

Lab 5: Using Encryption to Enhance Confidentiality and Integrity

Osamudiamen Eweka

Cyb-605-Z2 Principles of Cybersecurity

Utica University

Introduction

In the conducted lab report, the participant delves into the integral role of encryption techniques in enhancing the security measures of digital communication. By engaging in a comprehensive exploration of both symmetric and asymmetric encryption methodologies, the report not only illuminates the theoretical underpinnings of these practices but also showcases practical implementation using encryption tools like Kleopatra and OpenSSL.

In exploring the crucial role of encryption techniques for enhancing the security of digital communications, the text delves into symmetric and asymmetric key ciphers, illustrating their significance in establishing secure communication channels. Symmetric key ciphers, while efficient for certain scenarios, face scalability and key distribution challenges, whereas asymmetric key ciphers, introduced by Whitfield Diffie and Martin Hellman, revolutionize secure communications by enabling secure exchanges over public channels without pre-shared keys. The practical use of encryption tools like OpenSSL and the management of cryptographic keys underscore the indispensable role of encryption in safeguarding data integrity and confidentiality in the digital realm (Kim, 2021). The hands-on experience gained from generating, exchanging, and managing cryptographic keys serves as a testament to the pivotal importance of encryption in maintaining data confidentiality and integrity. This exercise, as detailed in the report, not only broadens the participant's understanding of cryptographic principles but also underscores the necessity of employing advanced security protocols to navigate the complexities of modern digital environments effectively.

Objective

The main purpose of this lab report is to critically evaluate the effectiveness of symmetric and asymmetric encryption methods in enhancing digital communication security. By utilizing encryption tools such as Kleopatra and OpenSSL for practical exercises, the report aims to illustrate the significance of encryption in ensuring the confidentiality and integrity of information. Through this hands-on approach, the participant seeks to deepen their understanding of cryptographic principles and their application in real-world scenarios, thereby equipping themselves with the necessary skills to implement comprehensive security measures in the digital domain.

Lab Setup

The lab setup for enhancing confidentiality and integrity through encryption involved both hardware and software components. On the hardware side, the lab required a standard computer system capable of running the necessary software applications without specific requirements for high-performance components. The software setup was crucial and included Kleopatra (part of the GPG4Win suite) and OpenSSL. These tools were utilized for key generation, encryption, decryption, and digital signing tasks. The lab exercises were designed to run on commonly available operating systems, emphasizing accessibility and practicality in a typical educational or training environment. This combination of hardware and software facilitated a hands-on approach to understanding and implementing encryption techniques in real-world scenarios.

Section 1: Hands-On Demonstration

Part 1:

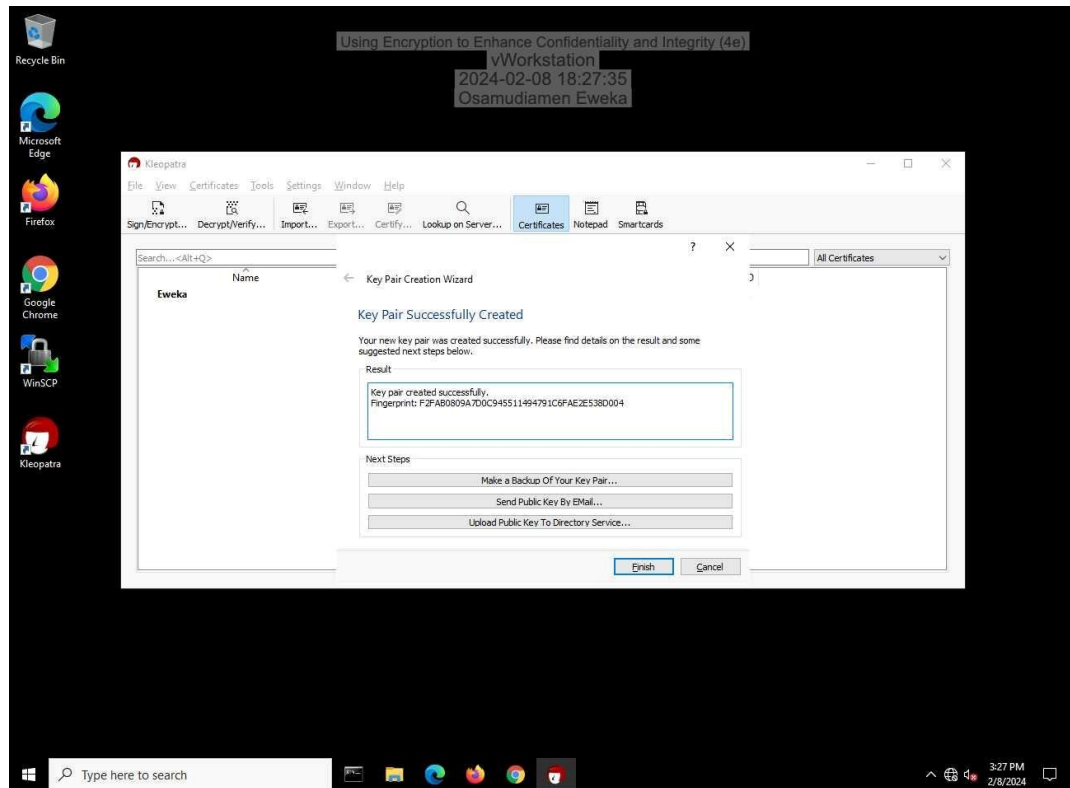
Create and Exchange Asymmetric Encryption Keys

In this segment of the lab, participants are tasked with creating asymmetric key pairs, emphasizing the practical application of encryption for secure communication. The process begins with the use of GPG4win, specifically its Kleopatra component, to generate these key pairs. This exercise underlines the benefits of asymmetric encryption for tasks where the emphasis is on security rather than speed, such as encrypting documents for safe transmission or storage. Moreover, it introduces the concept of digital signatures, which, although not encrypting data per se, play a crucial role in ensuring the integrity and authenticity of communications when used alongside encryption methods.

Participants follow a step-by-step procedure starting with logging into a virtual workstation and launching Kleopatra. They then navigate through the Key Pair Creation Wizard within Kleopatra to create a new personal OpenPGP key pair. This part of the lab serves to illustrate the practical steps involved in encryption key management, including the selection of key types, entering user details, and securing the key with a passphrase. The exercise further explores the importance of digital signatures in confirming the sender's identity and ensuring the message's integrity upon receipt. This hands-on approach not only aids in understanding the theoretical aspects of encryption and digital signatures but also provides insights into the challenges and solutions related to key management and trust establishment through Public Key Infrastructure (PKI). Figure 1 (Eweka, 2024) shows the implementation of this album steps, creating a fingerprint for a key pair.

Figure 1

Make a screen capture showing the fingerprint for your key pair.



