*Function* HMAC

*Input Arguments*

* Msg- plain text in binary
* k- k is the key in binary
* n is the length of prime P in binary
* iv- n length bits of all 0s
* ipad-0x5c
* opad-0x36
* here iv, ipad, opad is a fixed constant
* any value of iv with length equal to length of P should be taken

*Output*

HMAC in binary string

*Logic*

1. This function internally uses fixed\_hash and var\_hash
2. The key k is prefixed with trailing 0s till its length is equal to n (k<P)
3. The values ipad, opad are repeated till their length is equal to n
4. Each message block is substring of length n
5. Ipad, opad are XORed with k and the respective k\_ipad and k\_opad are obtained.
6. K\_ipad, iv is given as input to fixed\_hash and hashed\_k\_ipad is formed.
7. K\_opad, iv is also given as input to fixed\_hash and hashed\_k\_opad is formed.
8. Now, the values msg and hashed\_k\_ipad is given to var\_ash function
9. The output of the above function along with hased\_k\_opad is given to fixed\_hash function
10. The output after calling the fixed\_hash function, is the required HMAC

*Usage*

This is used to create CCA secure encryption