DEFCON 24

Steganography Workshop

About the Speaker

- 20 books (Including 3 on forensics, 5 on computer security, 1 on cryptology)
- 40 industry certifications
- Member of the American College of Forensic Examiners
- Diplomat of the American College of Forensic Examiners
- Chair of the cyber forensics board for the American College of forensic Examiners
- 2 Masters degrees
- 7 Computer science related patents including 1 on Steganography
- Over 25 years experience, over 15 years teaching/training
- Helped create CompTIA Security+, Linux+, Server+. Helped revise CEH v8. Created OSForensics certification
- Frequent consultant/expert witness

www.chuckeasttom.com

chuck@chuckeasttom.com

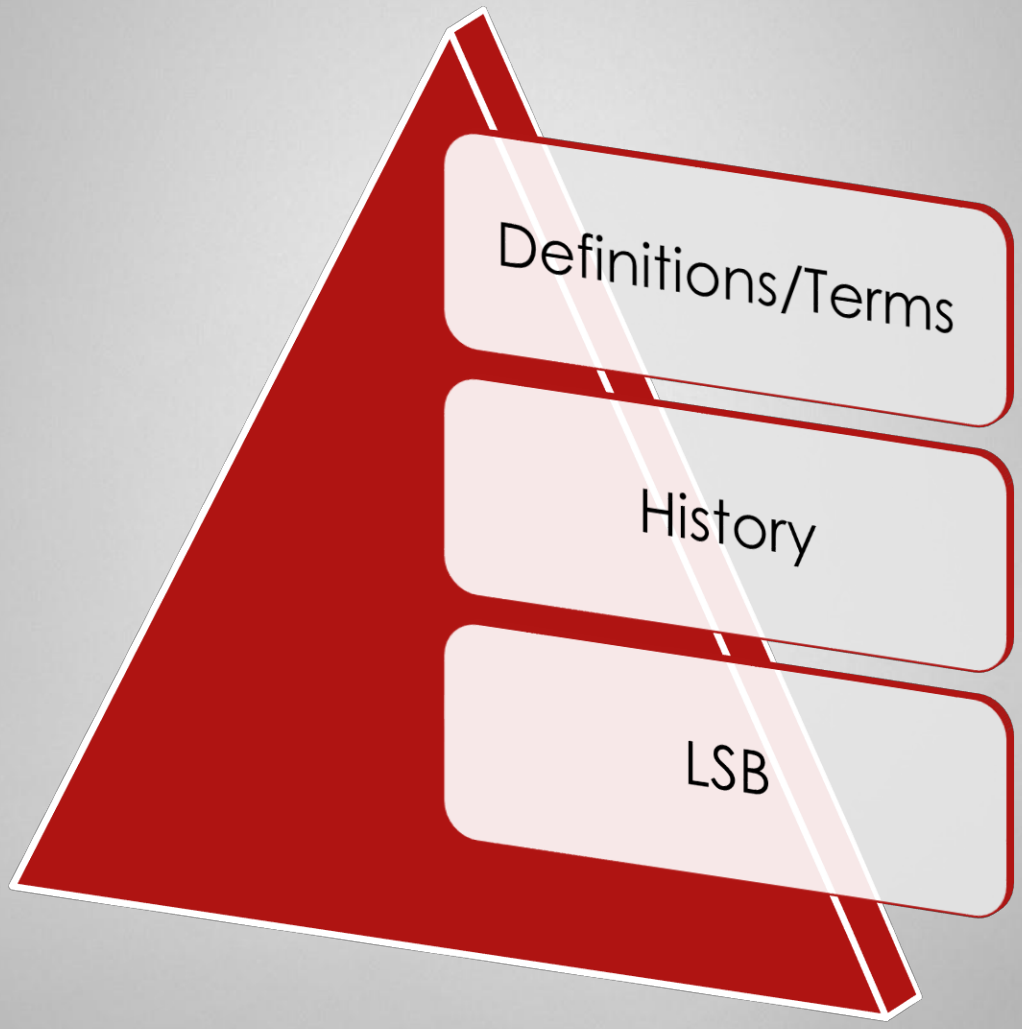
Yes feel free to email me!

What are we going to do?

What will you learn today and what will we do?

- Start with basics many of you may know
  - Use a few popular tools
- Dive into stego detection
  - Use a few tools
- Talk about variations
  - Stego file systems
  - Distributed stego
- Walk through stego code you get to take with you!
  - Yes I am going to give you stego code, walk through it, we will experiment with it, then you can take it and do whatever you wish with it.
- **LOTS OF HANDS ON**

Stego Basics



Steganography

- **Steganography** refers to any methodology used to hide a message (including text, sound, or picture) in a separate file. Most commonly text or an image is inserted into another image. However there are permutations where video is hidden in another video, or sound in sound or even sound in video. The image/sound/video that the underlying message is hidden in is referred to as a carrier or cover file or signal.

- The most common implementation of steganography utilizes the least significant bits in a file in order to store data. By altering the least significant bit one can hide additional data without altering the original file in any noticeable way.

