TUNIS BUSINESS SCHOOL

IT360 PROJECT TOOLS AND DEVELOPMENT PHASES

KENZA BACHA

ASMA BOUBAKER

2023/2024

This deliverable provides an overview of the project's tools and technologies used in the development process, emphasizing the importance of security throughout the development lifecycle of the image encryption algorithm.

I. Introduction:

The algorithm designed is a user-friendly command-line interface (CLI) that allows users to encrypt, decrypt images, and generate a shared secret key conveniently from the Linux terminal, as well as WINDOWS cmd.

II. Tools:

- List of tools to be used throughout the development process.
  - **Programming Languages:** All of the project's code is developed using Python, i.e. the encryption and decryption process, as well as the Diffie Hellman key exchange algorithm are all developed in one single python code file. The program will consist of a class named 'Image encryption' mainly containing the objects 'encrypt', 'decrypt', and 'key generation'.

    Libraries needed:
    - Pillow: It provides extensive support for opening, manipulating, and saving many different image file formats.
    - Random: it is used to generate random keys.
    - PyCryptodome: comprehensive library that provides a wide range of cryptographic algorithms and protocols, including symmetric and asymmetric encryption, hashing, digital signatures, key derivation, and more.
    - Sys: the definition and manipulation of command-line arguments within a Python program.
    - Argparse: allows you to easily define and handle command-line arguments and options, making your scripts more user-friendly and robust.
    - Os: module in Python provides a way to interact with the operating system, allowing you to perform tasks such as file operations, process management, and environment variable manipulation.
  - **Development Environment:** Visual Studio Code.
  - **Version Control:** GitHub
  - **User interface:** LINUX Command Line (CLI).

III. Development Phases:

- **Step 1: Setup and Imports**

Importing the necessary libraries for handling file operations, cryptography, and command-line argument parsing.

- **Step 2: Argument Parsing**

Using modules to parse command-line arguments for input and output image paths, as well as for specifying the mode of operation (encryption, decryption, or key generation).

- **Step 3: Diffie-Hellman Key Exchange**

Implement the Diffie-Hellman key exchange algorithm to generate a shared secret key between two parties. Each party generates its private key and corresponding public key. Then they exchange public keys and derive the shared secret key. All coded in python using the specific libraries.

- **Step 4: Encryption process**

If the program is in encryption mode, read the input image file, apply padding to match the block size of the encryption algorithm (AES), encrypt the padded data using the derived shared secret key, and save the encrypted image along with encrypted transposition key to the specified output file.

- **Step 6: Decryption process**

If the program is in decryption mode, read the encrypted image file as well as the encrypted transposition key specified in the metadata of the image, decrypt the data using the derived shared secret key, remove any padding, and save the decrypted data to the specified output file.

- **Step 7: Running the Command**

Run the algorithm from the command line, specifying the input image file paths, as well as the mode of operation (encryption, decryption, or key generation) with the predefined arguments.