Encoding Secret Messages in Audio Steganography

Installation and Usage Guide

Anastasia Grosch

CIS 542: Digital Forensics

Table of Contents

1 System Requirements

The audio steganography program runs on MATLAB and requires the ”Symbolic Math Toolbox” add-on. The program is compatible on any device and operating system that supports MATLAB and the needed toolbox.

2 Software Installation Steps

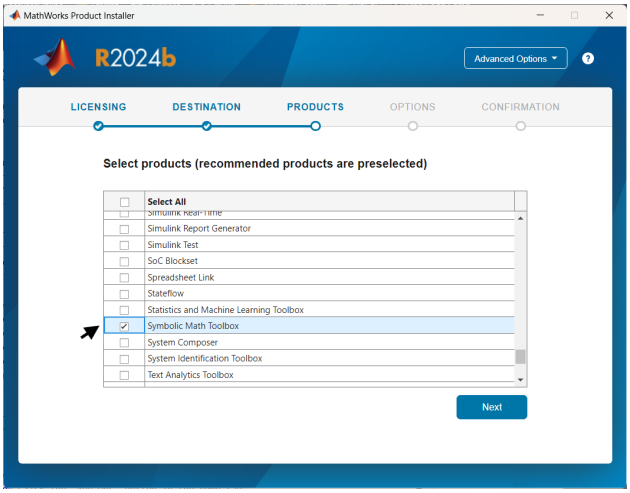
If you already have MATLAB installed, skip to the add-on instructions in Section 2.2.

2.1 MATLAB Installation

Follow the steps for downloading MATLAB’s installer for UMass Dartmouth members outlined here: `https://www.umassd.edu/cits/downloads/matlab/`.
Note: Only download the installer, do not follow UMassD’s installation steps.

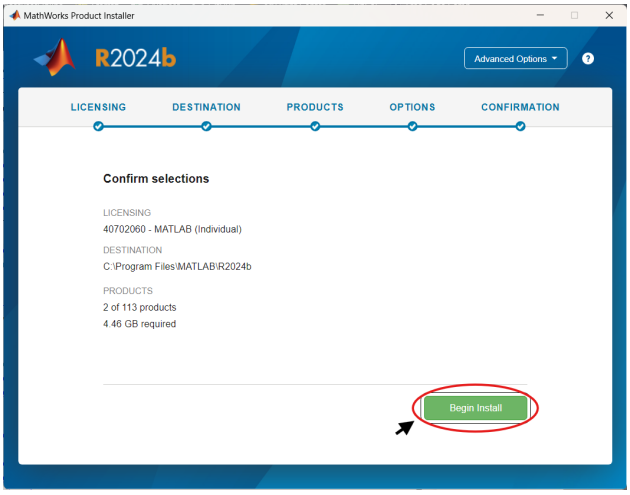
Once you download the MATLAB installer, follow the steps below.

1. Run the downloaded MATLAB installer executable file.
2. Accept the License Agreement.
3. Select desired (default) license and destination folder.
4. In Products, set the checkbox for the ”Symbolic Math Toolbox” and click Next.



5. Select desired options.

6. Select Begin Install.

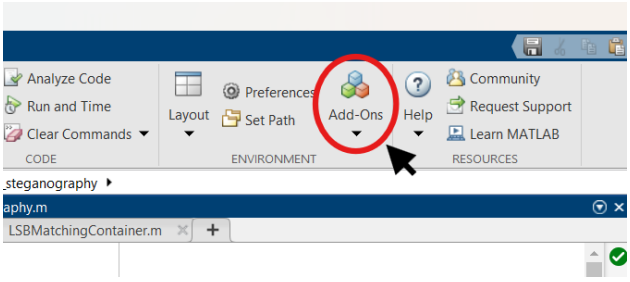


Once MATLAB completes its installation, you are ready to download/run the audio steganography program! Skip to Section 2.3.

