Steganography Tool Specification

Product vision

The *New Software Application* (NSA) company expects the delivery of two tools that leverages the *Least Significant Bit* (LSB) method to implement steganography:

- A *dissimulation* tool, able to hide a message into an existing image;
- A *revelation* tool, able to extract a message from an existing image.

These two tools must be assessed with the definition of benchmarks measuring the execution time with respect to the size of message to be encoded. The tools are delivered as plain commands (no GUI). One can configure the way the message is dissimulated or revealed using command-line arguments.

Technical information

To prevent putting all their eggs in the same basket, the NSA asks you to develop the tools using two different technological ecosystems. One of the tools must be defined with object-oriented concepts (using the Java ecosystem) and the other one must be defined using procedural concepts (using the C ecosystem). When the tools reject an element, the program must be ended with an exit code different than 0. As the tools will be executed automatically, respecting the described interface is mandatory. When numbers must be rounded, it is always using a floor rounding.

The servers run a Linux distribution (Ubuntu 14.04.3 LTS), on top of 64 bits architectures. The version of java available on the server is Java 8 (using the Oracle implementation of the language). The version of the C compiler is gcc 4.8.4.

Dissimulation tool

Global overview

The dissimulation tool is used to hide a message (text or file) inside an existing image. It supports classical image format as inputs (i.e., JPEG, PNG, BMP, PGM and PPM) and outputs (excepting JPEG). The tool will reject an unsupported format. The algorithm used to hide the message inside the picture can be controlled: color channel impacted, numbers of bits impacted in a channel and pattern of dissimulation to be used. The tool computes metrics, including the time consumed to hide the message, the correlation between the input and output images (using the histogram comparison method or the template matching one) and the number of pixels impacted by the dissimulation process. The message is stored in the image and postfixed by a magic number used to identify the end of the message. The tool will reject a message that contains the magic number. A message that does not fit in the image will also be rejected.

Command-line interface

The tool is delivered as a command named "**`dissimulate`**". It accepts the following arguments (in any order) on the command line:

- `-Fin`: the flag allows the user to specify the file format for the input image.
  - Accepted format: JPEG, PNG, BMP, PPM and PGM
  - If not specified, the tool detects the format automatically.
- `-Fout`: the flag allows the user to specify the file format for the output image
  - Accepted format: PNG, BMP, PPM and PGM
  - If not specified, the tool detects the format automatically.
- `-in`: specify the input file to use as image container
  - This argument is mandatory
- `-out`: specify the output file where to store the image containing the message
  - If not specified, the output file is a PNG file named "result.png" in the current directory
- `-msg`: specify the message to be used
  - This argument is mandatory
  - If the argument is a double-quoted string, it is used as a textual message
  - Otherwise, it can be valued with "stdin" to read from the standard input or with an existing file name to use the contents of this file as message.
  - If the argument value is invalid (e.g., un-existing file, too big for the container), the tool will reject it before starting the dissimulation process.
- `-b`: specify the number of bits to be used when hiding the message
  - If not specified, the default value is 1
- `-c`: the channels, as a coma-separated list
  - The acceptable channels are Red, Green, Blue, Alpha and Gray.
  - Default values are Red,Green,Blue for colored image and Gray for gray-based ones.
  - The tool will reject a channel that does not exist in the image container.
- `-p`: specify the pattern to be used to hide the image
  - "direct" stores the message line by line, starting by the first line
  - "reverse" stores the message line by line, starting by the last one
  - "external_spiral" stores the message using a clockwise spiral pattern, starting in the top-left corner and following the external border of the image
  - "internal_spiral" stores the message using a clockwise spiral pattern starting at the middle of the picture, and then going top, then right, …
  - The default value is "direct"
- `-magic`: define the magic number to be used to identify the end of the stored message.
  - By default the magic number is the following hexadecimal numbers: `48 45 4C 50`
- `-metrics`: asks the tool to display metrics on the standard output
  - histogram: correlation between in and out, using the histogram method
  - template: correlation between in and out, using the template matching method
  - time: time used to process the message, in milliseconds
  - impact: numbers of pixels impacted by the process

Revelation tool

The revelation tool (command **`reveal`**) is used to extract a message from an image, hidden with the dissimulation tool. It accepts the following arguments:

- `-Fin`: the flag allows the user to specify the file format for the input image.
  - Accepted format: PNG, BMP, PPM and PGM
  - If not specified, the tool detects the format automatically.
- `-in`: specify the input file to use as image container
  - This argument is mandatory
- `-out`: specify the output file where to store the message
  - If not specified, the tool uses the standard output
- `-b`: specify the number of bits to be used when hiding the message
  - If not specified, the default value is 1
- `-c`: the channels, as a coma-separated list
  - The acceptable channels are Red, Green, Blue, Alpha and Gray.
  - Default values are Red,Green,Blue for colored image and Gray for gray-based ones.
  - The tool will reject a channel that does not exist in the image container.
- `-p`: specify the pattern to be used to hide the image
  - "direct" stores the message line by line, starting by the first line
  - "reverse" stores the message line by line, starting by the last one
  - "external_spiral" stores the message using a clockwise spiral pattern, starting in the top-left corner and following the external border of the image
  - "internal_spiral" stores the message using a clockwise spiral pattern starting at the middle of the picture, and then going top, then right, …
  - The default value is "direct"
- `-magic`: define the magic number to be used to identify the end of the stored message.
  - By default the magic number is the following hexadecimal numbers: `48 45 4C 50`

Acceptance criteria

- The tools work as a couple. There is no value associated to the delivery of a dissimulation tool that can hide messages the revelation one cannot extract.
- The tools use standard interfaces, meaning that the revelation tool from team X must be able to extract a message hidden by the dissimulation tool of team Y.
- The minimal viable product associated to these tools are the tools that support all the default values and error detection mechanisms.

Deliveries

- Your GIT repository must contain 2 directories: **`dissimulation`** and **`revelation`**.
- Each directory must contains a build.sh script that will support the compilation of the source code on the NSA server

END OF DOCUMENT.