More Advanced Steganography

INSTRUCTOR: JOHN ORTIZ
SENIOR COMPUTER ENGINEER
UTSA
STEGO@SATX.RR.COM

INTRODUCTION TO STEGANALYSIS

Introduction to Steganalysis

- Terms and Definitions
- Approaches
- Tools
- Techniques

Steganalysis

- Steganalysis is the art of uncovering hidden messages and/or rendering them useless
  - Could be detection, extraction, or destruction
- Any cover can automatically be manipulated with the intent on destroying any hidden data
  - Ability to detect that hidden data exists can reduce processing time
- Steganalysis has applications to
  - Cyber warfare
  - Computer forensics/cyber crime
  - Digital traffic analysis

Steganalysis

- Every image is composed of its visually significant and visually insignificant portions
  - Applies to audio and video as well
- Separating the two components opens up an avenue of attack
  - The message must be in the visually insignificant portion
  - Reduces the amount of data to be analyzed

Steganalysis

- Stego-Only: Steganalyst only has access to the stego-object
- Multi-Stego: Multiple stego-objects are known
- Known Cover: The original cover object is known as well as the stego-object
- Known Message: Message and stego-object are known
- Cover Message Stego: Cover, message, and stego-object are known
  - What is left to do?
  - Perhaps *proving* the message is there.
  - Determining the algorithm

Steganalysis

- Chosen Stego: The steganography algorithm is known as well as the stego-object
- Chosen Message: The steganography tool is available – can generate unlimited stego-objects
  - Some ambiguity in actual meaning
- Known Stego: The tool, cover, and stego-object are known

Steganalysis

- Even in optimal circumstances, detecting or extracting a message may be difficult
- Sometimes, it's easier to attack the password rather than the system itself
  - Some people just do not know how to make good passwords!
  - Some people can be bought or tricked
  - For others, a gun pointed at their head is enough to convince them

Steganalysis

- Avenue of attack is dependant on information available
- Many available steganography algorithms leave distinct signatures
  - Remember Chang's algorithm with the modified Q Table?
- Some signatures may be in the output file format, others may be in specific statistics
  - Signatures may be dependant on technique or the specific tool

Steganalysis

- Blind steganalysis is another name for a stego-only attack
  - No information about the tool is available
- Any attack without any knowledge of the steganography system is challenging for detection and extraction
- It is much easier to destroy a message than detect or extract it
- The more information available to the steganalyst, the higher the likelihood of success

Steganalysis – A General Approach

- Get as much information as possible about the technique, the tool, covers, messages , all tools easily accessible, the goal of the hiding, and people using it
  - Knowing the general technique can greatly reduce the analysis
  - Having the actual tool available is even better
  - Knowing the general types of covers used can also reduce analysis time
  - Knowing the messages ease the task as well
    - Text?, Images?, Sounds?, Video?, Executables?

Steganalysis – A General Approach

- Use partial information to obtain more information
  - Example, by using the tool and embedding many representative messages, patterns may emerge
- Know your own goal and tailor your approach
  - Detection
  - Extraction
  - Destruction
- How much distortion of the stego-object is tolerable
  - If you are trying to destroy a message without your opponent knowing it was destroyed, your technique will be different than if that is not a concern

Steganalysis – A General Approach

- Knowing the technique will narrow the scope of what you're looking for
- Example, if the technique is LSB, regardless of the tool and how sophisticated the encryption
  - Destruction and/or removal are trivial
  - Extraction of the bits is trivial, may be more difficult to determine message
    - May require a cryptographic attack
- If the technique is to hide in the DCT coefficients, the task is more difficult
  - Converting formats and/or recompressing at lower quality will likely destroy the message

Steganalysis – A General Approach

- If the technique is echo hiding, add your own negative echo
  - Won't be exact since echo of modified audio is different than echo of original cover, but may be enough to prevent the recipient from decoding the message

Steganalysis – A General Approach

- Knowing the specific tool is a more specific variation of knowing the technique
- Can tailor approach based upon any signatures the tool leaves
  - Don't know any signatures? Find some! Make numerous stego-objects and look for patterns
  - Examine the source code if available
  - May be perceptible or statistical patterns
- Palettes may be in certain orders, or have duplicate colors, or even close colors
- For one DCT technique we looked at, randomly swap the matching Q-Table coefficients