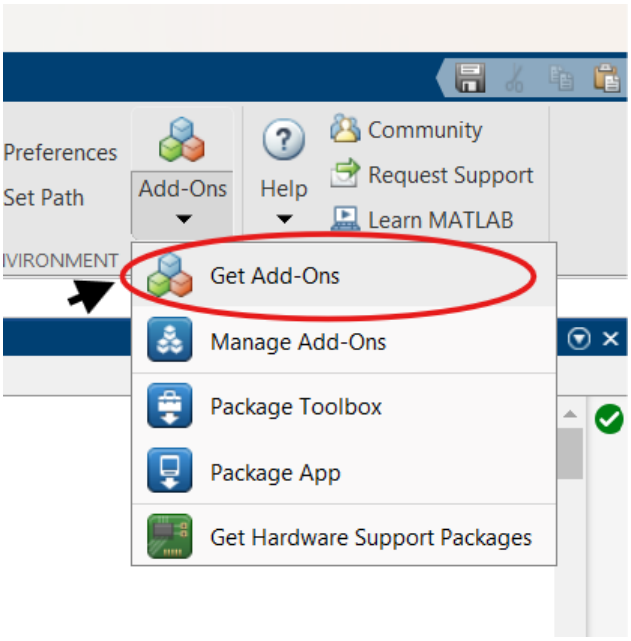
2.2 Symbolic Math Toolbox Installation

If the ”Symbolic Math Toolbox” is already installed on your system, skip to the program download instructions in Section 2.3.

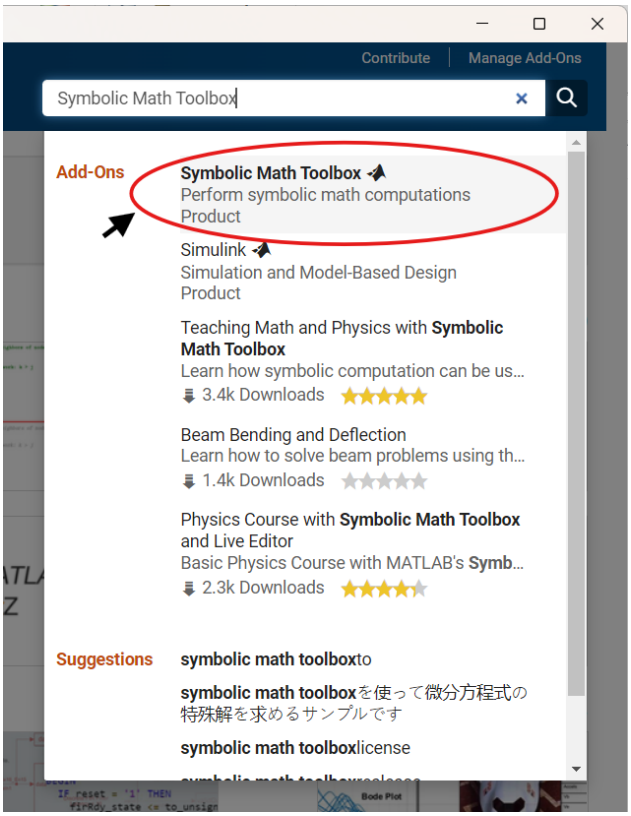
1. Click the ”Add-Ons” button in the Home tab.



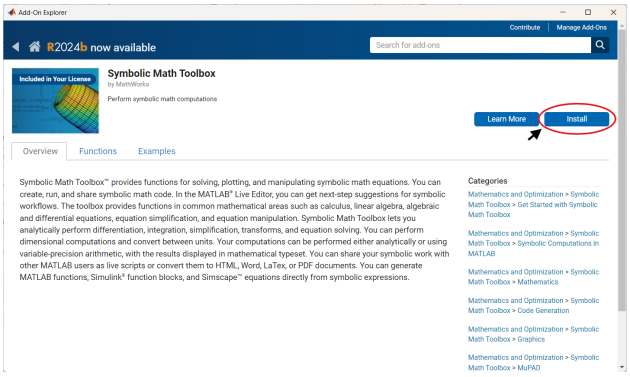
2. Select ”Get Add-Ons”.



