

Martis Hastam

Introduction:

Martis Hastam translates to The Spear of Mars in Latin. The front line defense of any digital data is a secure encryption layer to shield it. Martis Hastam does exactly that. It uses an AES256-GCM encryption to encrypt documents including text files, pdf files, images, audio, and video files. Using a tried and tested NIST approved cipher is vital in protecting data in this ever-changing digital world.

Technical:

The software can be split into two sections: The User Interface, and the cryptography engine.

User Interface:

The UI is built entirely in java using the JavaFX platform, which uses FXML technology.

Cryptography Engine:

The cryptography engine on the other hand is built in C++ using the Crypto++ class library. C++ was chosen due to its speed and efficiency compared to other languages. Crypto++ was selected due to its reliability, efficiency, and the fact that its open source. Meaning the implemented algorithms have been extensively tried and tested. Rather than encrypting the entire document once and producing one key. I encrypt each line of the document using a different key. This adds an extra layer of protection.

Tutorial:

IMPORTANT

As of now only file paths without spaces work. Next iteration will fix this issue.

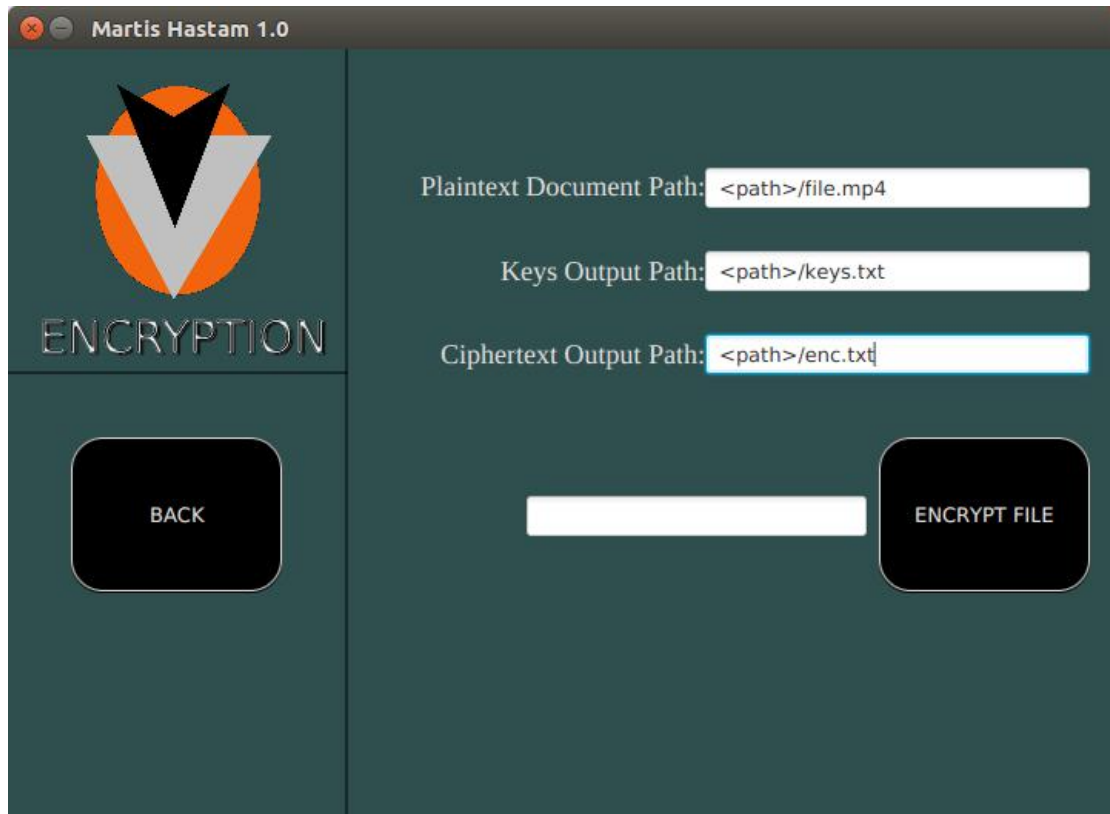
- I. Unzip Martis-Hastam anywhere where the path to the unzipped file does not have spaces.
- II. Double click on Martis Hastam.jar

Now you get this page:



III. Click ENCRYPTION.

Now you get this page:



The screenshot shows a software window titled "Martis Hastam 1.0". The interface is split into two main sections. The left section features a logo consisting of a stylized 'V' inside an orange circle, with the word "ENCRYPTION" written below it. At the bottom of this section is a black button labeled "BACK". The right section contains three input fields for file paths, each preceded by a label: "Plaintext Document Path:" with the value "<path>/file.mp4", "Keys Output Path:" with the value "<path>/keys.txt", and "Ciphertext Output Path:" with the value "<path>/enc.txt". Below these fields is a long, empty white rectangular box. To the right of this box is a black button labeled "ENCRYPT FILE".