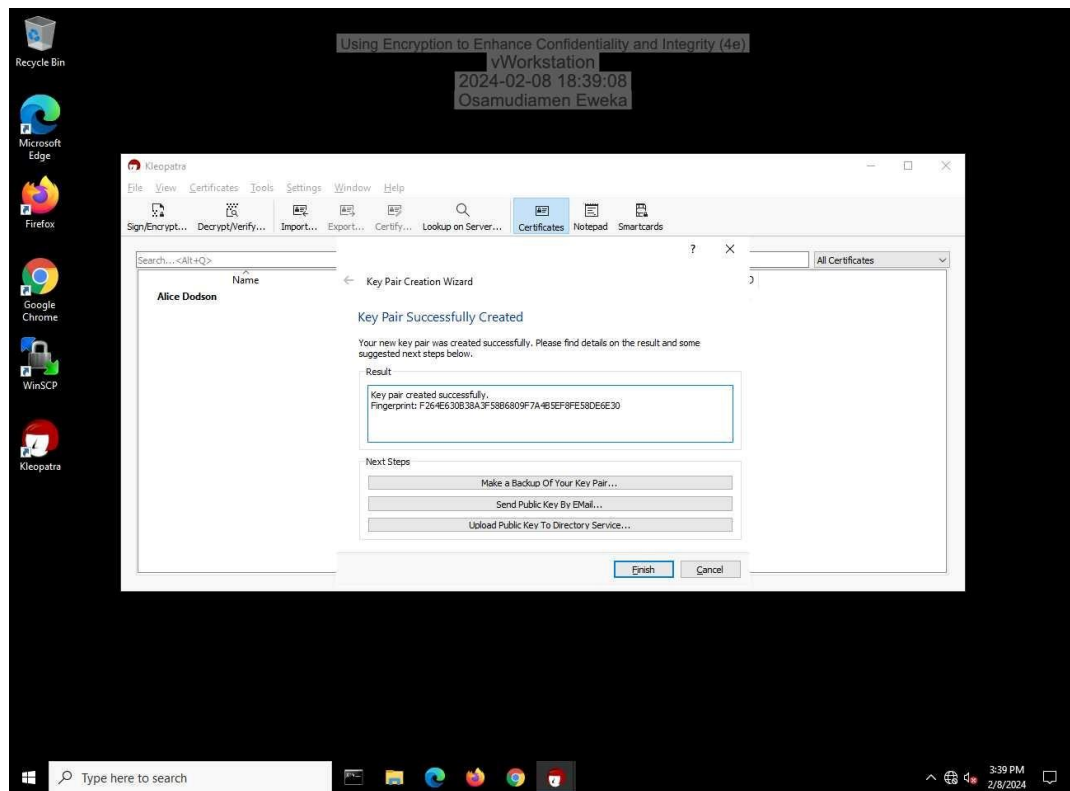
Upon completing the key pair creation with Kleopatra, the next steps involve exporting the newly generated public key to facilitate secure communication with another user, Alice. To achieve this, from the Kleopatra's Certificates page, you select the certificate associated with your name and navigate through the menu to export the certificate. This action exports your public key into a shared, accessible directory, ensuring Alice can import it for encrypted communication.

Subsequently, the lab exercise transitions to assuming Alice's perspective. Logging out of the current user account and signing into Alice's account on the virtual workstation, you repeat the process of key pair creation. This involves launching Kleopatra, generating a new personal OpenPGP key pair for Alice, and ensuring the key is protected with a passphrase. The exercise meticulously

guides you through the steps of key management, from creation to exporting the public key, demonstrating the practicalities of secure digital communication through asymmetric encryption. This hands-on approach not only solidifies understanding of encryption technologies but also emphasizes the importance of key management practices in safeguarding digital interactions. Figure 2 shows the screen capture of this procedure (Eweka, 2024).

Figure 2

Screen capture showing the fingerprint for Alice's key pair.

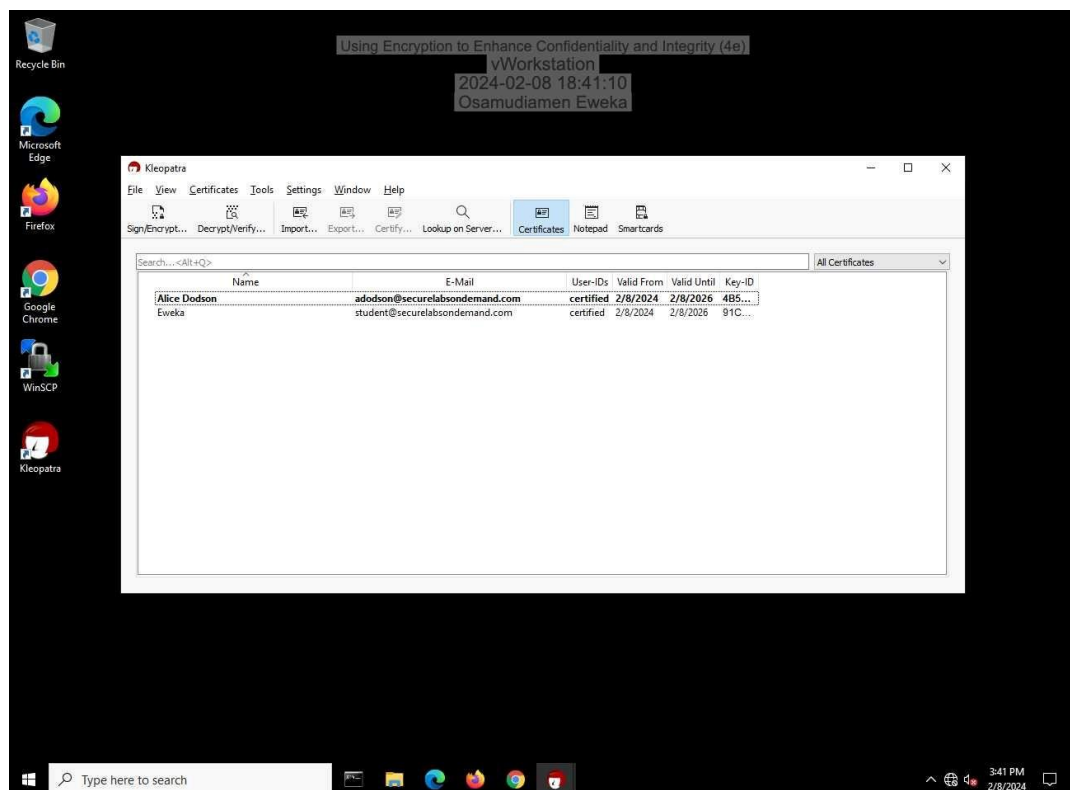


In this next phase of the lab, Alice imports and certifies the public key you've shared, integrating it into her certificate cache on Kleopatra. This process begins with selecting the "Import Certificates" option from Kleopatra's toolbar, navigating to the shared location where your public key file resides, and importing it into Alice's Kleopatra application. Upon importing, a certification

warning prompts action to confirm trust in the key's identity, leading to the certification of the public key. Certifying the key establishes it as trusted within Alice's web of trust, enabling her to use this certified public key for encrypting communications and verifying digital signatures from you. This step is crucial for ensuring the integrity and authenticity of the communication between Alice and you, laying a foundational trust for secure exchanges. Figure 3 shows the public key in Alice's certificate cache (Eweka, 2024).

