Advanced JPEG Steganography and Detection

Overview:

- The JPEG Algorithm
- Exploiting the JPEG algorithm
- Detecting Exploited JPEGs

Why JPEG? - The JPEG Algorithm

- The Joint Photographic Experts Group (JPEG) algorithm combines both lossless and lossy data compression techniques
- Achieves a small file size with little perceptible difference
- Particularly suited to natural images
- One of the most common image formats if not the most common

Algorithm Overview

Colour Plane Conversion → Convert unsigned to signed → DCT → Quantiser → Run-length Encoding → Entropy Encoder

Colour Plane Conversion

- The JPEG algorithm first converts RGB to YC_rC_b
  - Y is the luminance component
  - C_r and C_b are the red difference and blue difference chroma components
  - Grayscale images only have the Y component
- The human eye is less sensitive to chrominance than luminance

Discrete Cosine Transform

- The 2-dimensional Discrete Transform is applied to an 8×8 image block
- This process breaks down the frequency components of an image

  - Low frequencies are smooth transitions in an image
  - High Frequencies are sudden changes (like in a cartoon)
- The purpose of this is to modulate the influence of different spectral components on the image
  - i.e. Higher Frequencies contribute less information to the image and therefore can be reduced or eliminated

Quantisation and Entropy Coding

- The DCT results are quantised at a desired quality level
- Entropy coding is applied
  - A combination of Run-Length Encoding and Huffman coding
- To view the image, the process is reversed
- The restored image looks similar or almost exact to the human, but mathematically it is completely different
- For a high quality JPEG, there is almost no perceptible difference

Quantisation and Coding

- An 8×8 quantisation table is used to scale these coefficients
  - This is where the greatest loss occurs
  - The result is the “quantised DCT coefficients”
  - The Q-Table controls the quality
- The quantised DCT coefficient matrix generally has a lot of ZERO values, which are Run-Length coded away
  - The remaining compression is lossless
- With the small remaining numbers, Huffman or Arithmetic encoding is applied
- Loop to the next 8×8 block and repeat until the image is complete

Exploiting the JPEG Algorithm - JPEG Hiding

Swap DCT

- Choose two DCT coefficients that have the same value in the quantisation table
    - Select middle frequencies so hidden bits are in significant portions of the image
- Select a cover block
- Get DCT transform of the block
- Read a message bit from the file to be hidden
- If the condition is already true, continue
- If the condition is not true, SWAP the coefficients
    - Note: this is done prior to quantisation, so the difference must be large enough to hold true after quantisation

Weaknesses in this approach

- A particular cover block may be a poor candidate for hiding
- Capacity is 1 bit per 8×8 block
    - For a 256×256 image, thats 32×32 = 1024 blocks max

DCT Least Significant Bit (LSB)

- JSteg uses this approach
- Alter the LSB of each quantised DCT coefficient to hold our message
- More than one bit/block capacity
    - Depends on the number of non-zero coefficients
- Can use any coefficients that are not zero or one
    - Using zero would increase capacity, but also distortion, and image size
    - A clear indication of data hiding would be a low number of DCT coefficients with a zero value
- Can't use '1' because … 0001_2 → change LSB and 0000_2

Outguess

- Hides in LSBs of DCT coefficients

- Pseudo-randomly permutes selection DCT coefficients that are not zero or one
- After embedding, a second pass is made to make sure corrections to unused coefficients ,such that DCT histogram is preserved
    - Reduces capacity as some coeffificents are used for correction
    - Makes detection more difficult

F5

- F5 takes a different approach to hiding in the DCT coefficients
- F5 has a fairly high capacity, but very low detectability
- F5 decrements the magnitude of the coefficient values when the LSB does not match the message
    - As opposed to overwhelming them with the message btis
    - Note that 1 and -1 become zero - called shrinkage
        - must be decremented to zero since a 2 will become a 1
- Skips a zero for embedding and extraction
- Inverts the meaning for negative DCT coefficients
    - An LSB of 1 in a negative coefficient represents a zero
    - Prevents uneven distribution of odd vs even coefficients
- Uses permutative straddling
    - Spreads the message over the entire image
    - Like the cryptographic spreading of the other techniques