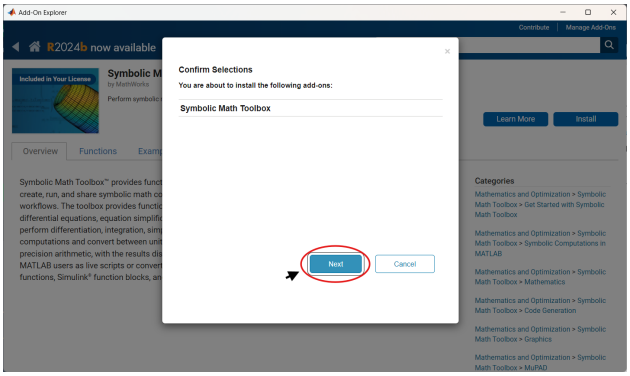
3. In the Search bar, type ”Symbolic Math Toolbox” and press enter.



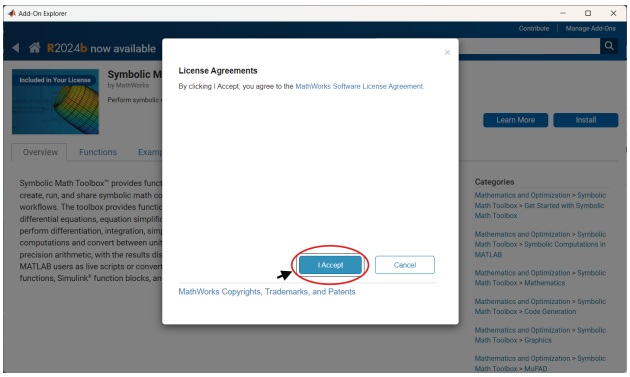
4. Select Install.



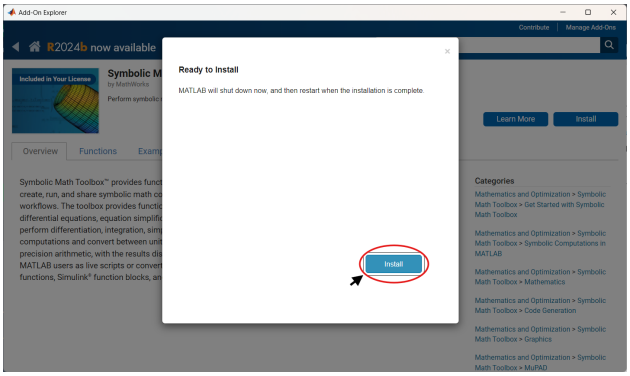
5. Confirm toolbox installation.



6. Agree to the license agreement.



7. Select Install.



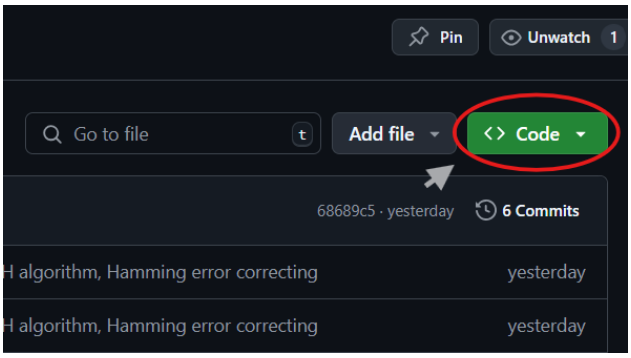
Once the toolbox completes its installation, you are ready to download/run the audio steganography program!