Historical Steganography

- The ancient Chinese wrapped notes in wax and swallowed them for transport.
- Histaius wanted to communicate regarding a revolt against the Persians. He shaved the hair on one of his slaves head, then tatooed the message and after the hair grew back sufficiently, sent the slave with the message.
- A man named Demaratus was purported to have written messages regarding Xerxes impending invasion. The messages were on wax covering wood, Demaratus had a second message directly on the wood.

Historical Steganography

- In 1518 Johannes Trithmeus wrote a book on cryptography and described a technique where a message was hidden by having each letter taken as a word from a specific column.
- Morse code has been written into yarn, then the yarn sown into clothing.

Historical Steganography

- Null ciphers (unencrypted messages) have also been used also used. The real message is "camouflaged" in an innocent sounding message.

The following message was actually sent by a German Spy in WWII

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.

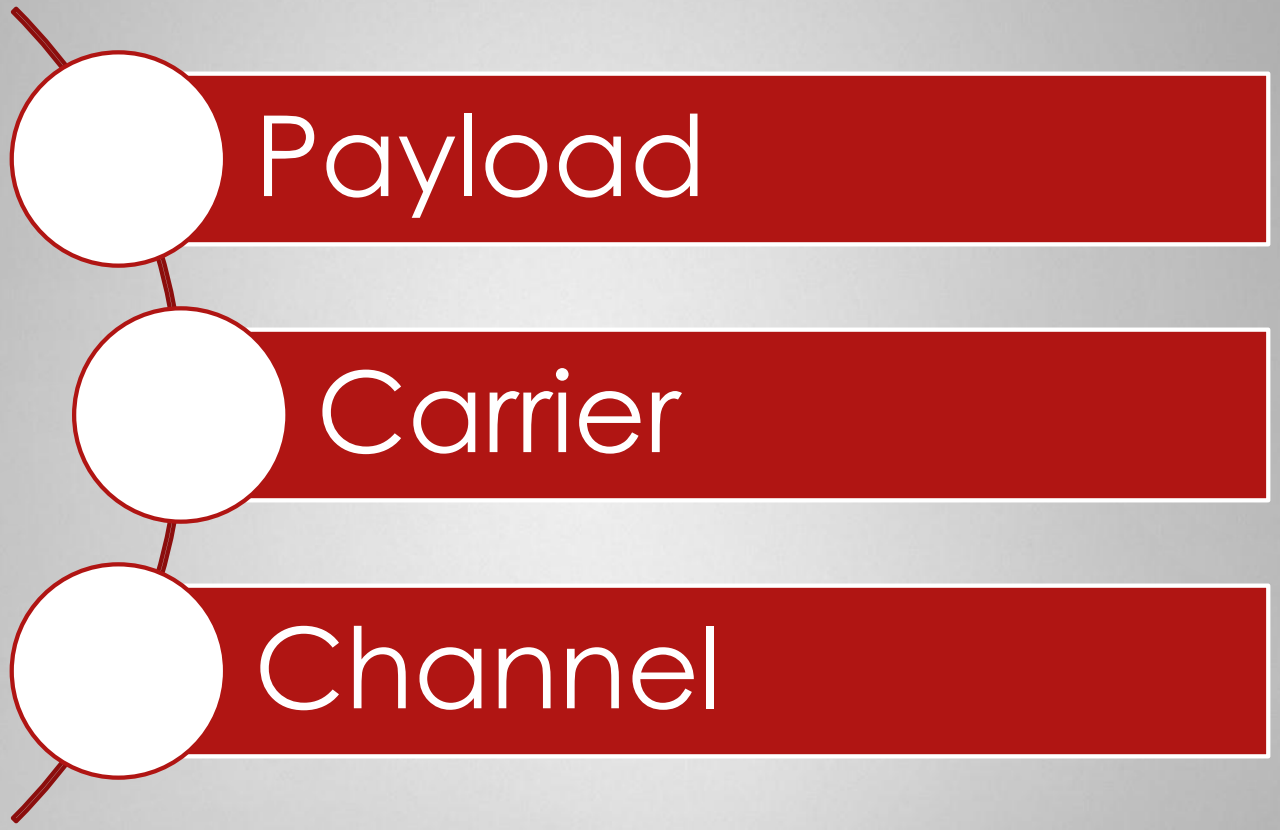
Taking the second letter in each word the following message emerges:

Pershing sails from NY June 1.

Historical Steganography - Continued

- During WW II the French Resistance sent messages written on the backs of couriers using invisible ink
- Microdots are images/undeveloped film the size of a typewriter period, embedded on an innocuous documents. These were said to be used by spy's during the Cold War.

Steganography Terms



Steganography details - LSB

- With least significant bit (lsb) replacement, certain bits in the carrier file are replaced.

