Title: "Image Encryption and Decryption using Triple DES in Python"  
  
Introduction:

The purpose of this project is to implement image encryption and decryption using the Triple Data Encryption Standard (DES) algorithm in Python. The code provided utilizes the tkinter library for the graphical user interface (GUI) and the Crypto.Cipher module for encryption and decryption operations. The project allows users to select an image file (in formats such as jpg, png, jpeg, or bitmap) and perform encryption or decryption based on their choice.  
  
Code Explanation:

The code begins by importing the necessary libraries and modules: tkinter, filedialog, Crypto.Cipher, and hashlib.md5. These libraries enable GUI functionality, file selection, and cryptographic operations.  
  
The "encrypt\_or\_decrypt" function is defined, which is triggered when the "Encrypt/Decrypt" button is clicked. Within this function, the user is prompted to select an image file using the filedialog.askopenfile method. The selected file's path is stored in the "file\_path" variable.  
  
The user is also required to enter a key for encryption or decryption. The key is obtained from the entry1 Text widget and stored in the "key" variable. The key is then processed using the md5 hashing algorithm and encoded as ASCII before being passed to the DES3.adjust\_key\_parity function. This function adjusts the key's parity and returns a 16-byte key, which is stored in the "tdes\_key" variable.  
  
A DES3 cipher object is created using the obtained key, the DES3.MODE\_EAX mode, and a nonce value of 'e'. The EAX mode is used for encryption or decryption.  
  
The selected operation (encryption or decryption) is determined based on the value of the "selected\_operation" variable. If encryption is chosen, the cipher.encrypt method is used to encrypt the contents of the selected image file. Otherwise, the cipher.decrypt method is used for decryption. The resulting encrypted or decrypted bytes are stored in the "new\_file\_bytes" variable.  
  
Finally, the modified image bytes are written back to the original file using the open(file\_path, 'wb') context manager.  
  
The GUI:

The graphical user interface (GUI) is created using the tkinter library. The root window is initialized with a size of 800x600 pixels.  
  
A label is added to prompt the user to select the desired operation (encryption or decryption). Radio buttons are provided for selecting the operation, with "Encrypt" set as the default option.  
  
The "Encrypt/Decrypt" button triggers the "encrypt\_or\_decrypt" function when clicked.  
  
A Text widget is provided for the user to enter the encryption or decryption key. A label is placed next to it to indicate the purpose of the input.

Conclusion:

This project demonstrates the implementation of image encryption and decryption using the Triple DES algorithm in Python. The provided code utilizes the tkinter library for GUI functionality and the Crypto.Cipher module for cryptographic operations. Users can select an image file, choose between encryption and decryption, and provide a key for the operation. The modified image file is then saved with the changes applied.