# MAC

```python
import hmac
import hashlib

def generate_mac(key,message):
  return
hmac.new(key.encode(),message.encode(),hashlib.sha256).hexdigest()

#sender
message = "Nanda Krishnan V"
key = "DUK"
mac = generate_mac(key,message)
print(f"Generated MAC{mac}")

#reciever
recieved_msg = "Nanda Krishnan V"
recieved_mac = mac
validate_mac = generate_mac(key,recieved_msg)

if validate_mac == recieved_mac:
  print("MAC is valid, Confidentiality not affected")
else:
  print("MAC is invalid, Confidentiality is affected")
```

```
Generated MAC3c27a37a86adde6d1462f1b1af3fc22faff73afdd5912782e0eda331ce2559fd
MAC is valid, Confidentiality not affected
```

*Submitted by Nanda Krishnan V, Msc Cyber Security*

```python
import hmac
import hashlib

def generate_mac(key,message):
  return
hmac.new(key.encode(),message.encode(),hashlib.sha256).hexdigest()

#sender
message = "Nanda Krishnan V"
key = "DUK"
mac = generate_mac(key,message)
print(f"Generated MAC{mac}")

#reciever
recieved_msg = "NandaKrishnan V"
recieved_mac = mac
validate_mac = generate_mac(key,recieved_msg)

if validate_mac == recieved_mac:
  print("MAC is valid,Confidentiality not affected")
else:
  print("MAC is invalid, Confidentiality is affected")
```

```
Generated MAC3c27a37a86adde6d1462f1b1af3fc22faff73afdd5912782e0eda331ce2559fd
MAC is invalid, Confidentiality is affected
```

```python
import hmac
import hashlib
from cryptography.fernet import Fernet

def generate_mac(key,message):
  return
hmac.new(key.encode(),message.encode(),hashlib.sha256).hexdigest()


encryption_key = Fernet.generate_key()
cipher_suite = Fernet(encryption_key)

#sender
message = "Nanda Krishnan V"
key = "DUK"
mac = generate_mac(key,message)
message_with_mac = message + mac
#print(f"Generated MAC{mac}")

encrypted_msg = cipher_suite.encrypt(message_with_mac.encode())
print(f"Encrypted Message: {encrypted_msg.decode()}")

#reciever
decrypted_msg_with_mac = cipher_suite.decrypt(encrypted_msg).decode()
recieved_msg = decrypted_msg_with_mac[:-64]
recieved_mac = decrypted_msg_with_mac[-64:]
validate_mac = generate_mac(key,recieved_msg)

if validate_mac == recieved_mac:
  print(f"MAC is valid, encrypted message: {recieved_msg}")
else:
  print("MAC is invalid, Confidentiality is affected")
```

```
Encrypted Message: gAAAAABnRFM5oAWYDJ_qEQX5MOQhL-iu3g_ORq23bLZmaTT4Rs-2pn99S-HYolb9WUXxIXFIFdhIw0u7TXI6PGnVcC2D0KrF-cjnTgh6zFUFIsbBSQpSFjit0iI8NdHUeGq9MfdDPYMqJ5fhmAUaQi
MAC is valid, encrypted message: Nanda Krishnan V
```

***Submitted by Nanda Krishnan V, Msc Cyber Security***