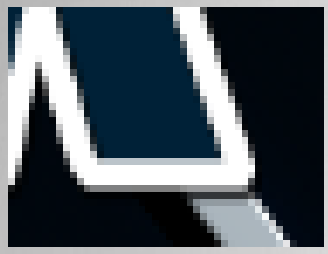
Steganography details - LSB

Before change

Red 0

Green 33

Blue 55



After change

Red 0

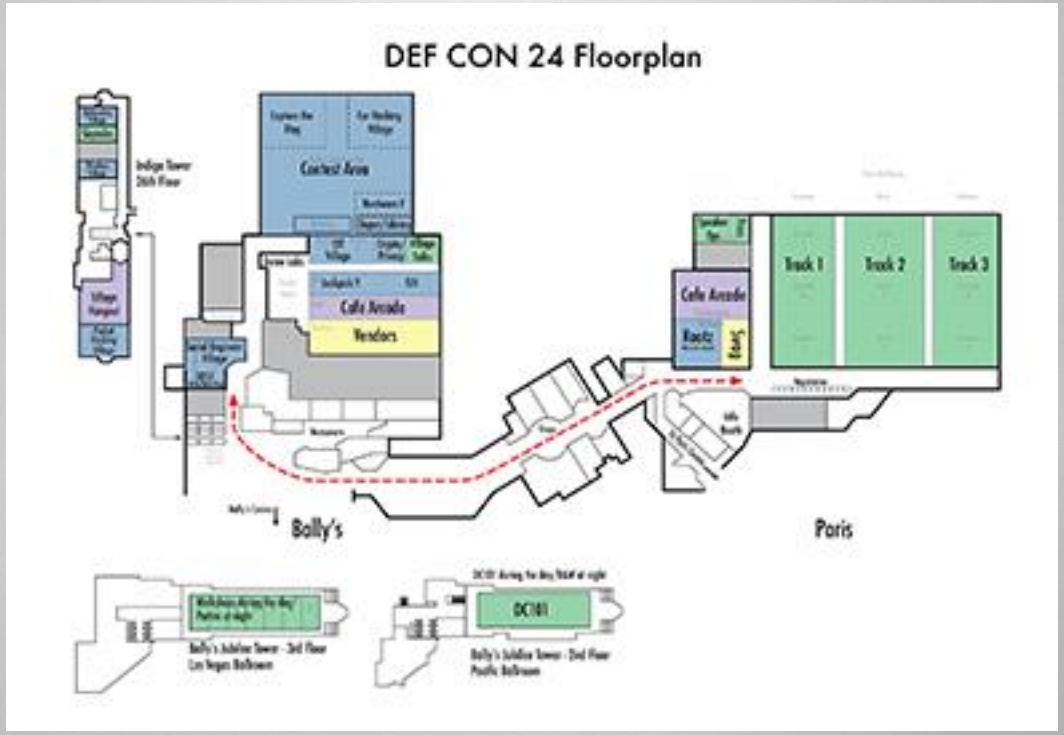
Green 33

Blue 54

The entire file has 469,032 bytes. If you change only 10% of them you can store 1 bit per byte, or 46,903 bits which is 46 KB

Steganography details - LSB

I downloaded the unaltered map of the Defcon 24 venue map. Only 62.5 KB (13% of the carrier file)



Lossy vs lossless

- Lossless algorithms that allows the original data to be perfectly reconstructed from the compressed data. Lossy (TIFF, JPEG) methods use an approximation that may lose some data.

Lossless and Lossy Compression

- **Lossless compression**
  - Reduces file size without removing data
  - Can use either Huffman or Lempel-Ziv-Welch coding
  - Winzip and PKZip both use lossless compression
- **Lossy compression**
  - As the name suggest you actually lose bits of information
  - **Vector quantization (VQ)**
    - Determines what data to discard based on vectors in the graphics file
  - LZIP uses Lossy compression

Encoding

- **Huffman encoding** is a way to assign binary codes to symbols that reduces the overall number of bits used to encode a typical string of those symbols.
- **lempel-ziv** is a lossless encoding algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch.

The Concepts of Huffman

- Huffman coding is a form of statistical coding
- Not all characters occur with the same frequency, but without proper encoding they all get the same space: 1 character gets 1 byte.

Overview of Huffman

1. Scan data and total up frequency of symbols.
2. Sort and prioritize symbols based on frequency.
3. Build Huffman code tree based on that prioritized symbol list.
4. Create a new file using the Huffman codes

Huffman Encoding

Huffman encoding is a concept from information theory. It was developed by David A. Huffman. The technique starts with a binary tree of nodes, usually stored in an array. The size of that array depends on the number of symbols. Each node can caontain the symbol itself, the frequency of the symbols appearance (called the weight), or a link to a parent node. Nodes can be 'leaf nodes' or 'internal nodes'. These nodes are arragned in a tree. For more details

https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffman_tutorial.html

Demo

- I will now walk you through a demo of Invisible Secrets and a demo of one other tool (QuickStego, OpenStego, etc.)

Lab

- Take one of the images provided to you and use any of the following tools to hide a text file in it. The text file should contain a message of at least 3 sentences
  - Invisible Secrets
  - Quick Stego
  - Open Stego
- Next examine the image, both properties and pixels to see if you can detect anything
- Then repeat but this time hide an image in an image.

Other forms of Stego

- Other forms of stego
- Stegonographic File Systems

Other forms of Steganography

- Echo Hiding: This method adds extra sound to an echo inside an audio file, that extra sound conceals information.
- Discrete Cosine Transform is often used for Video steganography. This method alters values of certain parts of the individual frames. The usual method is to round up the values.

Echo Data Hiding

- Echo introduced to hide data into audio signal
- Echo is varied with three parameters:
  - Initial Amplitude
  - Decay Rate
  - Offset
- Parameters such as amplitude and decay rate can be set below the threshold of human hearing.
- A short offset or delay makes the echo harder to detect.

