Text File Encryption Using Fernet

Information Security IA 1

Faculty: Shivani Deosthale

TY Comps C-1

Rhea Nair: 16010122316

Shubham Malgaonkar: 16010122331

Vedansh Savla: 16010122323

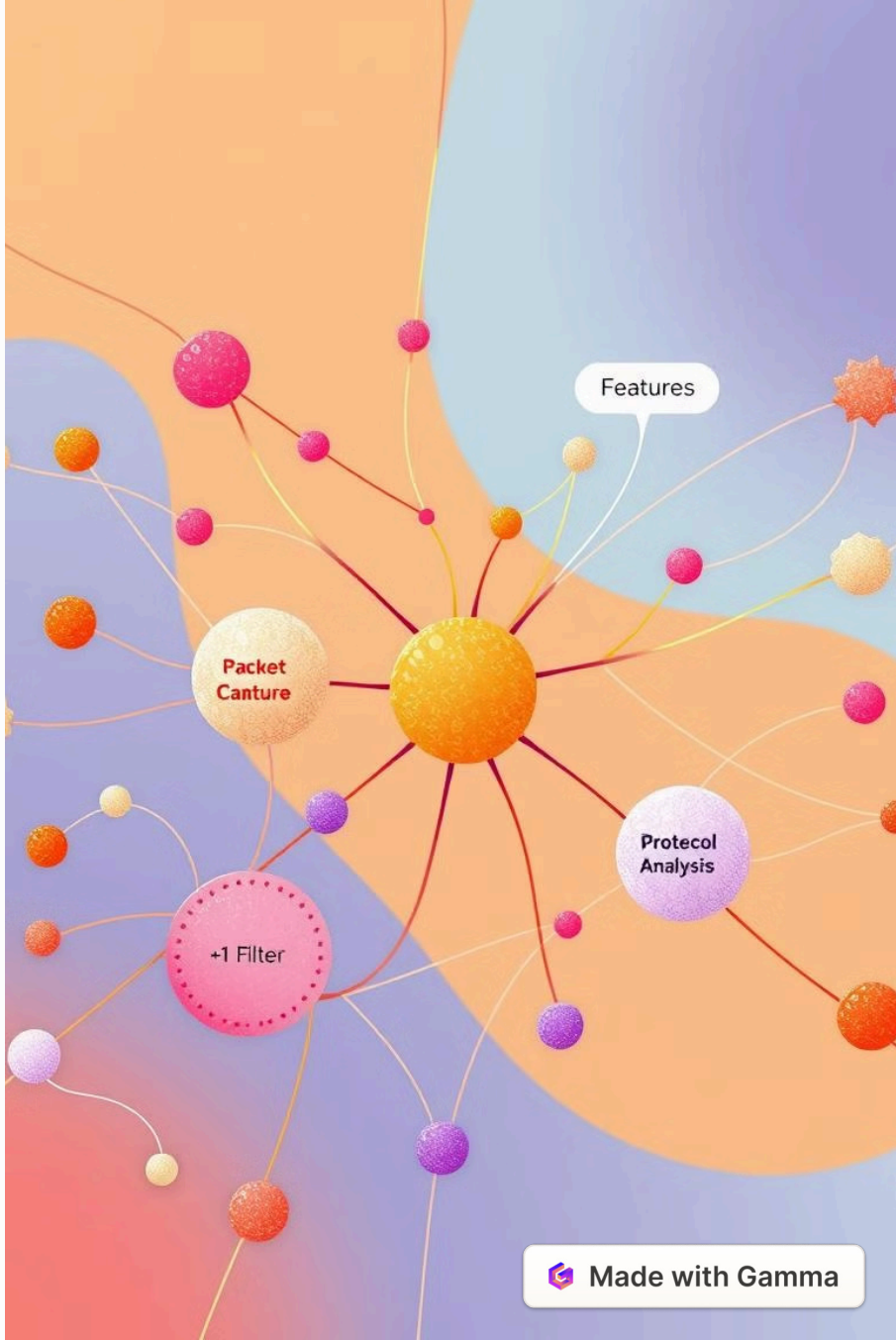
What is Fernet Module of the Cryptography Library?

Fernet is a symmetric encryption module within the Python Cryptography library that ensures secure data encryption and decryption. It employs AES-128 encryption in CBC mode with PKCS7 padding and includes an HMAC for authentication, guaranteeing the integrity and confidentiality of encrypted messages.