Figure 3

showing your public key in Alice's certificate cache.

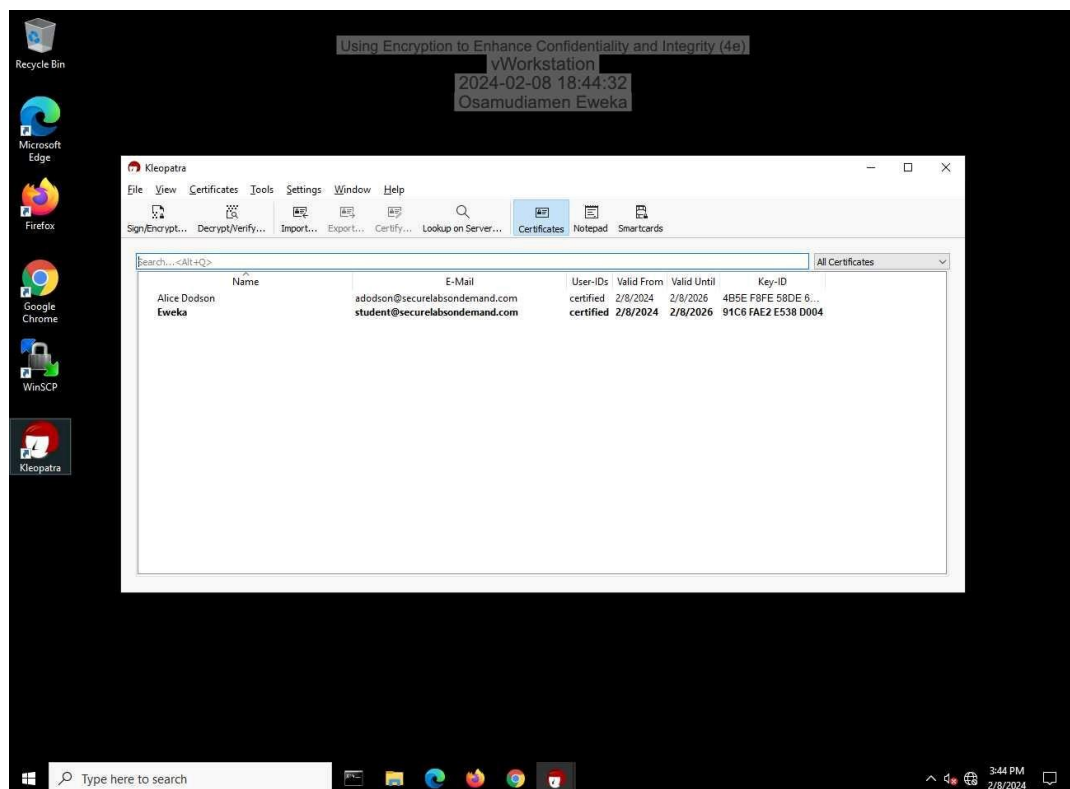


After logging back into the Student account, the next step involves repeating the process of importing and certifying a public key, this time focusing on Alice's key. By exporting Alice's public key to a shared folder and then importing it into the Student's certificate cache via Kleopatra, the

Student strengthens the web of trust between the two parties. This mutual exchange and certification of public keys ensure that both Alice and the Student can securely encrypt communications and verify each other's digital signatures, fostering a secure and trustworthy digital environment for their communications. This process underscores the importance of key management and the principles of trust in digital security, illustrating how encryption and certification play pivotal roles in maintaining the confidentiality and integrity of information exchanged between users. Figure 4 below shows the successful completion of this process.

Figure 4

Screen Capture showing Alice's public key in your certificate cache.



Section 1: Hands-On Demonstration

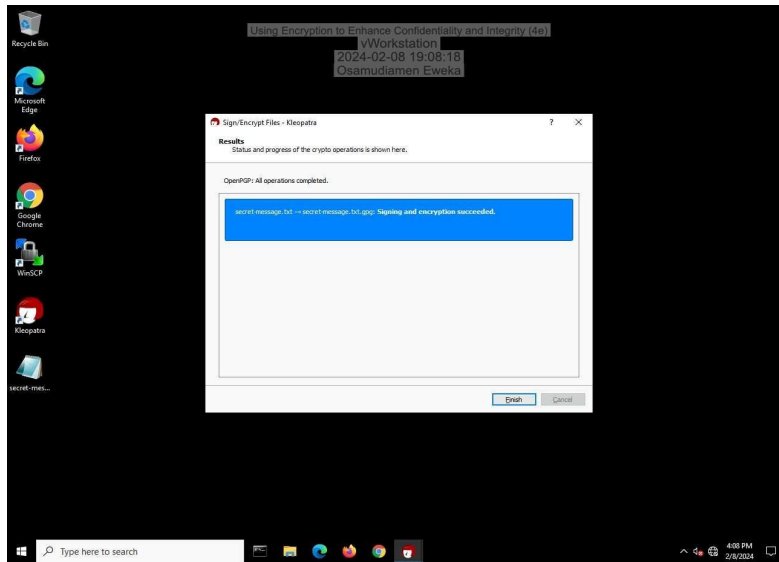
Part 2:

Encrypt a File Using Asymmetric Encryption

This part of the lab demonstrates the application of public-key (asymmetric) cryptography for secure communication. By creating a secret message and encrypting it with Alice's public key, only Alice, with her private key, can decrypt the message. This showcases the strength of asymmetric cryptography, note all cryptographic ciphers require keys. Some newer types of cryptographic algorithms derive their keys from other information, which is an approach that is similar to access controls that use inputs other than passwords. For example, identity-based encryption (IBE) uses the encryptor's identity to derive a key, and attribute-based encryption (ABE) uses descriptive attributes to encrypt and decrypt data (Kim, 2024), Allowing secure message transmission without the need to securely exchange a shared key as required in symmetric cryptography. The process involves digitally signing the message with the sender's private key for identity verification and encrypting it with the recipient's public key for confidentiality as illustrated in Figure 5 below (Eweka, 2024). This method ensures both the integrity of the message through digital signature and its confidentiality through encryption, highlighting the practical use of cryptographic principles for secure communication.

Figure 5

Screen Capture successful signing and encryption message.



Upon encrypting the secret message with Alice's public key, the resulting file, `secret-message.txt.gpg`, becomes a ciphertext as shown in Figure 6 (Eweka, 2024). This ciphertext is not readily interpretable as it consists of binary data that may display as odd or "unprintable" characters when viewed in a text editor. These characters are the visual representation of the encrypted information, illustrating the encryption process's effectiveness in transforming understandable text into a secure format that can only be decrypted by Alice's private key. This step underscores the essential principle of encryption: converting sensitive information into a secure format to protect its confidentiality during transmission.

