**5-2 Activity: Encryption Coding**

David J Allen

CS405 Secure Coding 23EW6

Southern New Hampshire University

Professor Mike Prasad

July 30, 2023

**5-2 Activity: Encryption Coding**

**Console Output Screenshot**

**A screenshot of a computer

Description automatically generated**

**Process Summary**

* In the encrypt\_decrypt function, the transformation of characters was missing, and the function was only copying the characters from the source string. The bug was fixed by implementing the XOR operation between the source string characters and the corresponding key characters.
* The read\_file function was returning a hardcoded string instead of reading the content from the specified file. The bug was fixed by using std::ifstream to open the file and reading its content line by line to construct the file\_text string.
* The save\_data\_file function was not implemented, and as a result, the data was not being saved to the file. The bug was fixed by opening a file with std::ofstream and writing the student name, timestamp, key, and data to the file in the specified format.
* The use of std::strftime with std::localtime raised a compiler warning about unsafe usage. The bug was resolved by replacing std::localtime with localtime\_s to generate the timestamp in a safer manner.

Thoroughly reviewing the code was essential to comprehend its intended functionality, the purpose of each function, and their interactions. After testing the code and comparing the output with expected behavior, discrepancies and incorrect results were identified. Additionally, compiler errors and warnings were examined to pinpoint specific issues, including the use of unsafe functions and missing implementations.

Once the bugs were identified, systematic fixes were applied to address the issues. The XOR encryption function was rectified to perform actual encryption, and the file reading function was implemented to extract content from the specified file. Furthermore, a file-saving function was created to store data in the desired format. Subsequently, the corrected code underwent testing with sample input files to ensure that encryption, decryption, file reading, and saving functionalities worked as intended.

Throughout the debugging process, adherence to industry-standard best practices was prioritized. This involved using meaningful function and variable names, incorporating in-line comments to elucidate code logic, and adhering to C++ conventions for function implementation. The overall debugging process involved meticulous code analysis, the identification of specific bug types, and the application of appropriate corrections to achieve the desired XOR encryption, file reading, and saving functionality. Consequently, the program is now capable of performing encryption and decryption operations on the specified data file format.