Fernet simplifies encryption for developers by providing an easy-to-use interface for encrypting and decrypting sensitive data without requiring in-depth cryptographic knowledge.

Key Features of Fernet

1. Symmetric encryption using AES-128
2. Secure key management with URL-safe base64 encoding
3. Message authentication to prevent tampering
4. Easy-to-use API for encrypting and decrypting data
5. Time-based token expiration for enhanced security



Limitations of Fernet

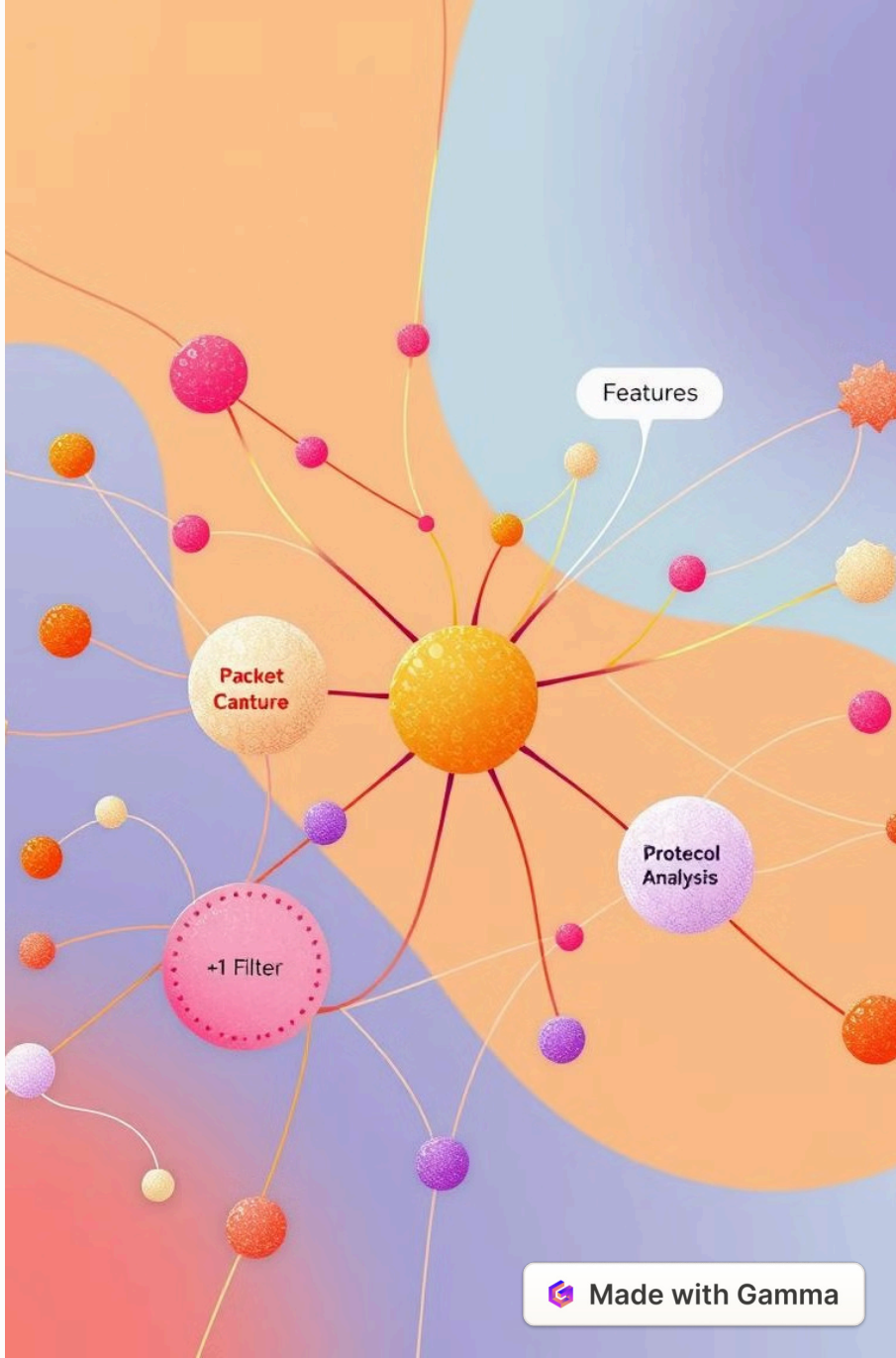
1. **Fixed Key Size**: Fernet uses a 256-bit key, which may not meet the needs of more security-demanding applications.

