EXPERIMENT NO: 1D

Conduct an experiment to encrypt and decrypt given sensitive data.

Aim:

To securely **encrypt and decrypt a sensitive message** (like a password) using symmetric encryption with Fernet from the cryptography library.

Algorithm:

1. **Generate a key using Fernet.generate_key().**
2. **Create a Fernet cipher with the generated key.**
3. **Convert the data (string) to bytes.**
4. **Encrypt the data using cipher.encrypt().**
5. **Decrypt the data using cipher.decrypt().**
6. **Convert bytes back to string to view the original message.**

Program:

```
[5]: from cryptography.fernet import Fernet
     key = Fernet.generate_key()
     print("Generated Key:", key.decode())

     Generated Key: FeC5LisCxE6ccLsHYpbDYsujt1Rr5XxGFLd9aaheo4E=

[6]: cipher = Fernet(key)
     data = "My bank account password is 1234".encode()
     print("\nOriginal Data:", data.decode())

     Original Data: My bank account password is 1234

[7]: encrypted_data = cipher.encrypt(data)
     print("Encrypted Data:", encrypted_data.decode())

     Encrypted Data: gAAAAABo6kT8215UKLv27YBvLjNrir6qSKoq2b1AyepvZmcs719vcpXzNarqcNfE4fjUVMvjK9GgeVDH59fOpupGcgf4DUhBkbT6KYEmchOMM-F4H9MhTdYC4UAyvqc1pxsKTTrw7
     mQH

[8]: decrypted_data = cipher.decrypt(encrypted_data)
     print("Decrypted Data:", decrypted_data.decode())

     Decrypted Data: My bank account password is 1234
```

Result:

The program aims to securely encrypt and decrypt a message using Fernet symmetric encryption. It generates a key, converts the message into bytes, encrypts it into unreadable data, and then decrypts it back to the original message. The result shows that the original message is safely restored after encryption.