Section 1: Hands-On Demonstration

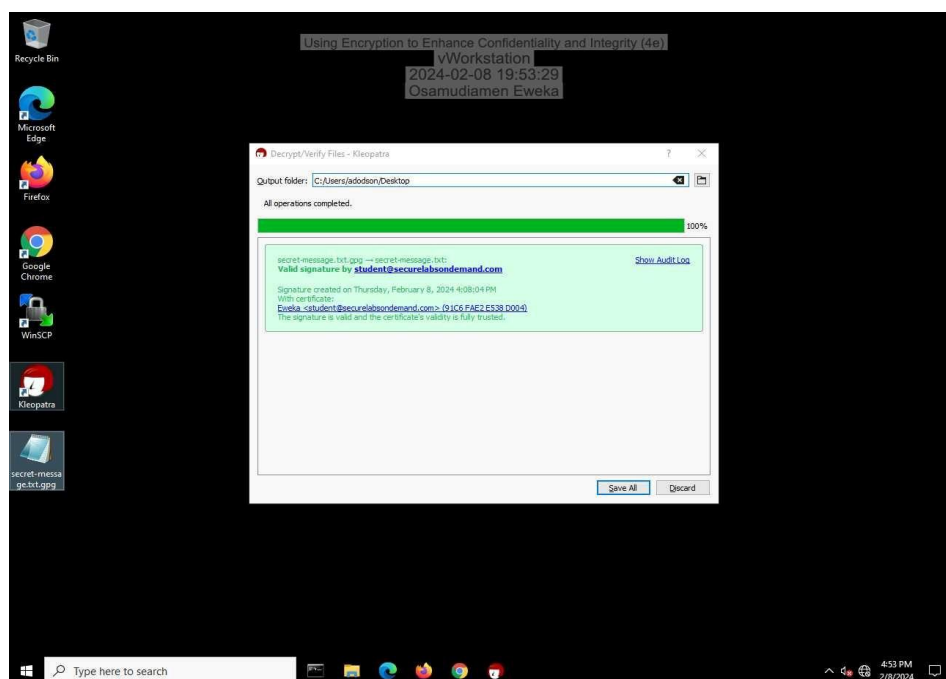
Part 3:

Decrypt a File Using Asymmetric Encryption

In this part of the lab, we proceed to decrypt the encrypted message, demonstrating the practical application of asymmetric cryptography. After transferring the `secret-message.txt.gpg` file to a shared folder accessible by both the Student and Alice accounts, Alice logs into her account, retrieves the file, and initiates the decryption process using her private key. This step highlights the decryption and verification process's dual nature, where the message is decrypted, and the sender's identity is authenticated through digital signature validation as illustrated in Figure 7 below (Eweka, 2024). This exemplifies how public-key cryptography ensures confidentiality and integrity, enabling secure and authenticated communication between parties.

Figure 7

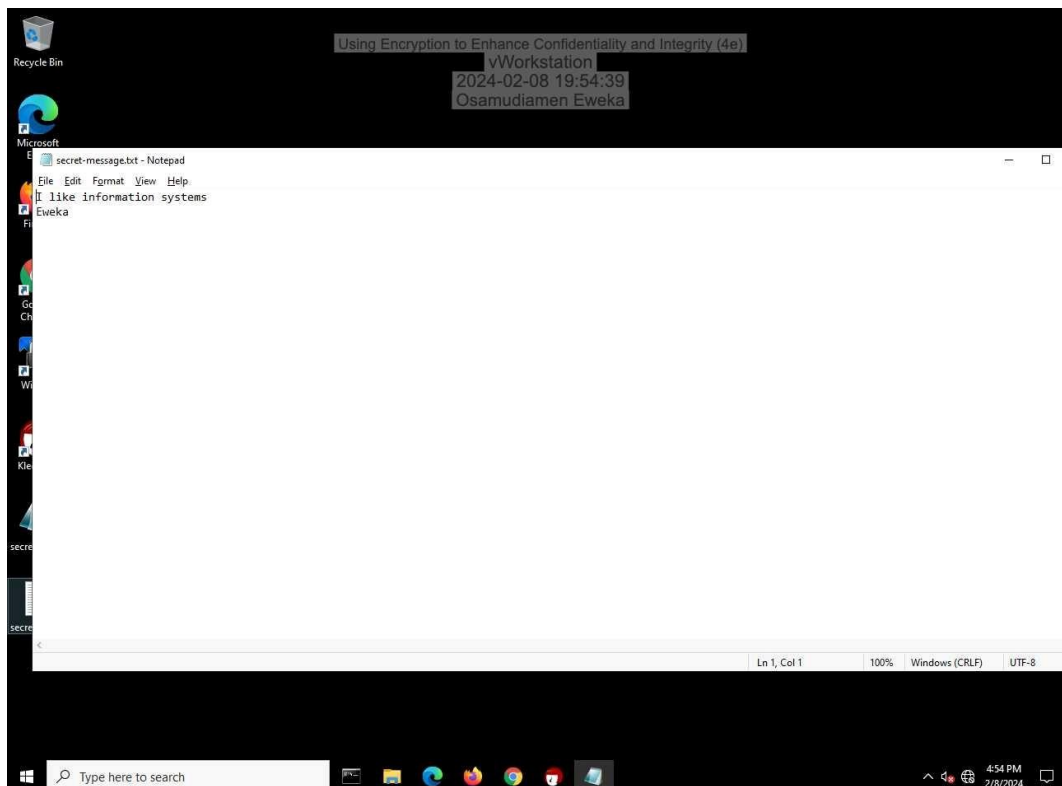
Make a screen capture showing the Decrypt/Verify Files window.



After the decryption process, clicking the "Save All" button in the Decrypt/Verify Files window saves the decrypted version of the `secret-message.txt` file to Alice's desktop. Opening this file with Notepad reveals the original message in a readable format illustrated in Figure 8 (Eweka, 2024), demonstrating the successful decryption of the message that was encrypted with Alice's public key. This step exemplifies the decryption process's ability to convert ciphertext back into plaintext, allowing the intended recipient to access the message's content securely. This process effectively showcases the practical application of asymmetric encryption and decryption for secure communication.

Figure 8

Make a screen capture showing the decrypted secret-message.txt file in Notepad.



Section 2: Applied Learning

Part 1:

Create an Asymmetric Key Pair

The lab exercise begins with participants logging into the TargetLinux01 system as instructors to utilize OpenSSL for establishing secure communications. Initially, a terminal window is opened, and an 8192-bit RSA private key is generated using AES-256 encryption, securing it with a passphrase. Following this, the corresponding public key is created from the private key, necessitating the passphrase for verification shown in figure 9 (Eweka, 2024). This process underlines the foundational role of asymmetric cryptography in securing internet communications and demonstrates the practical application of OpenSSL in generating cryptographic keys, essential for encrypted message exchanges. Through these steps, the lab emphasizes the critical aspects of cybersecurity, from key generation to the principles underlying secure communications.