2. **Limited Encryption Algorithms**: Only AES in CBC mode with PKCS7 padding is supported, limiting flexibility.

3. **No Key Revocation**: Once a key is exposed, it can't be revoked, posing a security risk.

4. **Performance Issues**: Not optimized for encrypting large files or datasets efficiently.

5. **Symmetric Encryption Only**: Uses the same key for both encryption and decryption, increasing vulnerability if the key is compromised.



Implementation details

```python
main.py > main
import encryptFile
import decryptFile
import generateAKey
import os

username = "abcd"
password = "kjsce"

def login():
    user = input("Username = ").strip()
    passw = input("Password = ").strip()

    if user == username and passw == password:
        return 1
    else:
        return 0

def main():
    while True:
        try:
            choice = int(input("\n1. Login\n0. Exit\nEnter choice = "))
        except ValueError:
            print("Invalid input! Please enter 1 to Login or 0 to Exit.")
            continue  # Retry input

        if choice == 1:
            if login():
                generateAKey.gen_key()
                encryptFile.encrypt()
                decryptFile.decrypt()
            else:
                print("\nInvalid login")
        elif choice == 0:
            print("Thank you!")
            os.remove("encrypted_file.txt")
            os.remove("SecretKey.key")
            break

if __name__ == "__main__":
    main()
```

main.py

```python
generateAKey.py > ...
from cryptography.fernet import Fernet

def gen_key():
    key = Fernet.generate_key()
    with open('SecretKey.key', 'wb') as file:
        file.write(key)
```

generatAKey.py

```
toBeSecret.txt

Shubham Malgaonkar
Rhea Nair
Vedansh Savla
```

toBeSecret.k

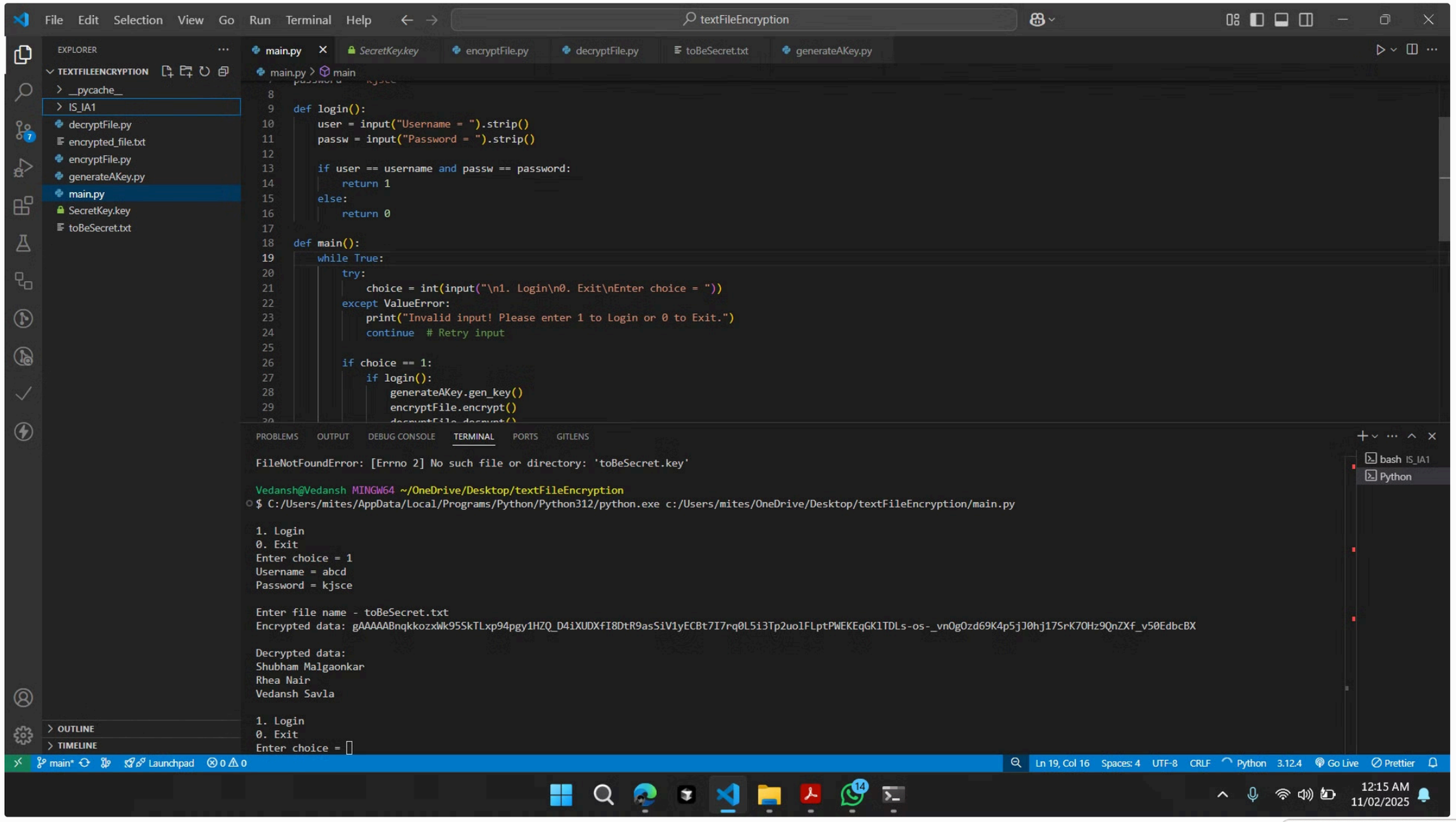
```python
decryptFile.py > ...
def decrypt():
    with open('SecretKey.key', 'rb') as file:
        key = file.read()

    with open('encrypted_file.txt', 'rb') as file:
        encryptedData = file.read()

    from cryptography.fernet import Fernet
    f = Fernet(key)
    decryptedData = f.decrypt(encryptedData)

    print('Encrypted data:', encryptedData.decode())
    print()
    print('Decrypted data:', decryptedData.decode())
```

