11- RSA Encryption Implementation

```python
import rsa
def loadKeys():
    with open('pk.pem', 'rb') as p:
        pk = rsa.PublicKey.load_pkcs1(p.read())
    with open('sk.pem', 'rb') as p:
        sk = rsa.PrivateKey.load_pkcs1(p.read())
    return pk, sk

def encrypt(message, key):
    return rsa.encrypt(message.encode('ascii'), key)

def decrypt(ciphertext, key):
  try:
      return rsa.decrypt(ciphertext, key).decode('ascii')
  except:
      return False
```

13- AES Encryption Implementation

```python
import base64
from Crypto.Cipher import AES

def encryption(pk,message):
    message =  padding(message)
    aes = AES.new(sk.encode("utf-8"), AES.MODE_ECB)
    ciphertext = aes.encrypt(message.encode('utf-8'))
    return str(base64.b64encode(ciphertext), encoding='utf-8')

def decrypt(sk=None, ciphertext):
    aes = AES.new(sk.encode('utf-8'), AES.MODE_ECB)
    ciphertext = base64.b64decode(ciphertext)
    message = aes.decrypt(ciphertext).decode('utf-8')
    return messagess
```