Statistically Invisible Steganography

- SIS performs a complexity analysis of each 8×8 DCT block
- Number of non-zero coefficients must exceed a threshold or the entire block is skipped
    - threshold = 0.3 to 0.6
        - 20 to 29 coefficients out of a block must be non-zero
- Adds up different sets of coefficients to produce a sum

- If the LSB of the sum equals the messages, next block
- If not, add/subtract 1 from the largest magnitude

YASS

- Yet another Steganographic Scheme that resists blind steganalysis
- What YASS does a little differently is to select blocks larger than 8×8
  - Example: 10×10 blocks
  - Has 9 possible sub-blocks
- Out of the larger block, YASS selects an 8×8 block, performs the DCT conversion and quantisation
- Hides in those coefficients
- Must use an error correcting code since there will be some errors when converted to JPEG

High-Capacity DCT

- An adaptive DCT LSB technique
  - Hides mostly in lower and middle frequency components
  - Performs a capacity estimation
  - Adapts to different characteristics of each block
- Experimentally, the quality must be ≥ ~75 to remain imperceptible
  - At quality = 50, (the standard) visual distortion is obvious
  - At quality = 60, it is noticeable if you are looking for it

Steganalysis - Detecting Exploited JPEGs

- Three levels of defeat for steganography
  - Detection
  - Extraction
  - Destruction
- Detection ability based on embedded data size

    - If you embed a single bit into a single DCT coefficient, it is much less detectable

General Approach

- Get as much information as possible
    - Adversary’s goal
    - Tool(s) likely used
    - Types of cover files
    - Type of message(s)
- Check for tool signatures
    - Ex: modified Quantisation table
    - Specific file existence
        - If you are forensically examining a disk
    - Specific types of distortion
        - For JPEG, the typical artifact is blockiness
- Type of cover files
- Type of message(s)
    - Text
    - Encrypted/compressed
    - Other images
- Apply analysis specific to the tool or to the target cover files

Defeating JPEG Steganography

- Two general approaches to JPEG Detection
    - Analysis of DCT coefficients
    - Block edge detection
        - Blockiness
- Some techniques rely on training on clean images
- Apply Statistics

    - Histograms and Entropy
    - Chi-Squared Test

Weaknesses:

- Outguess uses excess capacity to make adjustments to DCT coefficients
    - Keeps the balance
- SwapDCT does not change the value of any coefficients
    - This analysis reveals nothing for swapDCT
- F5 mitigates changes in coefficients
    - Uses matrix encoding to reduce the actual number of changes
    - Does not substitute bits, decrements existing values, maintaining balance
- For these and other techniques, different detection methods are needed

Workarounds:

- An approach to detect F5 is to predict the histogram of the original cover image
    - F5 does increase the number of zero coefficients
- Decompress the stego image, crop it by 4 columns, and recompress using the same quantisation table
    - Spatially, an image cropped by just 4 vertical columns is nearly identical
- Apply a blurring algorithm to reduce blockiness introduced by the cropping
- Compare a predicted histogram with the stego-image histogram
- Able to calculate the approximate message length as well

Extracting JPEG Steganography

- Extraction is much more difficult than detection
- Cryptography complicates extraction
    - Doesn’t prevent detection
- Knowing the method is critical

- If you extract LSBs from a JPEG that used Swap DCT, you gain no information about the message

Destroying JPEG Steganography

- Sterilisation of data hidden in a jpeg is easy
- Could ZERO or RANDOMISE the LSBs of the DCT coefficients
    - Too hard in some instances
- Could hide another message on top of prior message
    - Similar to randomisation
    - Use the same tool if known
- Resize the iamge
    - Not in multiples of 8
    - Resize by a single horizontal column and vertical rows
    - Completely changes DCT coefficients

References

Ortiz, J. (2014, August 4). *Advanced JPEG Steganography and Detection > Advanced JPEG Steganography and Detection by John Ortiz | Class Central Classroom*. Classcentral.com; Class Central. https://www.classcentral.com/classroom/youtube-advanced-jpeg-steganography-and-detection-by-john-ortiz-185955