Discrete Cosine Transform (DCT)

For each color component, the DCT is used to transform each successive 8x8 pixel block of the image into 64 DCT coefficients

The DCT coefficients F(u, v) of an 8 x 8 block of image pixels f(x, y) are given by this formula

$$F(u,v) = \frac{1}{4} C(u)\, C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

$$\text{where } C(u), C(v) = \frac{1}{\sqrt{N}} \quad \text{for } u, v = 0$$

$$C(u), C(v) = 1 \quad \text{otherwise}$$

The following operation quantizes the coefficients:

$$F^{Q}(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

Q(u,v) is a 64-element quantization table

How to embed

Other forms of Steganography - BPCS

- Bit-Plane Complexity Segmentation Steganography (BPCS)
- The carrier is usually a 24 bit color image (a true color image). The complex areas on the bit planes are replaced with the payload. A bit plane of any discrete digital file/signal (image, sound, etc.) is the set of bits that correspond to a given bit position. For example in 16 bit data there are 16 bit planes ranging from most significant to least significant. For 24 bit files there are 24 bit planes.
- BPCS can store much more data that LSB.

BPCS - Continued

- Break up image into regions (usually 8 X 8 pixels)
- Select a byte hide data in. The byte can be Red, Green or Blue value
- Decompose into bit planes, giving 8 bitplanes
- Now hide the data

Steganograpic File Systems

- Stores data in seemingly random files

- Proposed by Ross Anderson, Roger Needham, and Adi Shamir. Their paper proposed two main methods of hiding data: in a series of fixed size files originally consisting of random bits on top of which 'vectors' could be superimposed in such a way as to allow levels of security to decrypt all lower levels but not even know of the existence of any higher levels, or an entire partition is filled with random bits and files hidden in it.

Demo

- Audio stego demo

Lab

- Take some MP3 file of your own (or use the Vivaldi MP3 I provided) and hide a text file in it. Use a text file that has at least three sentences. Use any of the audio tools provided. You will find through experimentation that some work better than others.
- Repeat and hide the DefCon24 map in an MP3 file

Lets look at some tools

Stego Tools

S-Tools

Hermetic Stego (image)

ImageHide

JPHIDE and JPSEEK

StagaNote

OutGuess

gifShuffle

QuickStego

MP3Stegz (audio)

Our Secret

Xiao Staganography

OmniHide Pro

Masker

Stealth Files

MP3Stego

Steghide

Audiostegano

Hide4PGP

FoxHole

Data Stash

Byte Shelter

StegParty

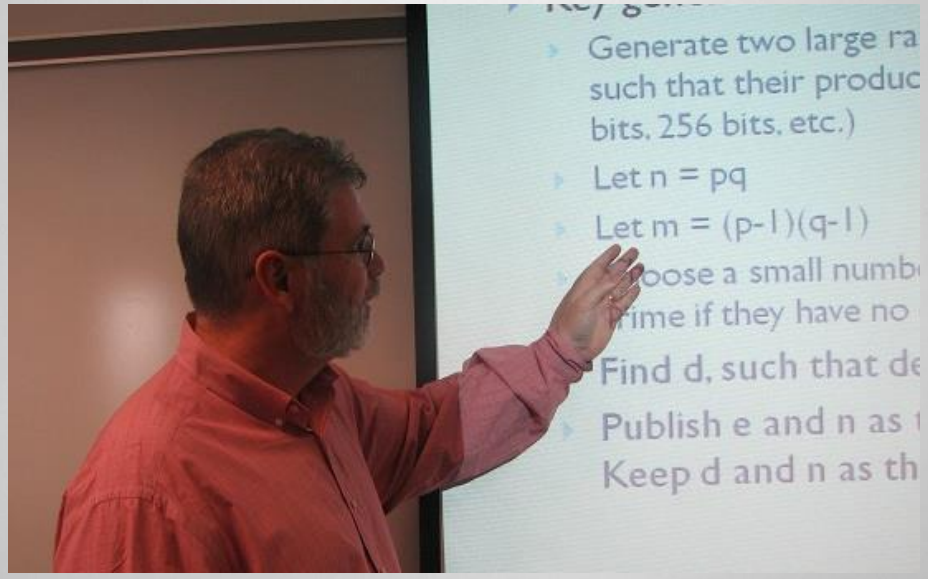
SpamMimic

StegoStick

Hide My Files

Demonstration

- I began with an image, a jpeg that was 68.3 K in size.



Demonstration Continued

- I took a text document that was 3,741 words/ 26 KB and hid it in this image using various tools.
  - Using Invisible Secrets the resulting image was 77.4 KB, an increase of 9.1 KB.
  - Using QuickStego the resulting image was a 520 KB bitmap, a more than 7 fold increase.

How to detect stego

- RQP
- Statistics
- Modification Date
- Other Techniques

Steganalysis

- By analyzing changes in an image's close color pairs, the steganalyst can determine if LSB substitution was used. Close color pairs consist of two colors whose binary values differ only in the LSB.

