**Documentation: CryptoSoft**

**1. Summary**

CryptoSoft is a library and application for encrypting and decrypting files using an XOR algorithm. This documentation details how it works, how to integrate it as a DLL into a project, and how to use it as a stand-alone application.

Version 3.0 introduces **single-instance management,** preventing multiple instances of CryptoSoft from running simultaneously on the same machine.

**2. Introducing CryptoSoft**

CryptoSoft offers a simple, fast XOR encryption method, based on an encryption key passed as a parameter. Its implementation is minimalist, guaranteeing compatibility with projects requiring fast file encryption.

Key features:

- File encryption and decryption via XOR

- DLL-based for integration into existing projects

- Can be used as a stand-alone console application

- Simple operation with two parameters: file to be encrypted and key

- Mono-instance execution (new in version 3.0)

- Compatible with EasySave 3.0 for encryption of backed-up files

**3. Technical explanation**

CryptoSoft is based on the XOR algorithm, a simple, fast, yet reversible encryption using the same key. The program supports:

**a) Verification of file existence**

Before performing encryption, the application checks whether the file exists, otherwise an error message is displayed.

**b) Transformation files**

- The file is read in bytes.

- It is then processed with the XOR algorithm, applying the key supplied.

- Once transformed, the file is overwritten with the new encrypted version.

- Write encrypted/decrypted file

- Execution time is measured with a stopwatch.

**c) Code structure**

The code is based on a main FileManager class, which provides the TransformFile() method for encryption.

Une image contenant texte, capture d’écran, document

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Explanation of class:**

- Checks if files exist

- Loads file contents into memory

- Applies XOR encryption with the supplied key

- Writes the encrypted or decrypted file to disk

- Returns encryption execution time

**Main program Program.cs (executable mode)**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Program explanation:**

- Retrieves the file path and encryption key passed as arguments.

- Creates an instance of FileManager.

- Executes file encryption/decryption.

- Returns execution time or error code if a problem occurs.

**4. Integration as DLL**

**a) DLL creation**

CryptoSoft can be compiled into a library (DLL) for integration into a project. Here are the steps:

1. Create a “Class library (.NET 8)” project

2. Add the FileManager class to the FileManager.cs file

3. Compile the project and retrieve CryptoSoft.dll from bin/Debug/net8.0

**b) Using the DLL in a C# project**

Add the DLL as a reference in Visual Studio:

1. Right-click on the .sln project → Add reference

2. Select CryptoSoft.dll by browsing the location of this DLL from “browse”.

**Example of use in a C# project**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

**5. Use console/graphics applications from the create DLL**

CryptoSoft can also be run as an independent console application, by retrieving the executable from *bin/release/net8.0/win-x64/publish/CryptoSoft.exe*, which can then be shared as a zipped file for use by all users with a new name Directory.

**a) Execution command**

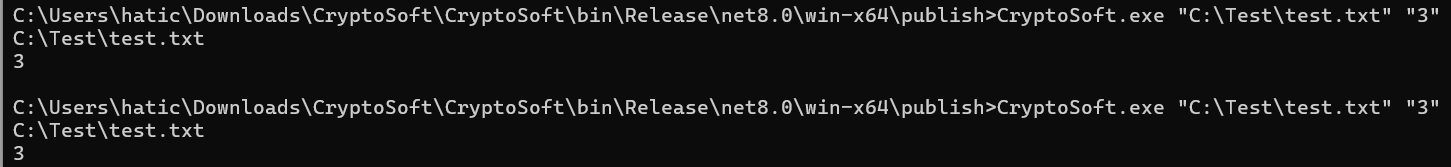
**To run CryptoSoft from a terminal:**

Example:

This is where the file is located, with the command “cd directory/from/file/exe”.



A 1st run for encryption with key “3” and a 2nd run for decryption.

****

**To run CryptoSoft from a console and graphical interface:**

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect. **Une image contenant texte, capture d’écran, affichage, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.**

****

Follow the instructions from the console using CryptoSoft.exe!

**To run CryptoSoft from a graphical interface:**

CryptoSoft-only can also be used via the graphical user interface just by executing the .exe in this zip file; or from the console interface from the documentation above:

**Une image contenant texte, capture d’écran, affichage, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.Une image contenant texte, capture d’écran, logiciel, affichage

Le contenu généré par l’IA peut être incorrect.**

****

Very easy to use, all marked “highly interactive”!

**6. Compiling a console or graphical application**

If you wish to create an executable:

1. Open the command prompt in the project folder

2. Execute the command:



The CryptoSoft.exe executable can be found in bin/Release/net8.0/win-x64/publish/

**See higher up in the zips!**

**7. Troubleshooting and common errors**

**a) Error "File not found"**

**Solution**: Check that the file passed as a parameter exists.

**b) Error "Index out of range"**

**Solution**: Be sure to pass two arguments (file + key).

**c) Unreadable file after transformation**

**Solution**: The file has been encrypted with XOR, so run CryptoSoft again with the same key to decrypt it.

**d) DLL does not exist**

**Solution**: Check that it is referenced in the project (bin/Debug/net8.0/).

**e) CryptoSoft is already running:**

**Solution:** Close the existing instance before opening a new one.

**8. Conclusion**

CryptoSoft is a simple, effective solution for encrypting and decrypting files using a minimalist XOR algorithm. Its DLL-based integration makes it easy to use within a C# project, while its console and graphical application modes allow it to be run independently in mono-instance.