Steganalysis – A General Approach

- Knowing the types of covers used can help in a number of ways
- Can eliminate types of covers that are not used
  - Palette-based formats (gif) will likely not carry DCT-based messages
    - Hmmm, sounds like an interesting project …
  - Echo hiding works with sound files, not images
  - StegExe works with executables, dynamic link libraries, screensavers only on a Windows platform
- Can do a statistical analysis of similar covers and determine standard deviation, variance, histograms, etc. to determine what's normal and what's not

Steganalysis – A General Approach

- How well does this file compress?
  - You may know a typical compression range, but some images fall outside
- A statistical analysis can also aid in message recovery
  - Not just what is outside the norm but why

Steganalysis – A General Approach

- Knowing the types of messages makes searching easier since you know what you're looking for
- Can modify your approach based upon statistics of the message
  - English text, especially with punctuation, has very well-known and specific properties
  - Natural pictures and sounds do too
  - Voice will be vastly different than classic or rock music
  - Pictures originally saved as JPEGs have unique, recognizable characteristics too

Steganalysis – A General Approach

- Knowing the readily available steganographic tools is beneficial
- Both the book and some other papers basically discuss approaches based on a particular tool
  - A database of signatures could be developed so that a specific tool could be identified quickly
  - Many adversaries will use some tool right off the Internet
    - Sometimes, if source code is available, it may be slightly modified

Steganalysis – A General Approach

- Knowing the goals of the adversary narrows the scope as well
  - Capacity? Robustness? Security?
- Do you expect a lot of information or a lot of redundant information?
- If the goal is robustness, your goal may be destruction
- If the goal is security, detection may define success

Steganalysis – A General Approach

- There is no silver bullet to steganalysis
- Success depends upon available information
- No perfect steganography system known to exist and it's even theorized that none will exist

Blind Steganalysis

- Artifacts
  - Visual / Audial indicators
  - Programmatic indicators
- Histograms
- Entropy
- Visualization
- Audialization

Blind Steganalysis

- Scenario: You are given a file and asked, "Is there any hidden data in the file that you can find?"
- You know nothing about the file's origin, potential steganographic tool used, type of message, etc.
- All you know is the file type
  - If it' a bitmap, you're probably not hiding in the DCT coefficients
  - Different approaches for different file types

Histogram

- We've seen examples of histograms already
- A histogram is a count of the number of times a particular value is present
- A measure of the frequency of occurrence
  - For example, suppose the following byte sequence:
  - 0, 2, 2, 3, 4, 2, 1, 2, 2, 5, 4, 4, 2
    - The number 2 appears 6 times
    - The number 4 appears 3 times
    - 0, 1, 3, 5 all appear only one time
- When examining a computer file, the possible values range from 0 to 255 (for single bytes)
- The histogram shows the relative (or exact) count of each value

Histogram

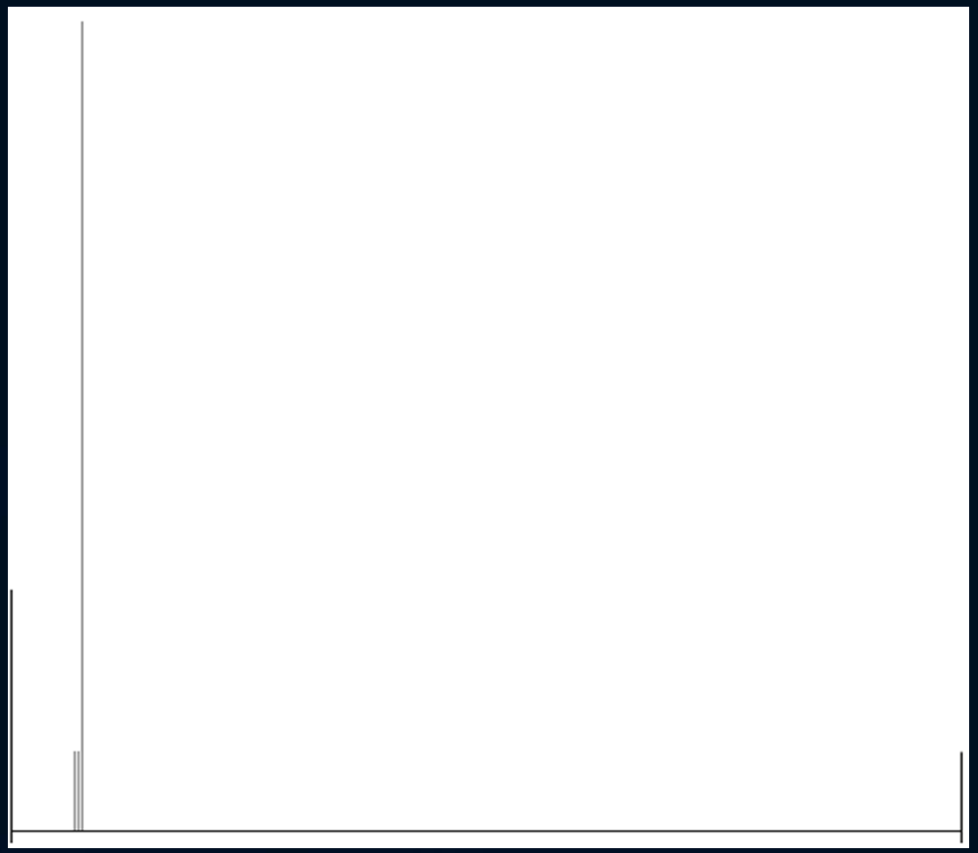
- Extremely useful for analysis of a file's contents
- Used to identify the likely data content of a file
- Many file types have unique histogram characteristics
  - There are exceptions
- An image (or audio) of the file can be useful too
  - Shows position of data file