2.3 Download Program

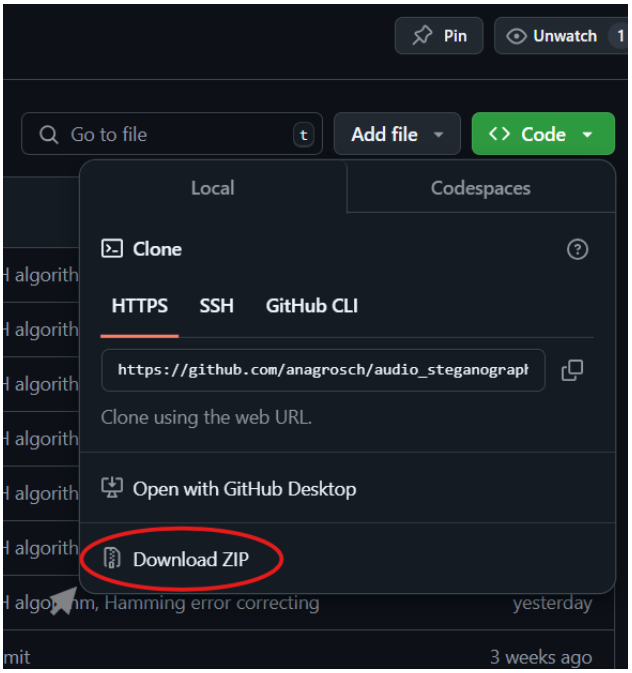
The audio steganography program is available on GitHub at `https://github.com/anagrosch/audio_steganography`. Download the program to run on your system by following the steps below.

Download repository as `.zip` file:

1. Go to the audio_steganography GitHub repository.
2. Click <> Code.



3. Select Download Zip.

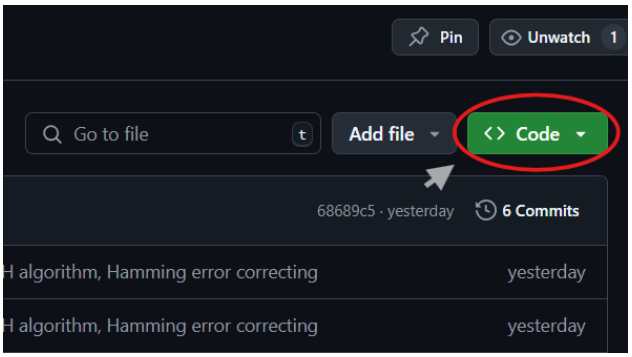


4. Extract files from downloaded `.zip` file.

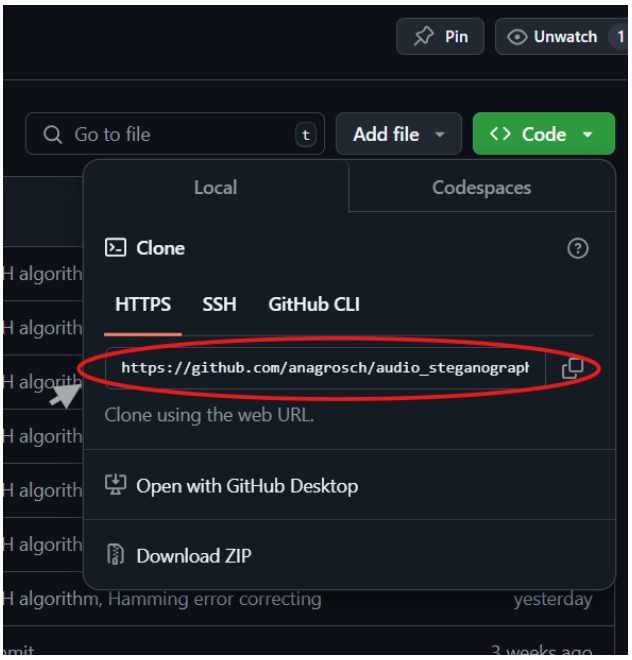
Clone repository:

1. Go to the audio_steganography GitHub repository.

2. Click <> Code.



3. Copy the HTTPS URL.

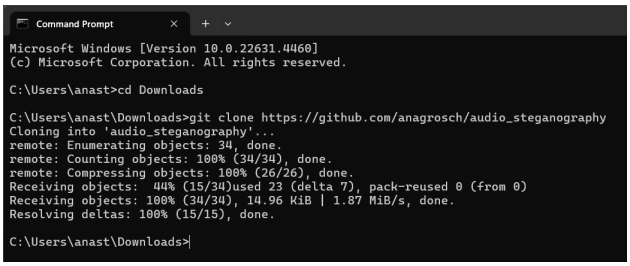


4. Open Command Prompt.

5. Change the current working directory to the directory you want to save the program in.

6. Type the following command with the copied URL and hit Enter:

   ”`git clone https://github.com/anagrosch/audio_steganography`”



