SECURITY\_MEASURES\_IN\_IIOT(4)

#pip install cryptography

import hashlib # For hashing to ensure data integrity

from cryptography.fernet import Fernet # For encryption

import random # For simulating sensor data

# Generate an encryption key

key = Fernet.generate\_key()

cipher = Fernet(key)

def get\_sensor\_data():

return random.uniform(50, 100) # Random temperature between 50°C and 100°C

# Encrypt the sensor data

def encrypt\_data(data, cipher):

return cipher.encrypt(data.encode())

# Decrypt the sensor data

def decrypt\_data(encrypted\_data, cipher):

return cipher.decrypt(encrypted\_data).decode()

# Verify data integrity using hashing

def verify\_data\_integrity(data):

return hashlib.sha256(data.encode()).hexdigest()

sensor\_data = f"Temperature: {get\_sensor\_data():.2f}°C"

print("Original Data:", sensor\_data)

# Encrypt and then decrypt the data

encrypted\_data = encrypt\_data(sensor\_data, cipher)

print("Encrypted Data:", encrypted\_data)

decrypted\_data = decrypt\_data(encrypted\_data, cipher)

print("Decrypted Data:", decrypted\_data)

original\_hash = verify\_data\_integrity(sensor\_data)

decrypted\_hash = verify\_data\_integrity(decrypted\_data)

if original\_hash == decrypted\_hash:

print("Data Integrity Verified: Hashes match.")

else:

print("Data Integrity Issue: Hashes do not match.")