Histogram

- The Visual Histogram tool generates
  - A bitmap histogram image of the relative counts (it is scaled)
  - A text histogram with exact counts
  - The entropy measurement
  - (optional) A visual representation of the file
- Note:
  - We have a count of how many times each byte value was present in the file
  - We know the total number of bytes (filesize)
  - Probability of occurrence ~ count/filesize

Histogram

- 00 13 13 00 11 13 13 12 13 13 13 00 FF 13 13 13
- 16-byte file
  - 00 – 3 times
  - 11 – 1 time
  - 12 – 1 time
  - 13 – 10 times
  - FF – 1 time



Entropy

- Entropy is a mathematical measure of the average uncertainty of a set of symbols
- Most often we consider bytes (values 0 – 255), as the set of symbols we care about
  - The MAX entropy is $\log_2$(#possible symbols)
  - For 256 symbols, the max entropy is 8.0000
  - For base 32 encoded files (i.e 32 symbols), the maximum entropy is 5.0000
  - Guess what the max entropy for base 64 encoded files is???

Entropy

- Entropy is a mathematical measure of the average uncertainty of a set of symbols
- Most often we consider bytes (values 0 – 255), as the set of symbols we care about
  - The MAX entropy is $\log_2$(#possible symbols)
  - For 256 symbols, the max entropy is 8.0000
  - For base 32 encoded files (i.e 32 symbols), the maximum entropy is 5.0000
  - Guess what the max entropy for base 64 encoded files is???
    - If you thought "6.0000" then you get a GOLD star!

Entropy

- P_j = probability of occurrence of a symbol
- $Lg(X) = \log_2(X)$ { 2 to what power = X }
- For byte-sized data, X = 256
- We can *estimate* the probability by counting
  - i.e. histogram
  - If symbol (byte) appears 25 times in 100-byte
  - file then P_j = 25 / 100 or 0.25

I KNOW it looks difficult, but it <u>really</u> is EASY! (Once you figure it out.)

$$Entropy = H = -\sum_{j=0}^{n-1} P_j \lg P_j = \sum_{j=0}^{n-1} P_j \lg \frac{1}{P_j}$$