3 Usage Instructions

The audio steganography program consists of the following files:

Main File	audio_steganography.m
Algorithm Files	LSBMatchingContainer.m PhaseCodingContainer.m BBFEchoHidingContainer.m
Supporting Files	HammingContainer.m readBinData.m bin2File.m
input_files	hidden_text.txt short_text.txt piano.wav piano.mp3

Table 1: List of Program Files

Quick tip: the program plays the input audio during both the encryption and decryption process. The encryption process also plays the output audio file. If listening to the same audio on repeat drives you insane, you can stop it by commenting out lines 57 and 89 with '%'.

```
54
55      % Play input audio
56      fprintf("Playing '%s'...", audioInput.filename);
57      %playClip(audioInput.fullfile);
58      fprintf("Done\n\n");
59
```

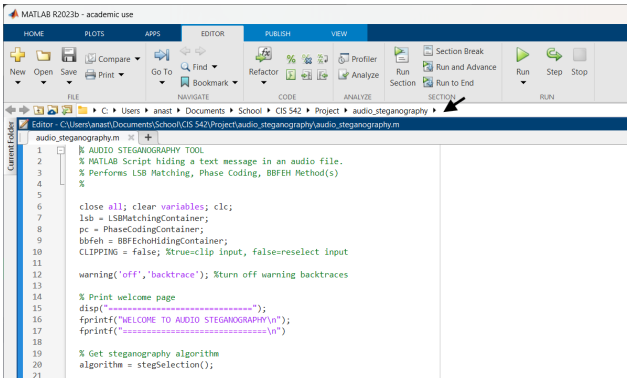
```
86
87      % Play output audio
88      fprintf("Playing output '%s'...", output.filename);
89      %playClip(output.fullfile);
90      fprintf("Done\n\n");
91
```

3.1 Start Program

To run the program, follow the steps below.

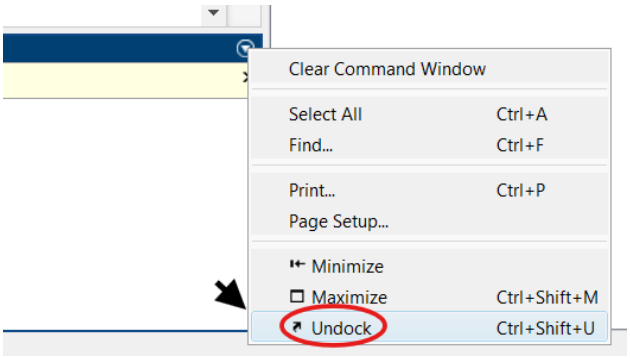
1. Open `audio_steganography.m` in MATLAB.

   Note: make sure the current folder matches the location of the program files

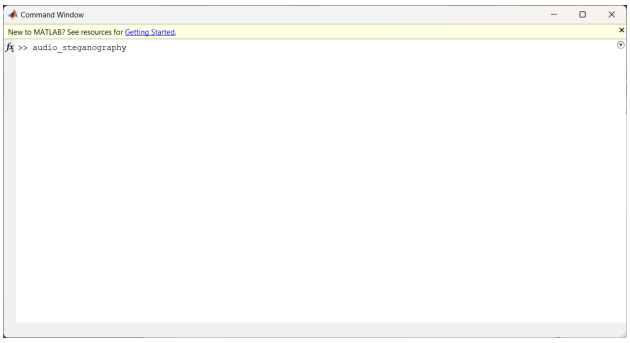


2. Click on the little arrow in the Command Window.

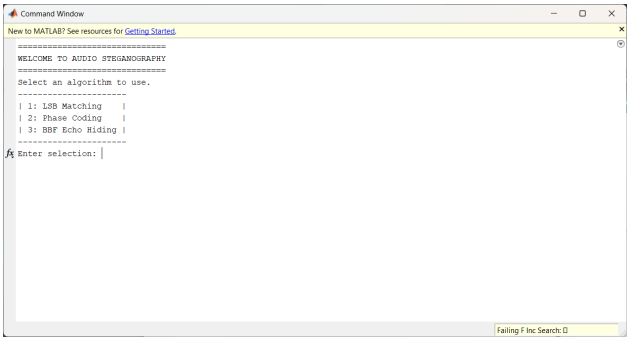
3. Select Undock.



