CS 340: Cyber Security

Steganography (30 points)

DUE: End of day Friday, February 20, 2015

Steganography is the art of hiding communication. The idea is to hide a message inside some cover media without modifying the perception of the media. As an example, many image files contain redundant information. Altering this information does not affect the visual appearance of the image. This fact can be used to create a steganographic system. By embedding a message into the redundant bits of an image, we can send secret messages by posting the images on the web. However, Steganography is not limited to only images. You can easily embed information to various files and formats.

**Your task:**

1. Research Steganography and the various implementations.
2. Find an algorithm (or devise your own algorithm) to perform Steganography on a commonly known file format other than an image. For example, you can use MP3, WAV, PDF, Word documents, Excel spreadsheets, etc.
3. Implement the program to embed and extract the information out of the file. Do not use existing code and/or library other than the standard Java API.
4. The focus is to embed plain text information into a file. Remember that the functionality of the original media file should not be affected i.e, an mp3 file should still play normally, a word document should still display the correct text, etc.
5. (10 points) Embedding information into a media file. Includes but not limited to: original file functionality not affected, no discernible change in file size (it should be proportional to the embedded message), checking whether the file size is sufficient to embed, etc.
6. (10 points) Extracting information from a media file. Includes but not limited to: original message can be extracted, no extraneous characters, etc.
7. (5 points) Your program should also be able to embed and extract information other than plaintext **using the same program** (an image, audio file, text and image, etc). Your program should also manage embedding the file extension and adding the proper extension to the extracted message file.
8. (5 points) Write a short and concise report (2 pages max) on how you approach the problem, explain the algorithm used, describe some of the methods developed and show some examples. The idea of the report is to provide an overview of your algorithm and the program without having to go through the source code. If you are implementing someone else’s algorithm, you must add reference(s) to the original algorithm.

**Requirements & Deliverables:**

1. Your program must be written in Java.
2. Your program should ask whether it wants to insert or extract information.
3. The program must be able to embed a plain text file and extract the text into a plaintext file.
4. If successful, the program should be able to embed other information.
5. Ask the user to specify the text file to be embedded / extracted into and also the file to hide the information in.
6. Make sure that you test your system to embed and extract a considerably large message (text, image, audio).
7. DO NOT hardcode critical components (file name, extension, etc), you will be penalized for this.
8. Submit your java file(s) and the report document (Word or PDF) through MUOnline. Be sure to provide detailed instructions and comment the code sufficiently on how to run your program.

S 340: Cyber Security

Steganography (30 points)

DUE: End of day Friday, February 20, 2015

Steganography is the art of hiding communication. The idea is to hide a message inside some cover media without modifying the perception of the media. As an example, many image files contain redundant information. Altering this information does not affect the visual appearance of the image. This fact can be used to create a steganographic system. By embedding a message into the redundant bits of an image, we can send secret messages by posting the images on the web. However, Steganography is not limited to only images. You can easily embed information to various files and formats.

**Your task:**

1. Research Steganography and the various implementations.
2. Find an algorithm (or devise your own algorithm) to perform Steganography on a commonly known file format other than an image. For example, you can use MP3, WAV, PDF, Word documents, Excel spreadsheets, etc.
3. Implement the program to embed and extract the information out of the file. Do not use existing code and/or library other than the standard Java API.
4. The focus is to embed plain text information into a file. Remember that the functionality of the original media file should not be affected i.e, an mp3 file should still play normally, a word document should still display the correct text, etc.
5. (10 points) Embedding information into a media file. Includes but not limited to: original file functionality not affected, no discernible change in file size (it should be proportional to the embedded message), checking whether the file size is sufficient to embed, etc.
6. (10 points) Extracting information from a media file. Includes but not limited to: original message can be extracted, no extraneous characters, etc.
7. (5 points) Your program should also be able to embed and extract information other than plaintext **using the same program** (an image, audio file, text and image, etc). Your program should also manage embedding the file extension and adding the proper extension to the extracted message file.
8. (5 points) Write a short and concise report (2 pages max) on how you approach the problem, explain the algorithm used, describe some of the methods developed and show some examples. The idea of the report is to provide an overview of your algorithm and the program without having to go through the source code. If you are implementing someone else’s algorithm, you must add reference(s) to the original algorithm.

**Requirements & Deliverables:**

1. Your program must be written in Java.
2. Your program should ask whether it wants to insert or extract information.
3. The program must be able to embed a plain text file and extract the text into a plaintext file.
4. If successful, the program should be able to embed other information.
5. Ask the user to specify the text file to be embedded / extracted into and also the file to hide the information in.
6. Make sure that you test your system to embed and extract a considerably large message (text, image, audio).
7. DO NOT hardcode critical components (file name, extension, etc), you will be penalized for this.
8. Submit your java file(s) and the report document (Word or PDF) through MUOnline. Be sure to provide detailed instructions and comment the code sufficiently on how to run your program.