Unit 8: Cryptography and Its Use in Operating Systems

1. Describe some of the issues encountered with cryptographic libraries.

The science of cryptography has been developed with the main aim of securing confidential messages sent between two parties. Cryptography is considered the art and science of concealing messages, ensuring privacy, and maintaining security in communication (Tutorialspoint, N.D.).

Cryptography safeguards sensitive data through coded algorithms, hashes, and signatures, aiming to achieve four key objectives: confidentiality, integrity, authentication, and non-repudiation. Cryptography relies on advanced cryptographic algorithms such as encryption, digital signatures, and hash algorithms to achieve these goals (AWS, N.D.).

The implementation of blockchain technology relies heavily on cryptographic algorithms such as asymmetric encryption, digital signatures, and hash functions. These algorithms are crucial in ensuring data integrity is transmitted through the network. Additionally, these cryptographic methods enable the tracking of transactions and provide a layer of data secrecy, thereby ensuring enhanced security (Rasslan et al., 2023).

According to Blessing et al. (2021) analysis of cryptographic libraries, memory safety concerns appear to pose a greater security risk than cryptographic vulnerabilities. Additionally, Blessing et al. (2021) have discovered a strong correlation between the complexity of these libraries and their potential for insecurity, indicating that overly intricate codebases may carry significant risks. In contrast, non-cryptographic systems are less complex and produce fewer vulnerabilities than their cryptographic counterparts.

Blessing et al. (2021) study found that memory safety bugs pose a more significant threat to cryptographic libraries than cryptography bugs. The study highlights that memory-related errors account for 37.1% of vulnerabilities, while cryptographic issues represent 25.8%. Other sub-categories, such as exposure of sensitive information, improper input validation, and numeric errors, account for the remaining 27.9%. This study reinforces the importance of addressing memory safety concerns to improve the overall security of cryptographic libraries.

On the other hand, Hazhirpasand et al. (2021) highlight that it is not uncommon for developers to employ cryptographic APIs, resulting in frequent cryptographic errors improperly. Extensive research has pinpointed problematic areas, including flawed hashing, vulnerable algorithms, and inadequate comprehension of core principles. Hazhirpasand et al. (2021) investigation discovered that difficulties encountered during installation, compilation, and utilisation of varying versions of a crypto library can hinder developers.

2. Explain the pros and cons of using standard cryptographic libraries.

Cryptology is used daily, particularly in secure online transactions, communication, and B2B (business-to-business) deals. Consequently, numerous cryptographic libraries have been developed and are utilised globally in various programming languages (Tech-Faq, N.D.).

In the current era of global networks and digitised information, data is stored, processed, and transmitted through computer systems and open communication channels. This makes it vulnerable to malicious attacks aimed at stealing sensitive information or disrupting critical systems. Thankfully, modern cryptography offers reliable techniques to safeguard against such threats while allowing legitimate users to access data securely (Tutorialspoint, 2019). Incorporating standard cryptographic libraries can be beneficial in several ways, but it is essential to consider both the potential advantages and drawbacks before deciding:

Advantages of using Standard Cryptographic Libraries:

1. **Confidentiality** – Encryption can protect data and communications against unauthorised access and disclosure.
2. **Authentication** Information can be safeguarded against spoofing and forgeries using cryptographic techniques like MAC and digital signatures.
3. **Data Integrity** Cryptographic hash functions are essential in giving users confidence in the accuracy of their data.
4. **Non-repudiation** – The digital signature offers the non-repudiation service to protect against any disputes that might develop due to the sender's rejection of message transmission.

Drawbacks of using Standard Cryptographic Libraries:

1. Even an authorised user may find it challenging to access strongly encrypted, authenticated, and digitally signed information at a time when access is vital for decision-making. An intrusive party may attempt to assault the network or computer system and turn it off.
2. Cryptography cannot guarantee high availability, one of the core components of information security. Other defence strategies are required to counter dangers like denial-of-service attacks and total information system failure.
3. Selective access control, another essential information security requirement, cannot be met using cryptography. For the same, administrative rules and processes must be used.
4. Cryptography does not protect against the dangers and vulnerabilities that result from shoddy systems, protocols, and method design. These require correct design and construction of a defensive infrastructure to be fixed.
5. Cryptography is not free. Costs include both time and money:
   - The addition of cryptographic algorithms causes a delay in the information processing.
   - Public key infrastructure must be built up and maintained to employ public key cryptography, which needs a substantial financial investment.
6. The computational complexity of mathematical issues is the foundation for the security of cryptographic techniques. Any improvement in the mathematical solutions to these issues or in processing capacity can make a cryptographic method insecure.