Steganalysis - RQP

- The Raw Quick Pair method
  - Based on statistics of the numbers of unique colors and close-color pairs in a 24-bit image.
  - Analyzes the pairs of colors created by LSB embedding
  - Countermeasure - Maintaining the color palette without creating new colors

Steganalysis – Examine Edges

- Many images and audio files use lossy compression (JPEGs and MP3's both do). In JPEG it is common that components like the high-contrast edges of black text on white background will distorted neighboring pixels, usually in a predictable fashion (edge ringing). Looking for unexpected values in the edge ringing can indicate steganography was used.

Steganalysis - Chi-Square Analysis

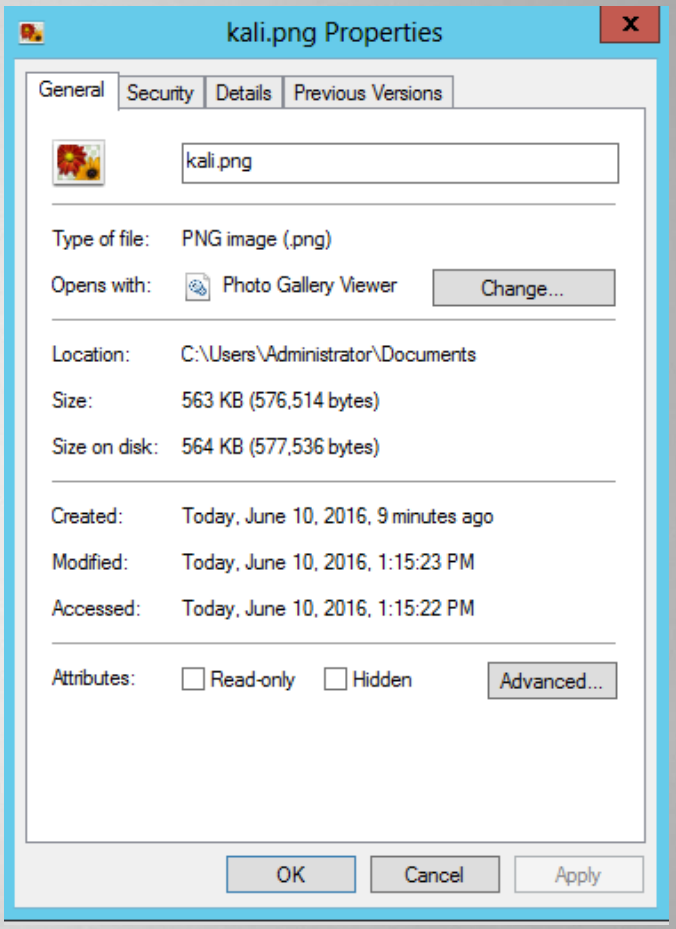
- Chi-Square Analysis calculates the average LSB and builds a table of frequencies and Pair of Values.
- Then it performs a chi-square test on these two tables.
- Essentially it measures the theoretical vs. calculated population difference

Steganalysis - Audio Steganalysis

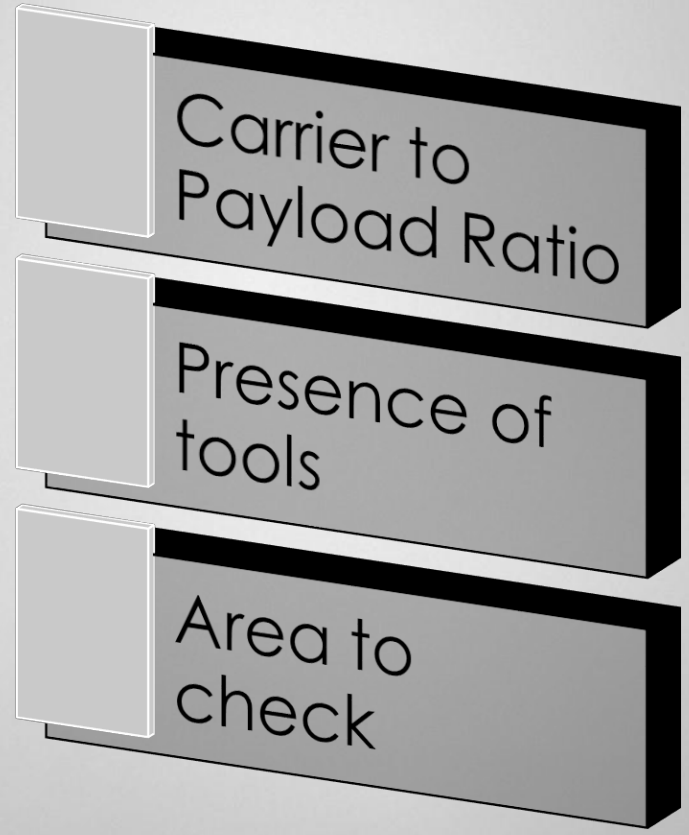
- Examine noise distortion in the carrier file.
- Noise distortion could indicate the presence of a hidden signal.

Steganalysis – Modification date

Most average users don't modify images or music files. Look for recent modification dates that do not match creation dates.



What affects steganalysis



Steganography Detection Tools

- Outguess Xdetect is one tool http://www.outguess.org/detection.php
- Steg Secret is another tool http://stegsecret.sourceforge.net/
- StegSpy has fewer limitations than StegDetect http://www.spy-hunter.com/stegspydownload.htm
- AccessData's Forensic Toolkit and Guidence Software's Encase can detect steganography.