Figure 10

Make a screen capture showing the instructor's key pair files.

```

Applications  Terminal - instructor@TargetLinux01  Fri 09 Feb, 07:26  instructor
Using Encryption to Enhance Confidentiality and Integrity (4e)
TargetLinux01
To run a command as administrator (user "root"), type "sudo 2024-02-09 10:26:35
See "man sudo_root" for details.
Osamudiamen Eweka
instructor@TargetLinux01:~$ openssl genrsa -aes256 -out instructor_private.key 8192
Generating RSA private key, 8192 bit long modulus (2 primes)
.....+++
.....+++
Enter pass phrase for instructor_private.key:
Verifying - Enter pass phrase for instructor_private.key:
instructor@TargetLinux01:~$ openssl rsa -in instructor_private.key -pubout -out instructor_public.key
writing RSA key
Enter pass phrase for instructor_private.key:
instructor@TargetLinux01:~$ ls *.key
instructor_private.key  instructor_public.key
instructor@TargetLinux01:~$
  
```

To finalize the setup for secure communication, the instructor's public key is copied to the `/tmp` directory using the command `cp instructor_public.key /tmp`. This action ensures the public key is accessible for subsequent secure communication processes. Then, to conclude the session, the command `exit` is used to close the terminal window, followed by logging out of the instructor account through the Applications menu. This sequence of actions effectively secures the instructor's participation in setting up the cryptographic environment for the lab exercise.`

Section 2: Applied Learning

Part 2:

Encrypt a File Using Symmetric Encryption

In this section of the lab, the student assumes the role of creating and encrypting a message for the instructor using symmetric encryption with OpenSSL. After logging into the TargetLinux01 machine and retrieving the instructor's public key, the student creates a plaintext file named `'secretmessage.txt'` and writes a message intended for the instructor. The student then uses OpenSSL to encrypt this message using 256-bit AES encryption, resulting in an encrypted file named `'secretmessage_ENCRYPTED.txt'`. During the encryption process, a unique password chosen by the student is used to generate a hash that serves as the encryption key, enhancing the security of the encrypted message. This step exemplifies the practical application of symmetric encryption and the importance of secure key management.

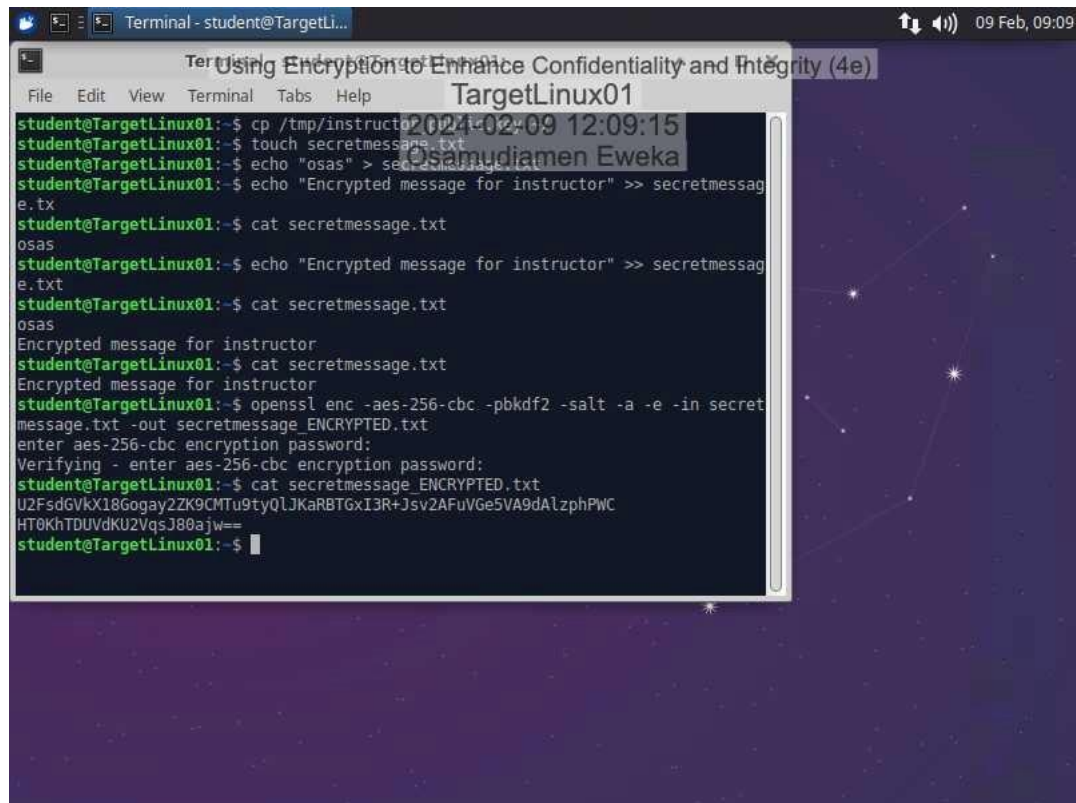
Question: Document the password you used to symmetrically encrypt the file:

Answer: Eweka

After encrypting the file using OpenSSL, executing the `'cat'` command to view its contents will display the ciphertext of `'secretmessage_ENCRYPTED.txt'`. This encrypted file will not resemble the original plaintext message; instead, it will show a series of seemingly random characters, demonstrating the effectiveness of the encryption as seen below in Figure 11 (Eweka, 2024). This ciphertext is the encrypted form of the original message, now secured with 256-bit AES encryption, and can only be decrypted with the correct password, illustrating the practical application of symmetric encryption for secure communication.

Figure 11

Make a screen capture showing the ciphertext in the secretmessage_ENCRYPTED.txt file.



```

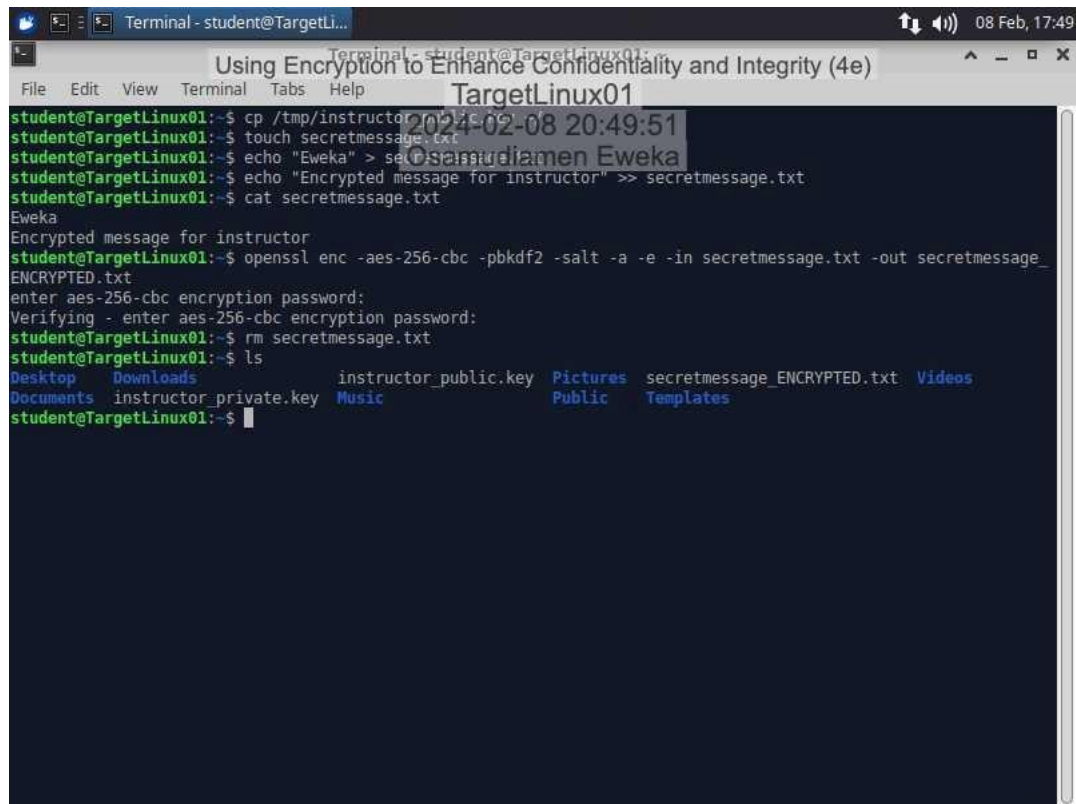
student@TargetLinux01:~$ cp /tmp/instructor/secretmessage.txt .
student@TargetLinux01:~$ touch secretmessage.txt
student@TargetLinux01:~$ echo "osas" > secretmessage.txt
student@TargetLinux01:~$ echo "Encrypted message for instructor" >> secretmessage.txt
student@TargetLinux01:~$ cat secretmessage.txt
osas
student@TargetLinux01:~$ echo "Encrypted message for instructor" >> secretmessage.txt
student@TargetLinux01:~$ cat secretmessage.txt
osas
Encrypted message for instructor
student@TargetLinux01:~$ cat secretmessage.txt
Encrypted message for instructor
student@TargetLinux01:~$ openssl enc -aes-256-cbc -pbkdf2 -salt -a -e -in secretmessage.txt -out secretmessage_ENCRYPTED.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
student@TargetLinux01:~$ cat secretmessage_ENCRYPTED.txt
U2FsdGVkX18Gogay2ZK9CMTu9ty0LJKaRBTGxI3R+Jsv2AFuVGe5VA9dAlzphPWC
HT0KhTDUVdK02VqsJ80ajw==
student@TargetLinux01:~$