4. In the Command Window, type `audio_steganography`.



5. Hit Enter to begin the program.

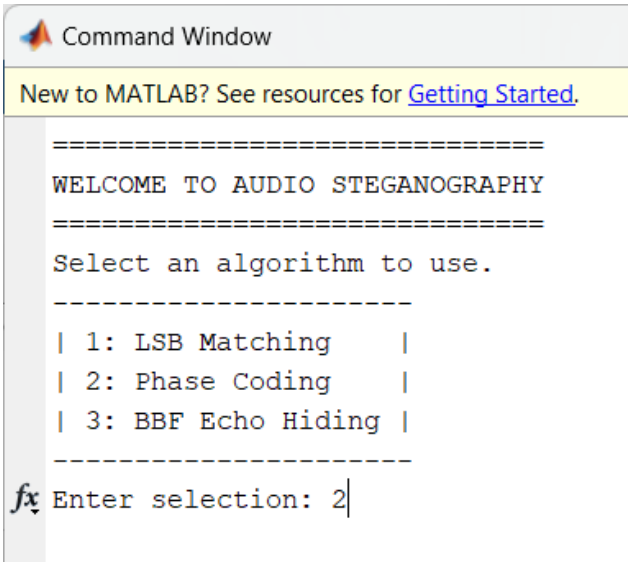


3.2 Encryption Process

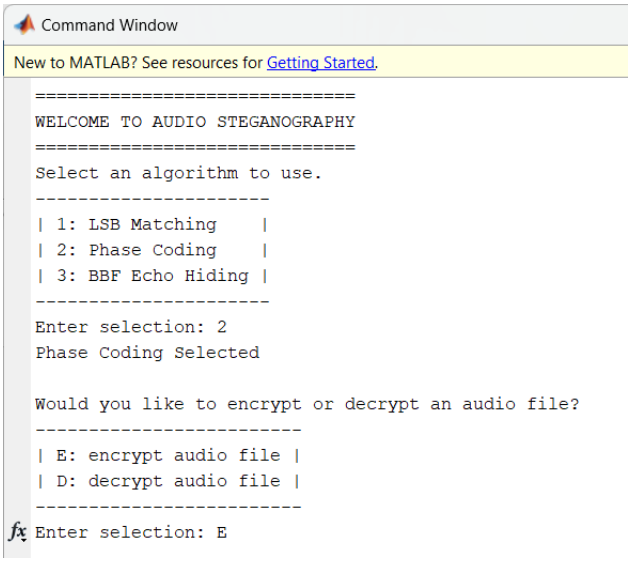
To embed a text message into an audio file, follow the steps below. This guide follows the procedure for encrypting with phase coding, but the execution steps are identical for each algorithm.

