The main use of a cryptographic hash function is to verify the authenticity of a piece of data. Cryptographic hash functions are widely used in cryptocurrencies to pass transaction information anonymously.