```

After deleting the original plaintext file `secretmessage.txt` using the `rm` command, executing the `ls` command lists the contents of the current directory. This action verifies the deletion of the original file and displays the remaining files, including the encrypted file `secretmessage_ENCRYPTED.txt`, ensuring that sensitive information is secured and only the encrypted version is retained for transmission or storage as shown below in Figure 12 (Eweka, 2024). This step is crucial in maintaining the confidentiality of the information by removing unencrypted copies.

Figure 12

Make a screen capture showing the output of the ls command.



The image shows a terminal window titled "Terminal - student@TargetLinux01" with a menu bar (File, Edit, View, Terminal, Tabs, Help) and a title bar (Using Encryption to Enhance Confidentiality and Integrity (4e)). The terminal output shows the following commands and their results:

```
student@TargetLinux01:~$ cp /tmp/instructor_public.key .
student@TargetLinux01:~$ touch secretmessage.txt
student@TargetLinux01:~$ echo "Eweka" > secretmessage.txt
student@TargetLinux01:~$ echo "Encrypted message for instructor" >> secretmessage.txt
student@TargetLinux01:~$ cat secretmessage.txt
Eweka
Encrypted message for instructor
student@TargetLinux01:~$ openssl enc -aes-256-cbc -pbkdf2 -salt -a -e -in secretmessage.txt -out secretmessage_
ENCRYPTED.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
student@TargetLinux01:~$ rm secretmessage.txt
student@TargetLinux01:~$ ls
Desktop  Downloads  instructor_public.key  Pictures  secretmessage_ENCRYPTED.txt  Videos
Documents  instructor_private.key  Music      Public    Templates
```

Section 2: Applied Learning

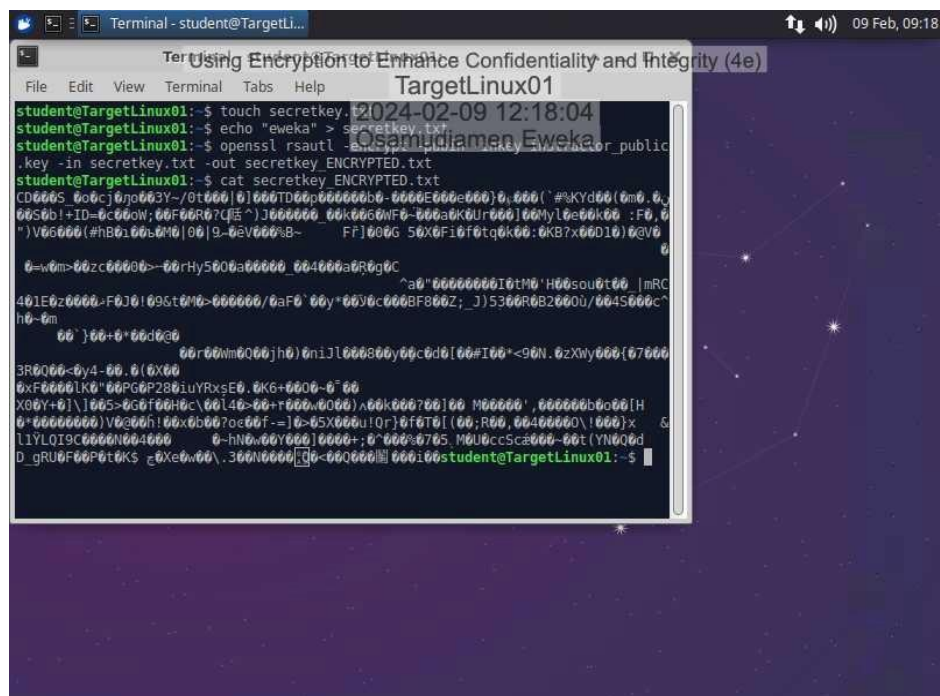
Part 3:

Transfer and Decrypt a File Using Hybrid Cryptography

This step in the lab demonstrates the use of hybrid cryptography by combining the strengths of both asymmetric and symmetric encryption methods. Initially, a file ('secretkey.txt') is created to store the symmetric key used for encrypting a message. This key is then encrypted using the instructor's public key through OpenSSL's RSA utility, producing an encrypted file ('secretkey_ENCRYPTED.txt'). Viewing the encrypted file's contents with the 'cat' command reveals ciphertext shown in Figure 13 (Eweka, 2024), showcasing the encryption process's effectiveness in securing the symmetric key. This approach underscores hybrid cryptography's utility in securely exchanging keys for encrypted communication.

