

wavEdit: A shell program to encode text into a wav soundfile and to extract the text from a soundfile via Steganography.

wavEditPanel: A simple GUI for wavEdit.

Octopus: the implementation of the encoding/decoding algorithm.

WavEditPanel:

The screenshot shows a Windows-style application window titled "Form1". It contains two tabs: "tabPage1" (selected) and "tabPage2". The main area is divided into several sections:

- Code File:** A text box containing "C:\snds\Examples\code.txt" with a red "1.)" label and a button "1a.)".
- Sound File IN:** A text box containing "C:\snds\Examples\input.wav" with a red "2.)" label and a button "2a.)".
- Sound File OUT:** A text box containing "C:\snds\Examples\output.wav" with a red "3.)" label and a button "3a.)".
- Seed:** A text box with a red "4.)" label and an "Auto Set" button with a red "4a.)" label.
- Residual / Saltiness:** A text box with a red "5.)" label and a "%" symbol.
- Block Length:** A text box with a red "6.)" label.
- Displacement:** A text box with a red "7.)" label.
- Program Location:** A text box containing "C:\snds\Projects\wavEdit\Debug\wavEdit.exe" with a red "8.)" label and a button "8a.)".
- Area 9.:** A large empty text box with a red "9.)" label.

At the bottom, there are three buttons: "Automatic" (red "10."), "Encode" (red "11."), and "Process" (red "12.).

Parameters:

1.) - 7.) are required for encoding/decoding

8.) has to be set to the location where wavEdit will be called. If you installed the snds directory at c:\ the defaults will be fine.

11.) has to be clicked to change the direction of the transfer (it is symmetrical) between

‘encode’ which integrates/hides the text file in 1.) into 2.) and

‘decode’ which extracts the hidden information from the sound file into the content of 1.)

and appends ‘.out’. to the filename.

12.) Executes the transformation depending on 11.)

- 1.) The address of the file containing the data to be embedded. It does not have to be plain text but it cannot contain one character 0x00 since this marks the end of the source within the embedding file. Double clicking within or ‘pushing’ the 1a.) button opens a file selector.
- 2.) The address of the source wave file which will have the code file (1.)) embedded. Double clicking within or ‘pushing’ the 2a.) button opens a file selector.
- 3.) The address of the file which will contain the code file with the source file embedded. Both files have the same size. Double clicking within or ‘pushing’ the 3a.) button opens a file selector.
- 4.) The first of the 4 part key (the seed for the PRNG). ‘Pushing’ the 4a.) button presets 4.) with a random seed.
- 5.) The percentage of unchanged values reducing chances that statistical methods can detect that there is hidden information. This also shrinks the amount of information hide-able in the wav file by the same percentage.
- 6.) The amount of characters that will be encoded in one block. Each block contains there very own set of distribution. This means that even if the first block was decoded via brute force the attacker has to start the next one again with a new attack. The attacker has also no idea how long the blocks are making it harder to even use brute force methods.
- 7.) The displacement adds this number to each distribution value modulus block length. So, even knowing the seed will not work as long as the displacement and block length are not known.
- 8.) The location of the wavEdit.exe. It will be called when the ‘Process’ key was pressed.
- 9.) The command line that will be used to call wavEdit.exe including the parameters.
- 10.) Sets parameters 4.) – 7.) randomly helping for the encoding. There is no need to use the feature if both parties have agreed prior on a certain set of parameters but it helps keeping others from guessing birthdays, ages, etc. that might have been used.
- 11.) Sets the mode to ‘encode’ or ‘decode’. Encoding means integrating/hiding the code Text (1.)) into the source wave file (2.)) while decoding extracts the code text from the file.
- 12.) Starts the transformation.