Challenge Title: Secure File Encryption Tool

Description:

Develop a command-line file encryption tool that allows users to encrypt files securely and decrypt them using industry-standard encryption algorithms. The tool should provide the following functionalities:

1. Encryption: Implement a function to encrypt a given file using a strong encryption algorithm (e.g., AES) with a user-provided passphrase.
2. Decryption: Implement a function to decrypt an encrypted file using the same passphrase used for encryption.
3. Key Derivation: Utilize a key derivation function (KDF) to securely derive encryption keys from user-provided passphrases, enhancing the security of the encryption process.
4. Error Handling: Implement robust error handling mechanisms to gracefully handle errors such as invalid input, file access issues, or cryptographic failures.
5. Secure Coding: Follow secure coding practices to mitigate common vulnerabilities such as buffer overflows, input validation issues, and cryptographic weaknesses.
6. Documentation: Provide clear and concise documentation explaining how to use the tool, including command-line arguments, usage examples, and security considerations.

Evaluation Criteria:

1. Security: Effectiveness of the encryption mechanism in protecting the confidentiality of the encrypted files.
2. Functionality: Completeness and correctness of the implemented encryption and decryption functionalities.
3. Usability: Ease of use and clarity of the command-line interface for interacting with the tool.
4. Secure Coding: Adherence to secure coding practices and avoidance of common security pitfalls.
5. Documentation: Clarity and completeness of the provided documentation, including usage instructions and security considerations.

Additional Information:

- Students are encouraged to use programming languages and libraries that support cryptographic operations (e.g., Python with PyCrypto and Java with Bouncy Castle).
- The tool should be designed to run on standard operating systems (e.g., Windows, Linux, macOS) and be easily deployable without additional dependencies.
- All submissions must include source code, documentation, and necessary testing and evaluation instructions.
- Students should prioritize security considerations throughout development, including securing sensitive data and cryptographic operations.
- A Key Derivation Function (KDF) is a cryptographic function that derives one or more secret keys from a single secret value or a set of secret values.