Figure 13

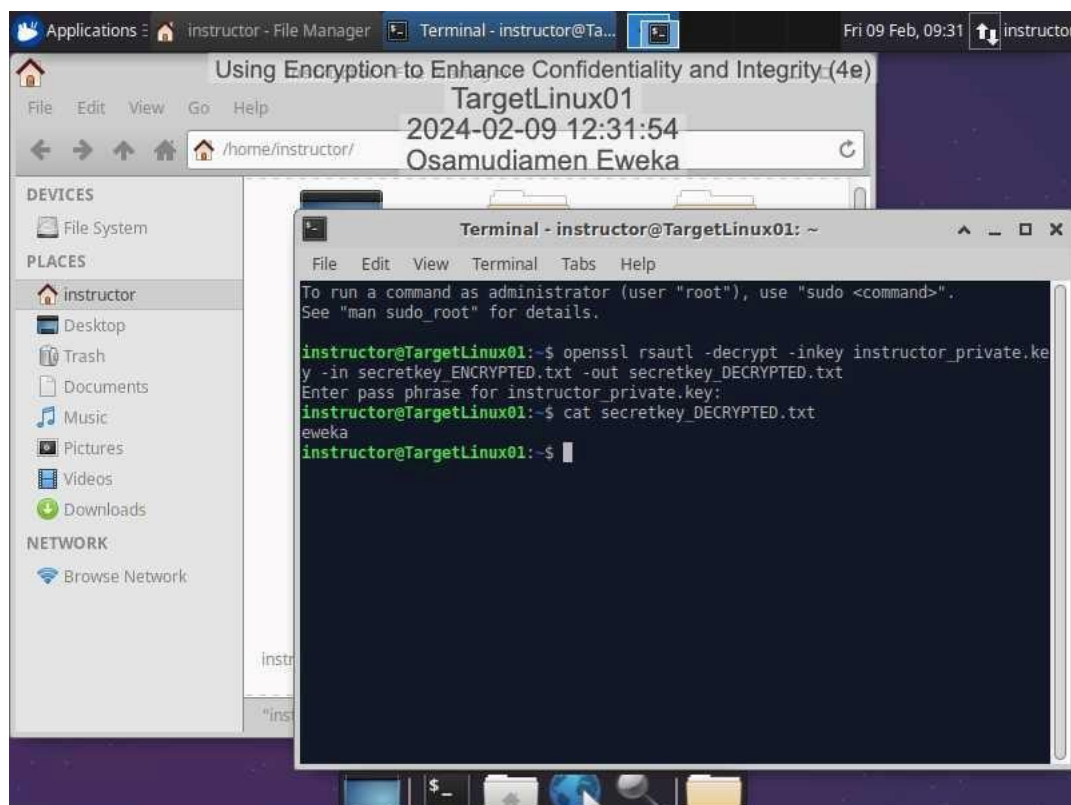
Make a screen capture showing the encrypted contents of the secretkey_ENCRYPTED.txt file.



Next, the instructor successfully decrypts the `secretkey_ENCRYPTED.txt` file using OpenSSL's RSA utility, accessing the symmetric key needed to decrypt the student's message. By applying the passphrase to the instructor's private key, the encrypted secret key file is converted back into plaintext, as shown in Figure 14 (Eweka, 2024), revealing the symmetric encryption key. This process highlights the decryption capability of asymmetric encryption, allowing the instructor to securely retrieve the symmetric key for further decryption of the student's message, exemplifying the practical application and security benefits of hybrid cryptography.

Figure 14

Make a screen capture showing the decrypted contents of the secretkey_DECRYPTED.txt file.



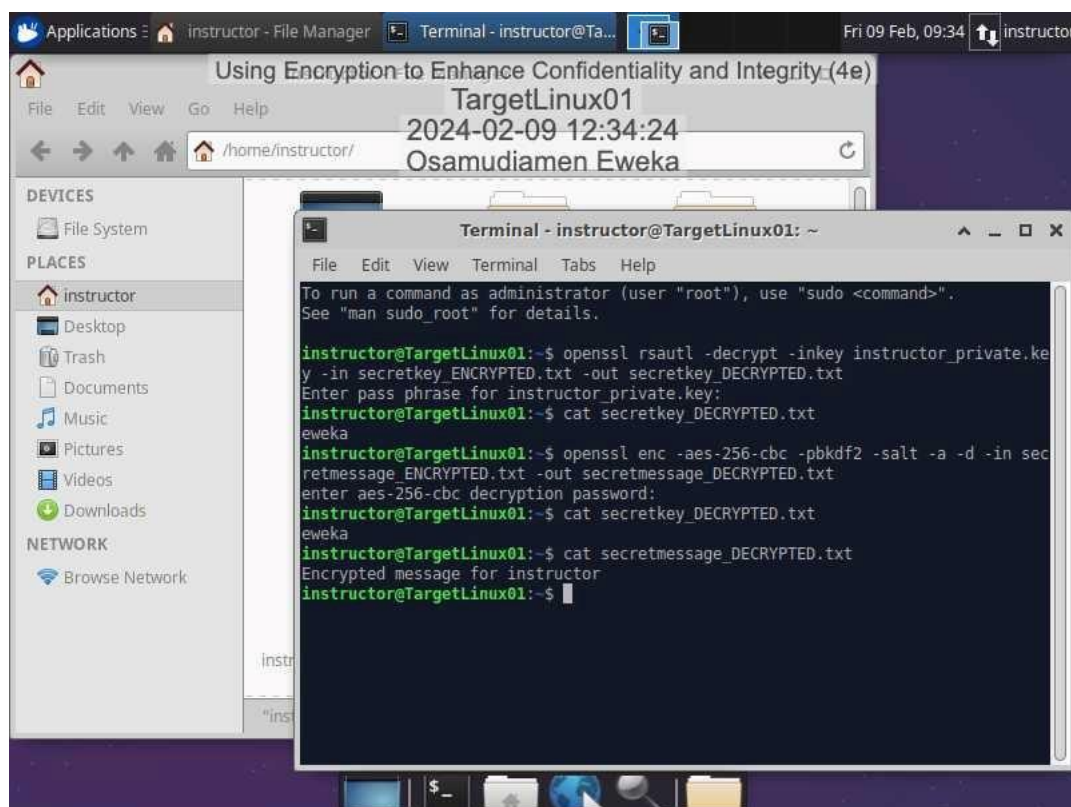
After decrypting the secret message using OpenSSL with the command:

(`openssl enc -aes-256-cbc -pbkdf2 -salt -a -d -in secretmessage_ENCRYPTED.txt -out secretmessage_DECRYPTED.txt`).

The instructor uses the decrypted symmetric key from `secretkey_DECRYPTED.txt` as the password. This decryption process transforms the ciphertext back into plaintext, making the original message readable again. Viewing the decrypted file's contents reveals the intended message as shown in Figure 14 (Eweka, 2024), demonstrating the practical application of hybrid cryptography in securely transmitting and accessing encrypted information.

Figure 15

Make a screen capture showing the contents of the `secretmessage_DECRYPTED` file.



Section 3: Challenge and Analysis

Part 1:

Digitally Sign a Document Using GPG

In this part after generating a new GPG key pair on the TargetLinux01 machine using the `gpg --gen-key` command with Quentin Compson's details, you then export the public key to a file using `gpg --armor --export qcompson@securelabsondemand.com > ~/keys/PublicKey.txt`. This process securely generates a cryptographic key pair, consisting of a private key for decryption and a public key for encryption. Verifying the key installation with `gpg --list-keys` confirms its addition to the public keyring, showcasing GPG's compatibility and functionality across different operating systems for secure communication and data encryption. This demonstrates GPG's utility in a cross-platform environment, emphasizing its role in enhancing cybersecurity measures.