1. Select a steganography algorithm to use for encryption and hit Enter.

   Note: '1' for LSB Matching, '2' for Phase Coding, '3' for BBFEH

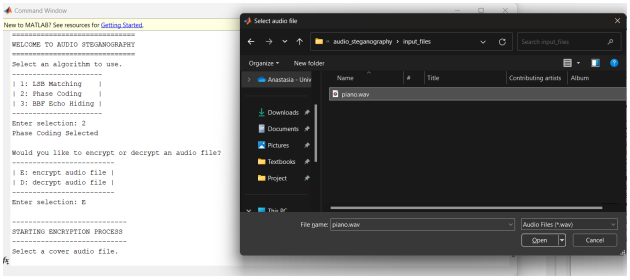


2. Type 'E' to select encryption and hit Enter.

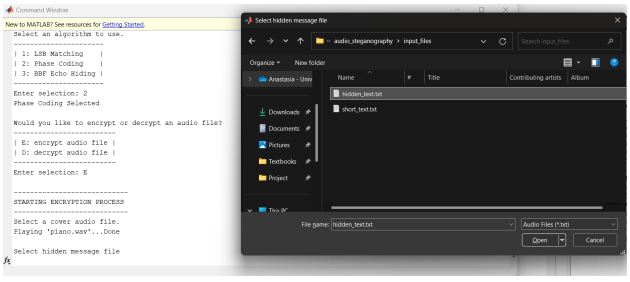


3. Select a cover audio for embedding.

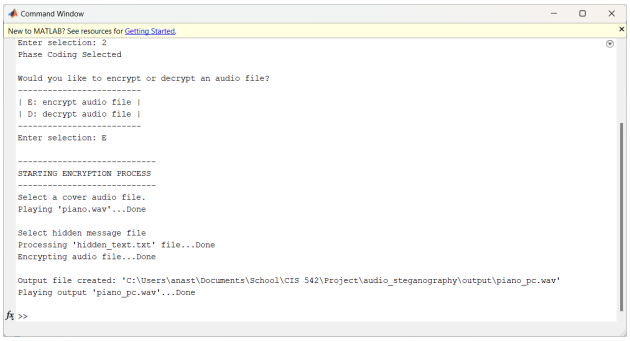
   Note: phase coding and BBFEH only support `.wav` *file type.*



4. Select a `.txt` file with the secret message to embed.



5. When encrypting completes, the output audio file name and location will print in the Command Window.



When the selected input file with the secret message exceeds the maximum allotted characters, the program prompts the user to select a different input file. You can change the program to instead clip the input message to the maximum allotted length by changing the `CLIPPING` variable to `true` in line 10.

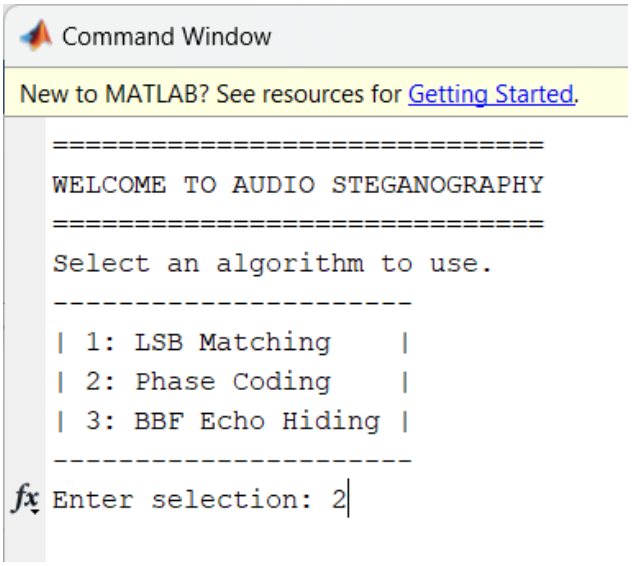
```
6   close all; clear variables; clc;
7   lsb = LSBMatchingContainer;
8   pc = PhaseCodingContainer;
9   bbfeh = BBFEchoHidingContainer;
10  CLIPPING = true; %true=clip input, false=reselect input
11
```

3.3 Decryption Process

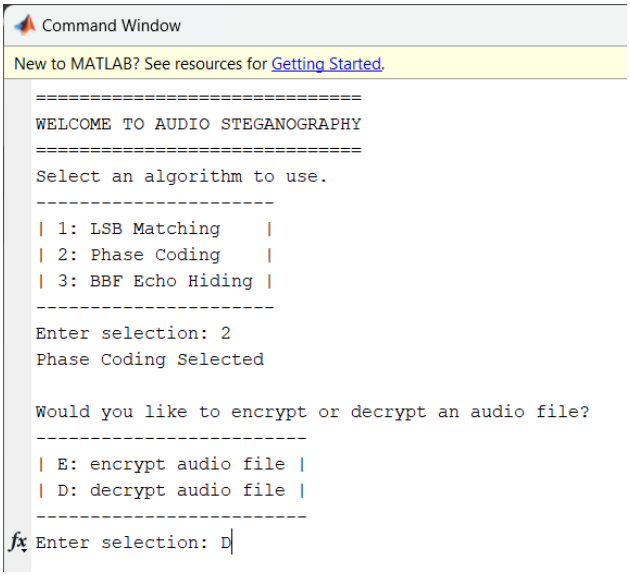
To extract a text message from an audio file, follow the steps below. This guide follows the procedure for decrypting with phase coding, but the execution steps are identical for each algorithm.

