Lab - Use Classic and Modern Encryption Algorithms

Objectives

Part 1: Use a Classic Encryption Algorithm

Part 2: Use a Modern Symmetrical Encryption Algorithm

Part 3: Use a Modern Asymmetrical Encryption Algorithm

Background / Scenario

Modern cryptography is primarily based on mathematical theory and computer science practice. Cryptographic algorithms are designed around computational complexity assumptions, making them difficult, if not impossible, for a threat actor to break. JCrypTool is a platform-independent, open-source software tool, and is part of the open-source project CrypTool. JCrypTool is an extendable e-learning platform presenting cryptography, cryptanalysis, and IT security in a modern and easy-to-use way. This lab will use JCrypTool to introduce classical, modern, symmetrical, and asymmetrical cryptographic algorithms.

Required Resources

PC with the **CSE-LABVM** installed in VirtualBox

Instructions

Part 1: Use a Classic Encryption Algorithm

In cryptography, a cipher is an algorithm for performing encryption or decryption. A cipher is a set of steps (an algorithm) for performing both an encryption, and the corresponding decryption. Early ciphers in cryptography were designed to allow encryption and decryption to take place by hand, while those which are developed and used today are only made possible by using computers. Classic algorithms are those invented up until around the 1950s.

A Caesar cipher, also known as the shift cipher, is one of the simplest and most widely known encryption techniques. The method is named after Julius Caesar, who used it in his private correspondence. Caesar is a type of substitution cipher in which each letter of plaintext is replaced by a letter some fixed number of positions away in the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.

Step 1: Launch the CSE-LABVM.

Step 2: Open and explore the JCrypTool.

a. Double-click the **jcryptool** icon on the desktop. The **jcryptool** directory opens.

b. Double-click the **JCrypTool** icon.

The tool has four windows:

- **File Explorer** - This is used to locate, open, and save files.
- **Help** - This is used to locate help files and tutorials.
- **Currently Open File** - This contains files to be operated on with cryptographic tools. The **unsaved001.txt** file should be open.
- **Crypto Explorer** - This provides access to cryptographic tools. By default, the Crypto Explorer is not displayed. To open it, click **Window** > **Show View** > **Crypto Explorer**.

Step 3: Use the Caesar algorithm to encrypt a text message.

a. To start, you will need to populate the currently open file with the message you wish to encrypt. Highlight all the text in the ***unsaved001.txt** file and replace it with the following message:

CRYPTOGRAPHY IS FUN. CAN YOU READ THIS SECRET MESSAGE?

b. In the **Crypto Explorer,** click **Classic** if it is not expanded and double-click **Caesar**.

c. In the **Operation** section, select **Encrypt** if it is not already selected.

d. In the **Alphabet** section, verify that the options **Select alphabet** and **Upper Latin (A-Z)** are selected. If not, select them now.

e. In the **Key** section, change **Enter key using a character** to **K**. Leave all the other options at the default.

f. Click **Finish** to save the options and encrypt the data.

g. A new file named ***out001.txt** opens with the encrypted message.

Step 4: Decrypt the encrypted ciphertext with the Caesar algorithm.

a. Move the ***out001.txt** file to the File Explorer window, if necessary. This ensures that the **Crypto Explorer** will use this file as the active file. You can also close the ***unsaved001.txt** file.

b. In the **Crypto Explorer** tab, double-click the **Caesar** algorithm again.

c. In the **Operation** section, select **Decrypt**.

d. Select the same settings to decrypt the current ciphertext in the output file ***out001.txt**.

e. Click **Finish** to save the options and decrypt the data.

f. Close all the files in the **File Explorer**. There is no need to save them.

Step 5: Change the Caesar algorithm settings.

a. Create a new input text file by selecting **File** > **New** > **Empty File in Texteditor**.

b. Type the following message: **Cryptography is fun. Can you read this secret message?**

c. In the **Crypto Explorer** tab, double click the **Caesar** algorithm again.

d. **Encrypt** should already be selected. For **Select alphabet**, set the value to **Upper and lower Latin (A-Z, a-z)**. For the amount of shift along the alphabet, set the value to **13**.

e. Click **Finish** to save the options and encrypt the data.

f. Close the file in the **File Explorer**. There is no need to save it.

Further Exploration

Experiment on your own with Caesar and other classic cryptography algorithms to see how they work.

Part 2: Use a Modern Symmetrical Encryption Algorithm

In this part, you will use a modern symmetrical encryption algorithm. One of the most popular versions of a modern cryptographic algorithm is Advanced Encryption Standard (AES). AES is a symmetric cryptographic cipher in software and hardware that is used throughout the world to encrypt sensitive data. The AES cipher requires an encryption key to control the encryption and decryption process. This algorithm is considered a strong cryptographic protocol based on its complexity and key length of128 bits.

Step 1: Use AES encryption to encrypt a text message.

a. Create a new input text file by selecting **File** > **New** > **Empty File in Texteditor**.

b. Type the following message: **Cryptography is fun. Can you read this secret message?**

c. In the **Crypto Explorer** tab, click **Symmetric** to expand it, if necessary, and then double-click **AES**.

d. Use the following settings.

- Operation: **Encrypt**
- Key source: **Custom key**
- Key length: **128**
- Key (hex): **AA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF**
- Mode: **(ECB) Electronic Codebook**
- Padding: **PKCS#5 Padding**

e. Click **Finish**. An output file with a .bin extension opens. You will see four rows with 16 hexadecimal values in each row. The ciphertext is shown to the right for each row.

Step 2: Use AES to decrypt a text message.

a. Double-click **AES** again.

b. Change the **Operation** to **Decrypt**, and then use the settings from Step 1 to decrypt the ciphertext.

c. Click **Finish** to save the options and decrypt the data. An output file with a .bin extension opens with the decrypted text.

d. Close all files.

Part 3: Use a Modern Asymmetrical Encryption Algorithm

In this part, you will use a modern asymmetrical encryption algorithm. Unlike symmetric encryption, asymmetric encryption encrypts and decrypts the data using two separate yet mathematically connected cryptographic keys. These keys are known as a 'Public Key' and a 'Private Key'. For a person to send an encrypted message to another person using asymmetric encryption, they request a public key from them, then use it to encrypt a message with an agreed upon algorithm. The other person decrypts the message using their private key. The message cannot be decrypted using the public key.

Step 1: Use RSA asymmetrical encryption to encrypt a text file.

a. Create a new input text file by selecting **File** > **New** > **Empty File in Texteditor**.

b. Type the following message: **Cryptography is fun. Can you read this secret message?**

c. In the Crypto Explorer, click **Asymmetrical** to expand it, and the double-click **RSA** to open the algorithm settings.

d. Use the following settings:

- Operation: **Encrypt**
- Keystore: Click **Create a new pair in the keystore**.
- In the **New key pair** dialog box, enter the following:
  - Contact name: **John Smith**
  - Password: **Secret**
  - Leave all other entries as the default.

e. Click **Finish**.

f. Click **Finish** in the **RSA - encryption** dialog box to encrypt the data. An output file with a .bin extension opens with the encrypted text.

Step 2: Use RSA asymmetrical encryption to decrypt a text file.

a. Double click **RSA**.

b. Select **Decrypt** for the operation.

c. Select Key = **"John Smith" – public key – 1024**

d. Click **Finish** to decrypt the ciphertext.

e. Enter the password **Secret**. Click **OK**.