Stegdetect

- stegdetect -t p sample.jpg
- Tries to detect the presence of embedded information in sample.jpg.
- **stegdetect** works only for JPEG images.

Stegdetect Continued

- The stegdetect utility analyses image files for steganographic content. It runs statistical tests to determine if steganographic content is present, and also tries to find the system that has been used to embed the hidden information.
- The options are as follows:
- −q Only reports images that are likely to have steganographic content.
- −n Enables checking of JPEG header information to suppress false positives. If enabled, all JPEG images that contain comment fields will be treated as negatives.
- −V Displays the version number of the software.
- −s float Changes the sensitivity of the detection algorithms. Their results are multiplied by the specified
- number. The higher the number the more sensitive the test will become. The default is 1.
- −d num Prints debug information.
- −t tests
  - Sets the tests that are being run on the image. The following characters are understood:
  - j Tests if information has been embedded with jsteg.
  - o Tests if information has been embedded with outguess.
  - p Tests if information has been embedded with jphide.
  - i Tests if information has been hidden with invisible secret

Detection Demo

- I used StegDetect on both files. It errorred out on the jpeg and could not read the bitmap.
- I used StegSpy and it discovered the bitmap stegonography but not the jpeg.

Lab

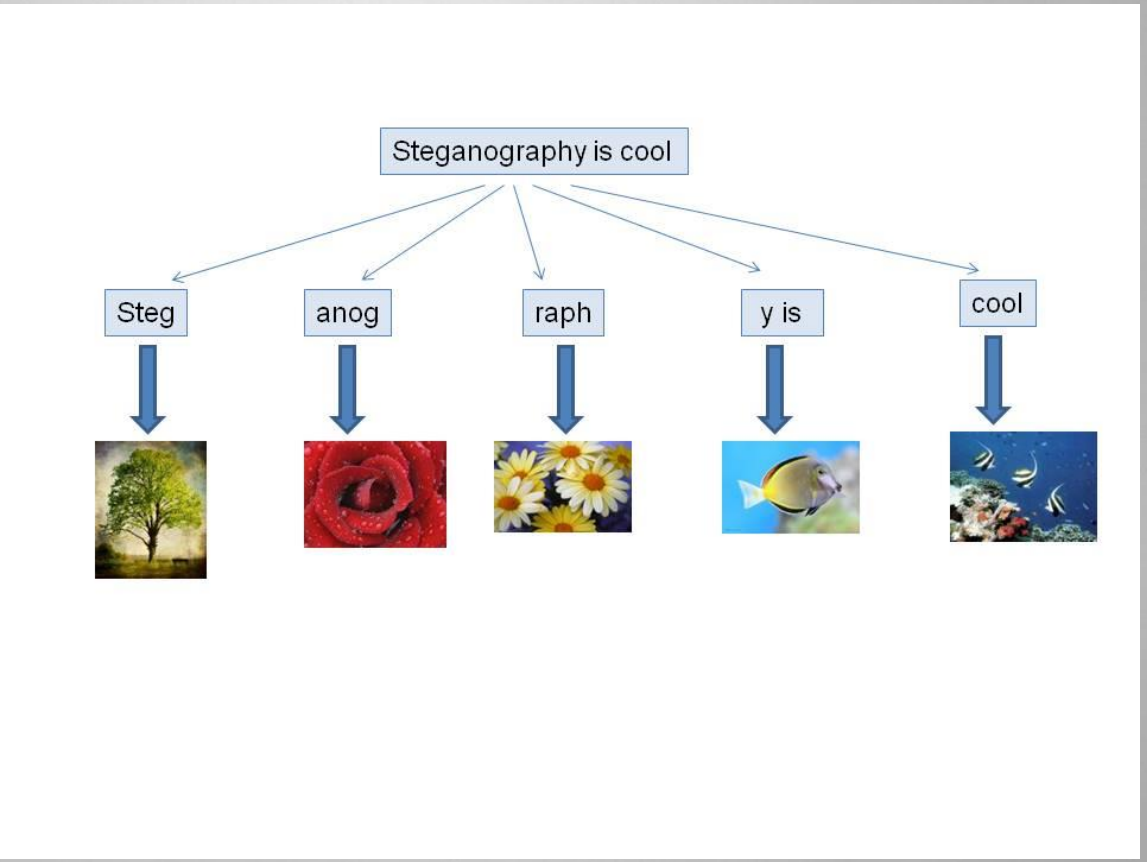
- Take one of the images you previously hid data in and exchange it with a fellow student.
- Then try to find what was hidden, or at least the fact that something was hidden.

