Automated Batch Encryption of Python Files and Management Tools

Introduction

This project provides an automated system for encrypting Python scripts using AES encryption. It includes tools to batch encrypt `.py` files, generate secure keys for each file, and optionally remove the original scripts post-encryption. The aim is to offer a secure and efficient method for protecting Python source code.

Project Components

The project consists of three primary scripts:

- `encrypt_folder.py`: This script automates the encryption of all Python files in a selected folder and its subdirectories.
- `delete_py_folder.py`: This optional script allows for the safe deletion of original Python files after encryption.
- `decrypt_file.py`: For accessing the encrypted content, this script provides a mechanism to decrypt individual files as needed.

Features and Functionality

1. **Batch Encryption** (`encrypt_folder.py`)
   - **Tkinter File Dialog**: A user-friendly graphical interface to select the folder containing Python files.
   - **AES Encryption**: Each file is encrypted using the robust AES (Advanced Encryption Standard) algorithm.
   - **Key Management**: Unique encryption keys are generated for each file, ensuring individual file security.
   - **Encryption Keys Log**: A `encryption_keys.txt` file is created, listing each encrypted file along with its corresponding base64-encoded key.

2. **Secure Deletion of Original Files** (`delete_py_folder.py`)
   - **Optional File Deletion**: Post-encryption, users have the option to permanently delete the original Python files.
   - **Directory Selection via GUI**: Users can select the folder for file deletion through a graphical interface.
   - **Safety and Confirmation**: The script confirms the deletion, ensuring that files are not deleted accidentally.

3. **Decryption of Encrypted Files** (`decrypt_file.py`)
   - **Selective File Decryption**: Users can decrypt any encrypted file as needed.
   - **Key Entry**: The decryption process requires the corresponding file's key, ensuring that only authorized users can access the content.
   - **Restoration to Original Format**: Decrypted files are restored to their original `.py` format.

Getting Started

- **Environment Setup**: Run these scripts in a Python environment with the necessary dependencies installed (`pycryptodome` and `tkinter`).
- **Running the Scripts**:
  - Launch `encrypt_folder.py` to encrypt files and generate the keys log.
  - If desired, use `delete_py_folder.py` to remove original files securely.
  - Access encrypted content as needed using `decrypt_file.py`.