decryptFile.py

```python
encryptFile.py > ...
def encrypt():
    with open('SecretKey.key', 'rb') as file:
        key = file.read()

    filename = input("\nEnter file name - ").strip()
    with open(filename, 'rb') as file:
        data = file.read()

    from cryptography.fernet import Fernet
    f = Fernet(key)
    encryptedData = f.encrypt(data)

    with open('encrypted_file.txt', 'wb') as file:
        file.write(encryptedData)
```

encryptFile.py

Output:



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Vedansh@Vedansh MINGW64 ~/OneDrive/Desktop/textFileEncryption
$ C:/Users/mites/AppData/Local/Programs/Python/Python312/python.exe c:/Users/mites/OneDrive/Desktop/textFileEncryption/main.py

1. Login
0. Exit
Enter choice = 1
Username = abcd
Password = kjsce

Enter file name - toBeSecret.txt
Encrypted data: gAAAAABnqkfeQHvchZv7aa3St-P6Xc2IbqthP9xX0em_TKlh5AE14uLyuZAg-zajLaEwYnQQ5OuX_vH2ecC12MxLRBAq4DNTM2x9qBQvnM28ID-diFRTUg6fSJRyffpfRO09w_SKgc5l

Decrypted data: Shubham Malgaonkar
Rhea Nair
Vedansh Savla

1. Login
0. Exit
Enter choice =
```

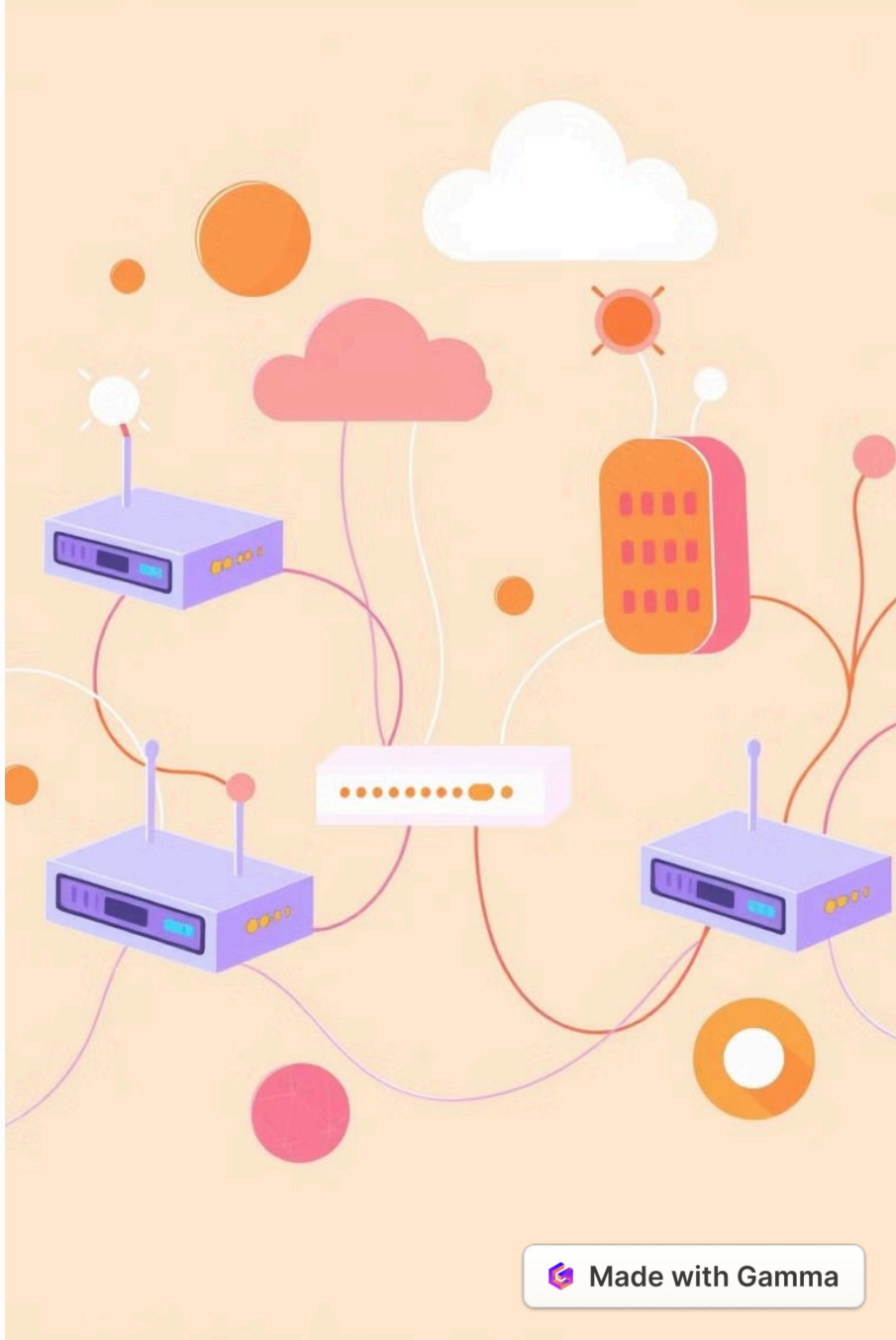
Use Cases for Fernet

Data Confidentiality

Compliance

Protection Against Data Breaches

Securing file storage and transmission



Best Practices for Using Text File Encryption

Keep the encryption key secure and never hard-code it in source files

Rotate encryption keys periodically for enhanced security

Use Fernet's token expiration feature to limit message validity.

Encrypt only sensitive data to avoid unnecessary overhead.

Error Handling in Encryption and Decryption

Encryption Errors:

- **Invalid Key**: Ensure the key used for encryption matches the one used for decryption. Handle exceptions when a mismatched key is provided.
- **Insufficient Permissions**: If the user doesn't have permission to access the file being encrypted, throw an appropriate error.

Decryption Errors:

- **Incorrect Key**: Handle cases where the wrong key is used for decryption by catching exceptions and notifying the user of the error.
- **Data Integrity**: If the encrypted file is altered (corrupted) before decryption, check for integrity errors and throw an exception.

Conclusion: The Role of Fernet in Secure Data Encryption

Text file encryption using Fernet offers a straightforward and secure method to protect sensitive information. By ensuring proper key management and adhering to best practices, developers can effectively safeguard the confidentiality, integrity, and authenticity of data. Implementing encryption not only mitigates the risk of unauthorized access but also helps meet compliance requirements and secure data storage. With continuous advancements in encryption technologies and security protocols, adopting strong encryption practices like Fernet is essential for maintaining data privacy in an increasingly digital world.

Thank You