1. Select a steganography algorithm to use for decryption and hit Enter.

   Note: '1' for LSB Matching, '2' for Phase Coding, '3' for BBFEH

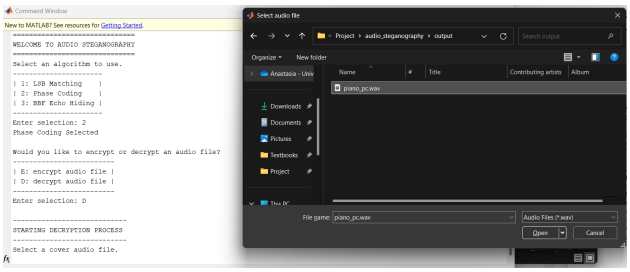


2. Type 'D' to select decryption and hit Enter.

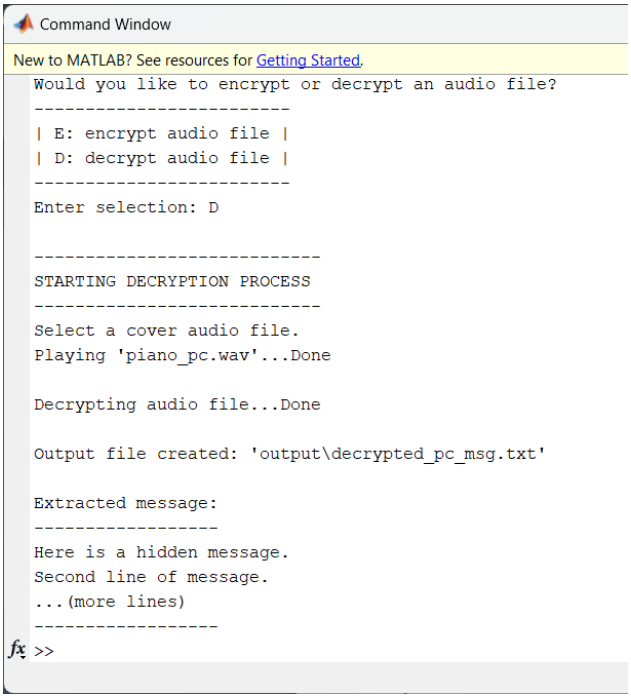


3. Select the encrypted audio file to extract a message from.

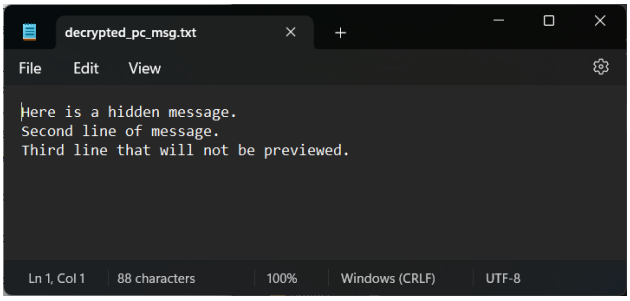
   Note: file located in `/output/` *folder. Full path will be printed in Command Window during encryption process.*



4. When decryption completes, the `.txt` output full file path and the first two lines of the decrypted message will print in the Command Window.



5. The output `.txt` file contains the full decrypted message.



4 Troubleshooting

Although not very common, the decryption process sometimes outputs a message with an incorrect character. Unfortunately, this is due to bit errors when writing the signal to a `.wav` file with MATLAB's `audiowrite` function and/or when sampling the signal with MATLAB's `audioread` function. The phase coding and BBFEH procedures include Hamming error correcting to resolve single bit errors, but when a character experiences multiple bit errors, the program cannot rectify it. Resolving this requires a more advanced error correction algorithm, and thus cannot currently be resolved when experienced.