Distributed Steganography

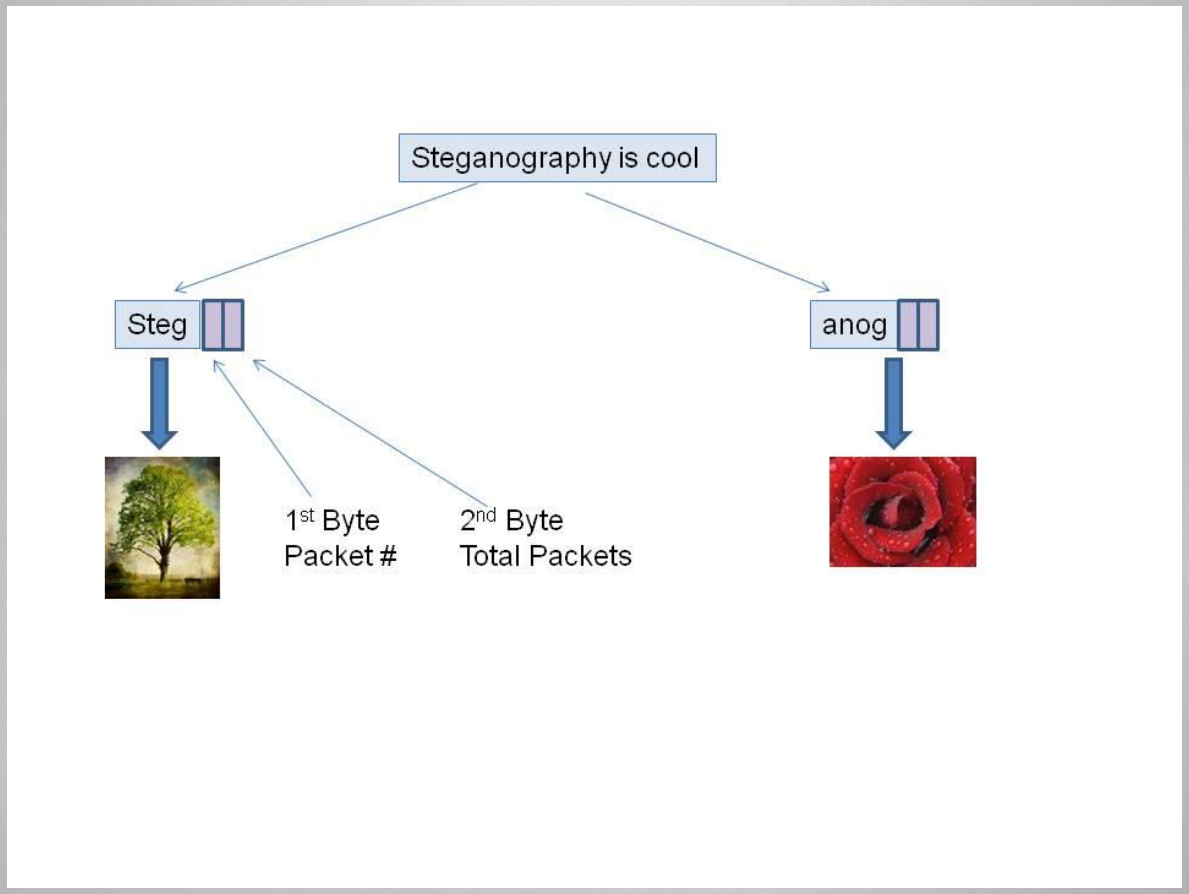
- How it works
- How it is meant to be used

