# Image Encryption/Decryption Project Report

This report explains the implementation and usage of the Image Encryption/Decryption Tool. The tool provides a simple interface for encrypting and decrypting images using the Advanced Encryption Standard (AES) algorithm in Cipher Block Chaining (CBC) mode.

## Overview

The Image Encryption/Decryption Tool is implemented in Python 3 and uses the following libraries:

* os: For file I/O and random number generation.
* Cryptodome.Cipher: For AES encryption and decryption.
* Cryptodome.Util.Padding: For padding and unpadding data to a multiple of the block size.
* termcolor: For colored output to the console.
* tqdm: For progress bars during encryption and decryption.

The tool provides a simple command-line interface with the following options:

1. Encrypt an image
2. Decrypt an image
3. Exit

The user is prompted to enter their choice and follow the instructions for each option.

## Encryption

When the user chooses to encrypt an image, they are prompted to enter the encryption key. The key is used to derive a 32-byte AES key using UTF-8 encoding and zero-padding if necessary.

The tool then generates a 16-byte initialization vector (IV) using os.urandom() and creates an AES cipher in CBC mode with the derived key and IV. The image file is read into memory and padded to a multiple of the block size using PKCS#7 padding from the Cryptodome.Util.Padding library.

The IV and encrypted image data are concatenated and written to a text file with the same name as the original image file, but with a .txt extension. The tool displays a progress bar during the encryption process.

## Decryption

When the user chooses to decrypt an image, they are prompted to enter the key used for encryption. If the key is incorrect, the tool displays an error message and prompts the user to enter the key again.

If the key is correct, the tool reads the encrypted data from the text file and extracts the IV and encrypted image data. The key is derived from the user input and used to create an AES cipher in CBC mode with the extracted IV.

The encrypted image data is decrypted and unpadded using the Cryptodome.Util.Padding library. The decrypted image data is written to a JPEG file with the same name as the original image file, but with a .jpg extension. The tool displays a progress bar during the decryption process.

## Conclusion

The Image Encryption/Decryption Tool provides a simple and easy-to-use interface for encrypting and decrypting images using the AES algorithm in CBC mode. The tool is implemented in Python 3 and uses popular cryptographic libraries such as Cryptodome and tqdm.

The source code for the tool is available and can be modified and extended for other use cases. However, it is important to note that cryptography is a complex field and any modifications or extensions to the code should be thoroughly reviewed and tested for security vulnerabilities.

By

* P Bhargav Sai
* K Lakshmi Kanth
* U Mukesh
* S Mohammed Khadeer