Figure 16

Make a screen capture showing the key fingerprint for the key pair you generated in this part of the lab.

```

Applications  [x] Terminal - instructor@TargetLinux01  Sat 10 Feb, 11:19  instructor
Using Encryption to Enhance Confidentiality and Integrity (4e)
TargetLinux01
2024-02-10 14:19:46
Osamudiamen Eweka
public and secret key created and signed.
pub  rsa3072 2024-02-10 [SC] [expires: 2026-02-09]
    608DC0F410218B877F4975406C0368A7BB8A359F
uid   Quentin Compson <qcompson@securelabsondemand.com>
sub   rsa3072 2024-02-10 [E] [expires: 2026-02-09]

instructor@TargetLinux01:~$ gpg --armor --export qcompson@securelabsondemand.com > ~/keys/PublicKey.txt
instructor@TargetLinux01:~$ gpg --list-keys.
gpg: invalid option "--list-keys."
instructor@TargetLinux01:~$ gpg --list-keys
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: checking the trustdb
gpg: next trustdb check due at 2026-02-09
/home/instructor/.gnupg/pubring.kbx
-----
pub  rsa3072 2024-02-10 [SC] [expires: 2026-02-09]
    608DC0F410218B877F4975406C0368A7BB8A359F
uid   [ultimate] Quentin Compson <qcompson@securelabsondemand.com>
sub   rsa3072 2024-02-10 [E] [expires: 2026-02-09]

instructor@TargetLinux01:~$

```


Next creating and signing a message with GPG involves several steps. First, a new file `unsignedmessage.txt` is created and populated with a specific message using the `touch` and `echo` commands. Displaying the contents with `cat` confirms the message's presence as shown in Figure 17 (Eweka, 2024). Next, the message is digitally signed using `gpg --armor --sign`, generating a `signedmessage.txt` file, which is then securely stored in a designated directory. This process exemplifies the use of GPG for ensuring message authenticity and integrity through digital signatures.

Figure 17

Make a screen capture showing the contents of the unsignedmessage.txt file.

```

Applications ▢ Terminal - instructor@TargetLinux01 Sat 10 Feb, 11:22 instructor
Using Encryption to Enhance Confidentiality and Integrity (4e)
TargetLinux01
2024-02-10 14:22:45
Osamudiamen Eweka
sub rsa3072 2024-02-10 [E] [expires: 2026-02-09]

instructor@TargetLinux01:~$ gpg --armor --export qcompson@securelabsondemand.com
> ~/keys/PublicKey.txt
instructor@TargetLinux01:~$ gpg --list-keys.
gpg: invalid option "--list-keys."
instructor@TargetLinux01:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2026-02-09
/home/instructor/.gnupg/pubring.kbx
-----
pub rsa3072 2024-02-10 [SC] [expires: 2026-02-09]
608DC0F410218B877F4975406C0368A7BB8A359F
uid [ultimate] Quentin Compson <qcompson@securelabsondemand.com>
sub rsa3072 2024-02-10 [E] [expires: 2026-02-09]

instructor@TargetLinux01:~$ touch unsignedmessage.txt
instructor@TargetLinux01:~$ echo "This is a test message that will be digitally
signed by my own user account" > unsignedmessage.txt
instructor@TargetLinux01:~$ cat unsignedmessage.txt
This is a test message that will be digitally signed by my own user account
instructor@TargetLinux01:~$

```


Section 3: Challenge and Analysis

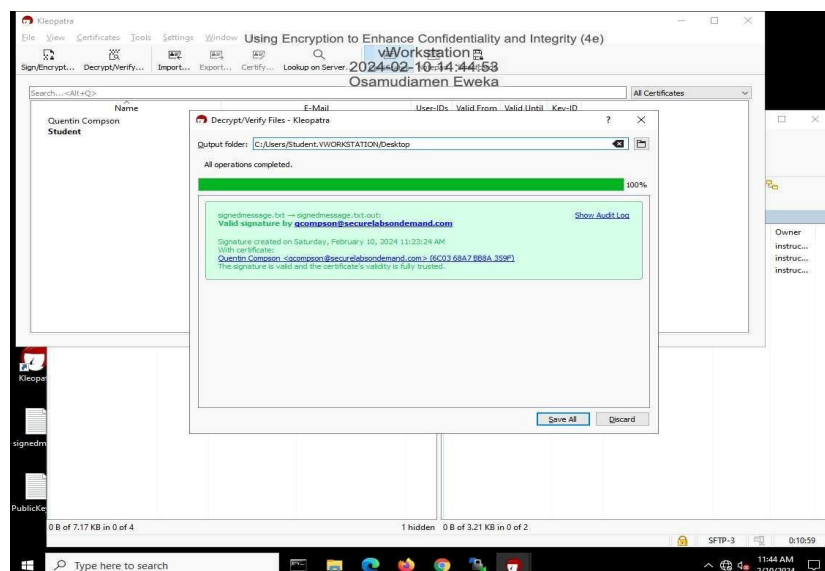
Part 2:

Verify the Digital Signature Using Kleopatra

This step involves transferring the `signedmessage.txt` file and the public key from TargetLinux01 to the vWorkstation using WinSCP, a secure file transfer application. After transferring the files, you'll launch Kleopatra on the vWorkstation to import the public key. This step is critical for establishing trust between the sender and receiver in a secure communication channel. The imported public key is then used to verify the digital signature of the `signedmessage.txt` file, ensuring the message's integrity and authenticity. This process exemplifies a practical application of hybrid cryptography, combining the strengths of asymmetric and symmetric encryption to secure communications across different platforms this process is illustrated below Figure 18 (Eweka, 2024).

Figure 18

Make a screen capture showing the successful signature verification on the signed message file.



Conclusion

This lab work provided a comprehensive exploration of secure communication techniques through the application of encryption and digital signatures across various platforms. Participants gained hands-on experience with GPG and OpenSSL, understanding the principles of asymmetric and symmetric encryption, and the practical implementation of hybrid cryptography. The lab illustrated the process of key generation, encryption, decryption, and secure file transfer, emphasizing the importance of cryptographic security in modern digital communications. This experience underscores the critical role of encryption in safeguarding information integrity and confidentiality in an increasingly interconnected world.

References

- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 1). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 2). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 3). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 4). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 5). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 6). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 7). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 8). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 9). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 10). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>
- Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 11). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 12). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 13). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 14). *Jones and Bartlett Learning Virtual Lab*. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 15). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 16). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 17). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Eweka O. (2024). Using Encryption to Enhance Confidentiality and Integrity (Figure 18). Jones and Bartlett Learning Virtual Lab. URL: <https://jbl-lti.hatsize.com/startlab>

Kim, D. (2021). *Fundamentals Of Information Systems Security + Cloud Labs*. Jones & Bartlett Cryptography <https://jbl-lti.hatsize.com/startlab>

Kim, D. (2021). *Fundamentals Of Information Systems Security + Cloud Labs*. Jones & Bartlett What Is Cryptography? <https://jbl-lti.hatsize.com/startlab>