Distributed Steganography

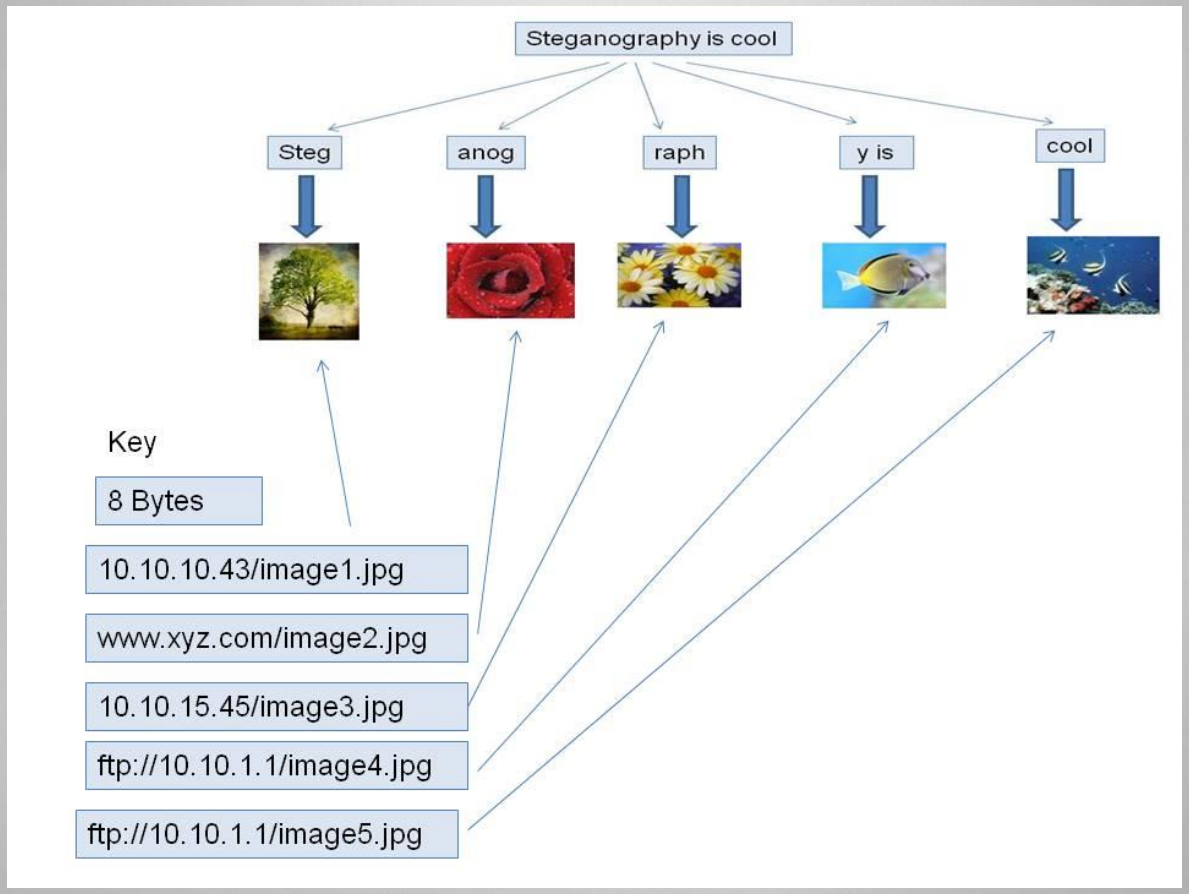
U.S. Patent No. 8,527,779 B1



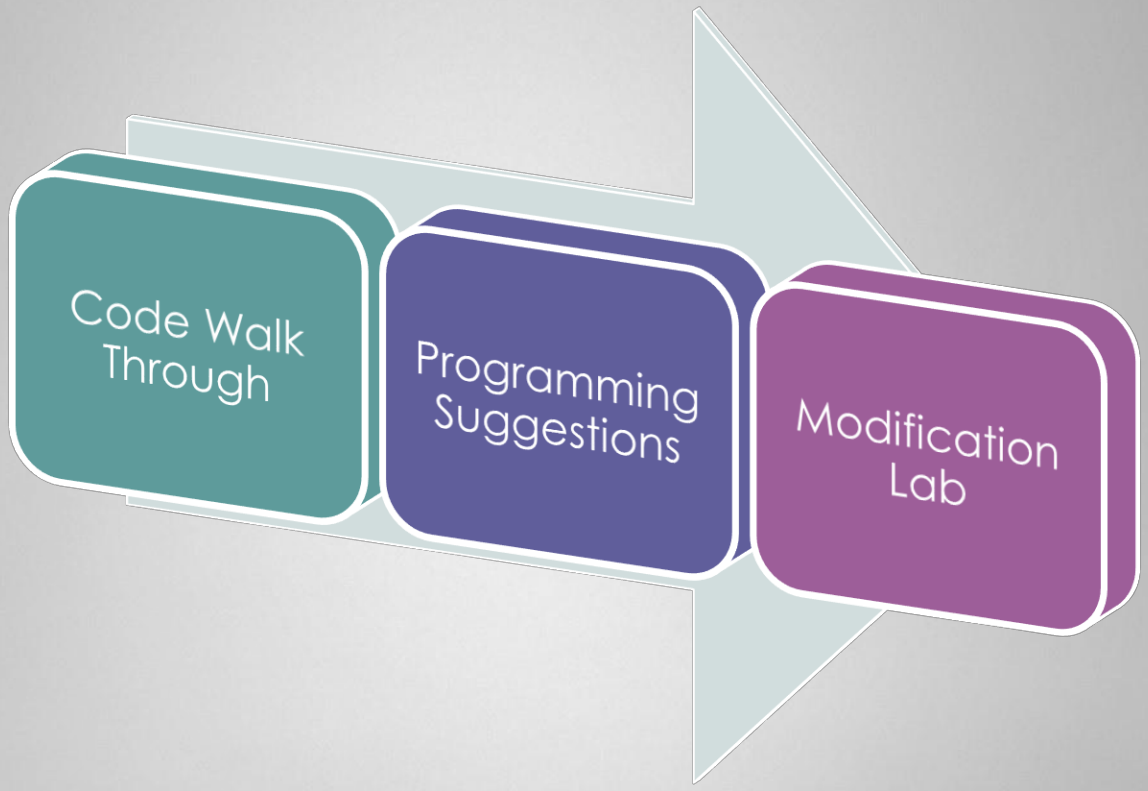
Distributed Steganography (Continued)



Distributed Steganography (Continued)



Lets look at source code!



Applications of Steganography

Covert Communications

Copy Control/Watermarking

Fingerprinting (traitor tracking)

Ownership Assertion

Additional Resources

- A white paper on a steganographic method http://www.scribd.com/doc/3367439/Steganography
- SANS paper on stego
- https://www.sans.org/reading-room/whitepapers/stenganography/steganography-past-present-future-552
- Technical paper on steganography http://www.jjtc.com/stegdoc/
- http://www.ws.binghamton.edu/fridrich/Research/acm_2001_03.pdf
- Detection of LSB steganography http://www.cecs.uci.edu/~papers/icme06/pdfs/0001377.pdf
- Detection of audio steganography http://www.ece.ucdavis.edu/~yliu/pub/papers/Tracy_ISC08.pdf