3. Demonstrate the use of cryptographic libraries in a simple application.

Cryptography secures communication and data with math. From online banking, it's essential for modern society to ensure messaging. Python is perfect for cryptography projects with its versatile libraries and modules (Ghost, 2023).

As GeeksforGeeks (2020) explains, safeguarding data during transmission or storage on a computer is achieved through cryptography. Python's cryptography package provides robust support for encryption and decryption of information. The Fernet module generates keys and encrypts and decrypts data within this package. By utilising this package, encrypted data remains secure and cannot be accessed or tampered with without the corresponding key. The Fernet module features a generate_key() method, which creates a new key, and an encrypt() method, which converts data into a "Fernet token".

For instance, below is a simple code that uses cryptography to encrypt and decrypt a message based on examples of GeeksforGeeks (2020):

```
"""
Fernet module is imported from the
cryptography package
"""

from cryptography.fernet import Fernet

# key is generated
key = Fernet.generate_key()

# value of the key is assigned to a variable
f = Fernet(key)

# the plaintext is converted to ciphertext
token = f.encrypt(b"This is the password")

# display the ciphertext
print(token)

# decrypting the ciphertext
d = f.decrypt(token)

# display the plaintext and the decode() method
# converts it from byte to string
print(d.decode())
```

Output:

```
C:\Users\hcham\anaconda3\envs\pythonProject-SSD\python.exe
C:\Users\hcham\PycharmProjects\pythonProject-SSD\cryptographic.py
b'gAAAAABlKcRBVVpD57HJsVAPV0pDCHXmDQh7XZlQtPxlgPMfDh6u0dZZ7B_3F1WxBlJ3N0MosP
ngSkzdgwFGwYM1KRwycgiTas5doZ8_PQdfE6TXWmP-FCk='
This is the password

Process finished with exit code 0
```

References:

Tutorialspoint (N.D.). *Cryptography with Python - Quick Guide - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_quick_guide.htm.

AWS (N.D.). *What is Cryptography? - Cryptography Explained - AWS*. [online] Available at: https://aws.amazon.com/what-is/cryptography/.

Rasslan, M., Nasreldin, M.M., Abdelrahman, D.R., Aya Elshobaky & Aslan, H.K. (2023). Networking and cryptography library with a non-repudiation flavour for blockchain. *Journal of Computer Virology and Hacking Techniques*. doi:https://doi.org/10.1007/s11416-023-00482-1.

Blessing, J., Specter, M.A. & Weitzner, D.J. (2021). You Really Shouldn't Roll Your Own Crypto: An Empirical Study of Vulnerabilities in Cryptographic Libraries. *arXiv:2107.04940 [cs]*. [online] Available at: https://arxiv.org/abs/2107.04940.

Hazhirpasand, M., Nierstrasz, O. & Ghafari, M. (2021). *Dazed and Confused: What's Wrong with Crypto Libraries?* [online] Available at: https://arxiv.org/pdf/2111.01406.pdf#

Tutorialspoint (2019). *Cryptography Benefits & Drawbacks - Tutorialspoint*. [online] Tutorialspoint.com. Available at: https://www.tutorialspoint.com/cryptography/benefits_and_drawbacks.htm.

Tech-Faq, (N.D.). *Cryptographic Libraries*. [online] Available at: https://www.tech-faq.com/cryptographic-libraries.html.

Ghost, P. (2023). *Using Python for Cryptography: An Introduction*. [online] Medium. Available at: https://professorghost.medium.com/using-python-for-cryptography-an-introduction-8299cb60a830

GeeksforGeeks. (2020). *Fernet (symmetric encryption) using Cryptography module in Python*. [online] Available at: https://www.geeksforgeeks.org/fernet-symmetric-encryption-using-cryptography-module-in-python/.