Unit 4 Seminar 2: Encryption methods

Learning Outcomes

- Identify and manage security risks as part of a software development project.
- Critically analyse development problems and determine appropriate methodologies, tools and techniques (including program design and development) to solve them.
- Design and develop/adapt computer programs and to produce a solution that meets the design brief and critically evaluate solutions that are produced.

Using a symmetric block cipher AES-128 in Python

Writing a key

encrypt.py - C:\Users\jj_ca\OneDrive\Desktop\encrypt.py (3.9.5)

File Edit Format Run Options Window Help

```
from cryptography.fernet import Fernet
key = Fernet.generate_key()
print (key)

file = open ('key.key', 'wb')
file.write(key)
file.close()
```

IDLE Shell 3.9.5

File Edit Shell Debug Options Window Help

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\jj_ca\OneDrive\Desktop\encrypt.py =============
b'nTiqtI8t07c2gjwh1lCjVG3lO8KODLGIuzQC6XXXx5o='
```

Check to read the key

encrypt.py - C:\Users\jj_ca\OneDrive\Desktop\encrypt.py (3.9.5)

File Edit Format Run Options Window Help

```
from cryptography.fernet import Fernet

file = open ('key.key', 'rb')
key = file.read()
file.close()
print (key)
```

```
>>>
============== RESTART: C:\Users\jj_ca\OneDrive\Desktop\encrypt.py =============
b'nTiqtI8t07c2gjwh1lCjVG3lO8KODLGIuzQC6XXXx5o='
>>>
```

File example to encrypt

jcssdunit4.txt - Notepad

File Edit View

```
Jonathan
Secure Software Development
Unit 4 Seminar
Cryptography exercise
```

Encrypting the file

*encrypt.py - C:\Users\jj_ca\OneDrive\Desktop\encrypt.py (3....

File Edit Format Run Options Window Help

```python
from cryptography.fernet import Fernet

file = open ('key.key', 'rb')
key = file.read()
file.close()

with open('jcssdunit4.txt', 'rb') as f:
    data = f.read()

fernet = Fernet(key)
encrypted = fernet.encrypt(data)

with open('jcssdunit4.txt.encrypted', 'wb') as f:
    f.write(encrypted)
```

After running

```
>>>
============== RESTART: C:\Users\jj_ca\OneDrive\Desktop\encrypt.py =============
>>> |
```

File is encrypted

jcssdunit4.txt.encrypted - Notepad

File Edit View

gAAAAABikGGWtJieJNQ-BRBx6DTSnGVnS-oqCVcrMibkQw2qXAqDjSDjTzNXCvAR74SxVB5VofO-f4yRl2vX5rxMoePub8Up7vX26Tk1IqW30tn3ZsEvRZLOwqJoHVRwRjQDtXfM0rcfT6MV3luTvp4dmkrpk6ClVb5WJE

Decrypt the file

File Edit Format Run Options Window Help

```python
from cryptography.fernet import Fernet

file = open ('key.key', 'rb')
key = file.read()
file.close()

with open('jcssdunit4.txt.encrypted', 'rb') as f:
    data = f.read()

fernet = Fernet(key)
encrypted = fernet.decrypt(data)

with open('jcssdunit4.txt.decrypted', 'wb') as f:
    f.write(encrypted)
```

Run

```
============== RESTART: C:\Users\jj_ca\OneDrive\Desktop\encrypt.py =============
>>>
```

jcssdunit4.txt.decrypted - Notepad

File Edit View

```
Jonathan
Secure Software Development
Unit 4 Seminar
Cryptography exercise
```

The original message of the file

jupyter Quit

Files Running Clusters Formgrader Assignments

Select items to perform actions on them. Upload New

0 /	Name	Last Modified	File size
img		18 hours ago	
SSD Unit 4 Seminar Encryption.ipynb	Running	10 minutes ago	2.7 kB
jcssdunit4.txt		an hour ago	76 B
jcssdunit4.txt.decrypted		14 minutes ago	76 B
jcssdunit4.txt.encrypted		15 minutes ago	184 B
key.key		10 minutes ago	44 B
README.md		10 minutes ago	981 B

- **Why did you select the algorithm you chose?**

Python module Fernet AES-128

Fernet is built on top of several standard cryptographic primitives. Specifically, it uses a 128-bit key for encryption and SHA256 for authentication.

Faster and more efficient whilst technically less secure, there is not much difference between 128 and 256. Base64 encoded and 32-byte key.

- **Would it meet the GDPR regulations? Justify your answer.**

AES 128 encryption meets the current standards of FIPS 140-2 and 197; however, personal data protection is limited. A password could be used to create the key and generate a salt before hashing to allow further protection. Article 32 (Intersoft Consulting, N.D.) provides more considerations for the authorised use of personal data, which can be provided by encryption at a low cost. (ICO, 2021)

You should ensure that any solution you implement meets current standards such as FIPS 140-2 and FIPS 197 (NIST, 2001). There are lists of approved algorithms (NIST, 2019) which are acceptable for use, and the symmetrical methods would justify that. FIPS compliance allows proper confidentiality, integrity and authenticity, which is vital for organisations that collect, share, store, transfer or disseminate sensitive data (Encryption Consulting, N.D.).

References

Encryption Consulting. (N.D.). What is FIPS? How do you become compliant with FIPS? Available from: https://www.encryptionconsulting.com/education-center/what-is-fips/ [Accessed 27 May 2022].

Ico. (2021). Encryption Available from: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/security/encryption/ [Accessed 27 May 2022].

Intersoft Consulting. (N.D.). GDPR. Available from: https://gdpr-info.eu/ [Accessed 22 January 2022].

Nist. (2001). FIPS 197 Advanced Encryption Standard (AES). Available from: https://csrc.nist.gov/publications/detail/fips/197/final [Accessed 27 May 2022].

Nist. (2019). Transitioning the use of cryptographic algorithms and key lengths. Available from: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf [Accessed 27 May 2022].