Project Report: Secure File Sharing Tool

1. **Introduction:**
   The **Secure File Sharing Tool** is a Python-based command-line application designed to encrypt and decrypt files using symmetric key encryption. The primary goal of this project is to ensure the confidentiality and security of files when shared or stored. This report details the project's objectives, design, implementation, and potential future enhancements.
2. **Objectives:**
   - To provide a simple and user-friendly interface for file encryption and decryption.
   - To implement secure symmetric encryption using the AES (Advanced Encryption Standard) algorithm.
   - To ensure the integrity of the encrypted files.
   - To educate users on the importance of data security through hands-on experience with cryptographic techniques.
3. **Tools and Technologies:**
   - **Programming Language:** Python
   - **Cryptographic Library:** PyCryptodome (for implementing AES encryption)
   - **Development Environment:** Any Python-compatible IDE (e.g. Visual Studio Code)
4. **Design:**
   The tool consists of two main functions:
   - **Encryption:** Converts plaintext files into ciphertext using a symmetric key.
   - **Decryption:** Converts ciphertext back into plaintext using the same symmetric key.
5. **Implementation Details:**
   Imports:
   Encryption Function:
   - Uses AES in CBC (Cipher Block Chaining) mode.
   - Generates a random Initialization Vector (IV).
   - Pads the plaintext to ensure it fits into the block size.
   - Writes the IV followed by the ciphertext to an output file.

   Decryption Function:
   - Reads the IV from the input file.
   - Decrypts the ciphertext and unpads it to retrieve the original plaintext.
   - Writes the plaintext to the output file.

   User Interface:
   - A simple command-line interface that prompts the user for input and outputs messages based on user actions.
6. **Testing:**
   - **Encryption Test:** Tested with various file types (text, images, etc.) to ensure correct encryption and generation of ciphertext files.
   - **Decryption Test:** Verified that the decryption function successfully restored the original files from the encrypted versions.

- **Error Handling:** Implemented checks for file permissions and existence to handle potential errors gracefully.

7. **Challenges Faced:**
   - **Permission Issues:** Encountered permission errors when trying to access certain directories, which were resolved by ensuring the output paths were correct and accessible.
   - **Understanding Cryptography:** Gaining a deeper understanding of symmetric encryption and its implications was crucial for successful implementation.
8. **Future Enhancements:**
   - **GUI Implementation:** Develop a graphical user interface for improved user experience.
   - **Key Management:** Implement secure key storage and retrieval mechanisms.
   - **File Integrity Checks:** Introduce mechanisms (like checksums) to verify the integrity of files before and after encryption.
   - **File Compression:** Add functionality to compress files before encryption for efficiency.
9. **Conclusion:**
   The **Secure File Sharing Tool** demonstrates the practical application of symmetric encryption techniques. It effectively encrypts and decrypts files, ensuring that sensitive data remains secure during storage and transmission. This project serves as a foundational step towards understanding and implementing cryptographic methods in real-world applications.
10. **References:**
PyCryptodome documentation: https://www.pycryptodome.org/
Cryptography tutorials and best practices:
https://www.geeksforgeeks.org/cryptography-tutorial/