IV. Enter the full path of the file you want to be encrypted with its extension.

V. Enter the full path of a text file with a name of your choice that will contain the keys to the encrypted document.

VI. Finally, enter the full path of a text file with a name of your choice that will contain the ciphertext. (FILE MUST HAVE A .TXT EXTENSION).

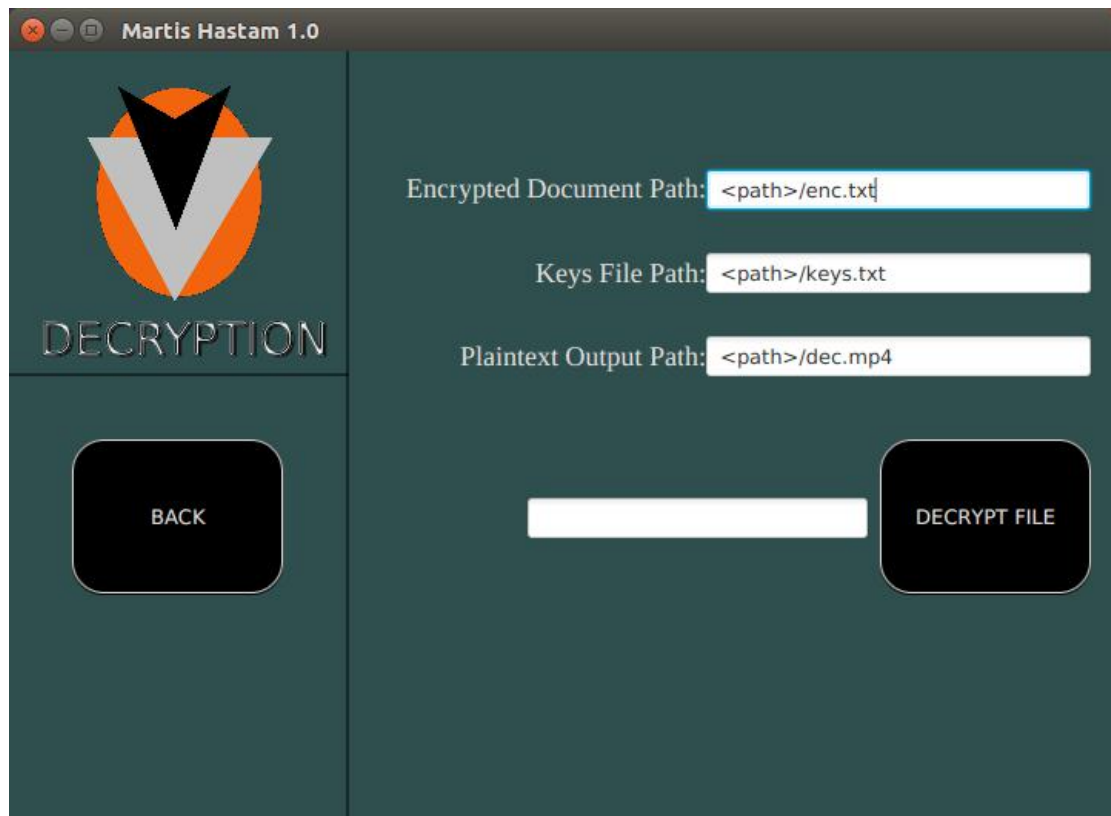
VII. Click on ENCRYPT FILE.

VIII. If all goes well, the bottom output field will output <Encryption Successful>

IX. Click BACK.

X. Once you are on the main page, click DECRYPTION.

Now you get this page:



XI. Enter the full path of the file you want to be decrypted with its extension.

XII. Enter the full path of the keys text file corresponding to the encrypted document.

XIII. Finally, enter the full path of a text file with a name of your choice that will contain the plaintext. (FILE MUST HAVE THE ORIGINAL FILE S EXTENSION).

XIV. Click on DECRYPT.

XV. If all goes well, the bottom output field will output <Decryption Successful>

@Author Tony Wagdi