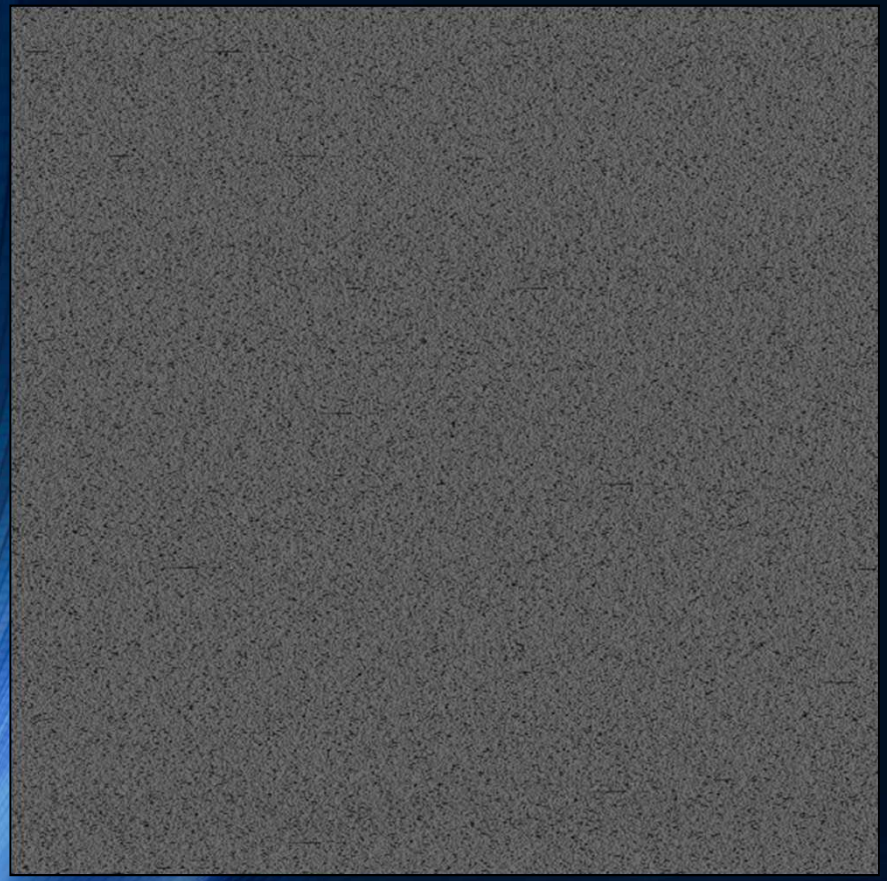
- Encrypted (random) files have the most uncertainty
- A file with a single value has the least, H = 0 (lg 1 = 0)

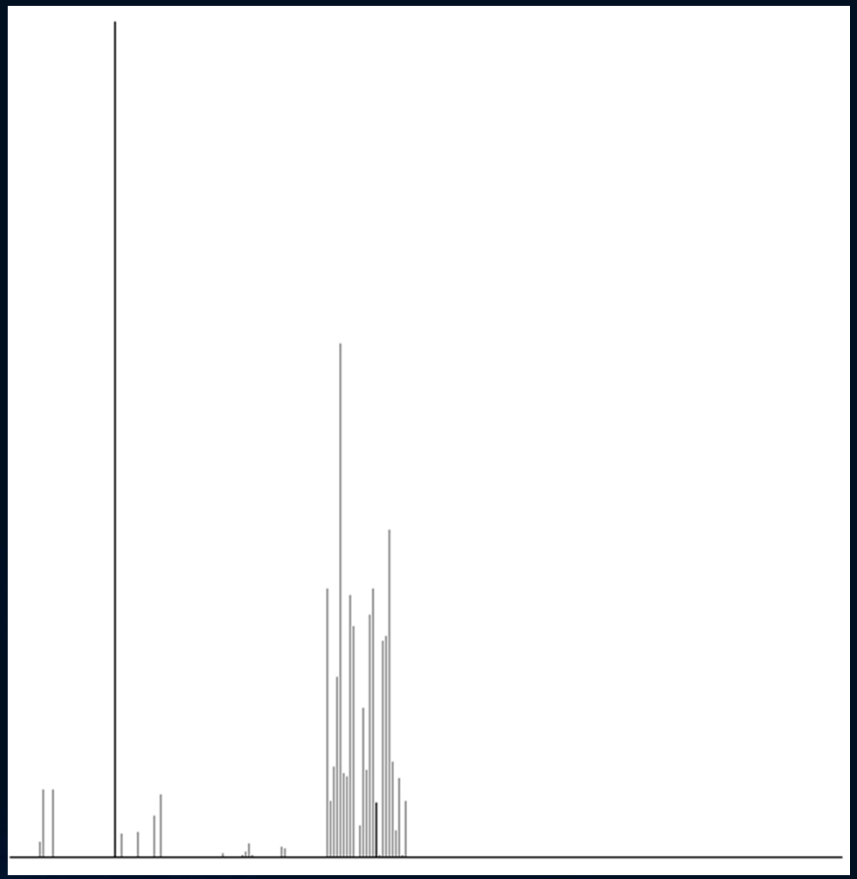
Entropy

- Bottom Line: Higher entropy, higher uncertainty – i.e. randomness
  - Encrypted: H = 7.999+
  - Compressed: H = 7.6+
  - Text: H = 4.5 +/-
- The entropy measurement is only accurate with sufficient data
  - Can't get entropy of 7.99+ for a 1-byte encrypted file
  - For fairly accurate measurement, need around 4 kilobytes
    - There is research on this, but that's for another day
  - Accuracy increases with increasing data size

File Type Characteristics

- Text File
  - H=4.48469





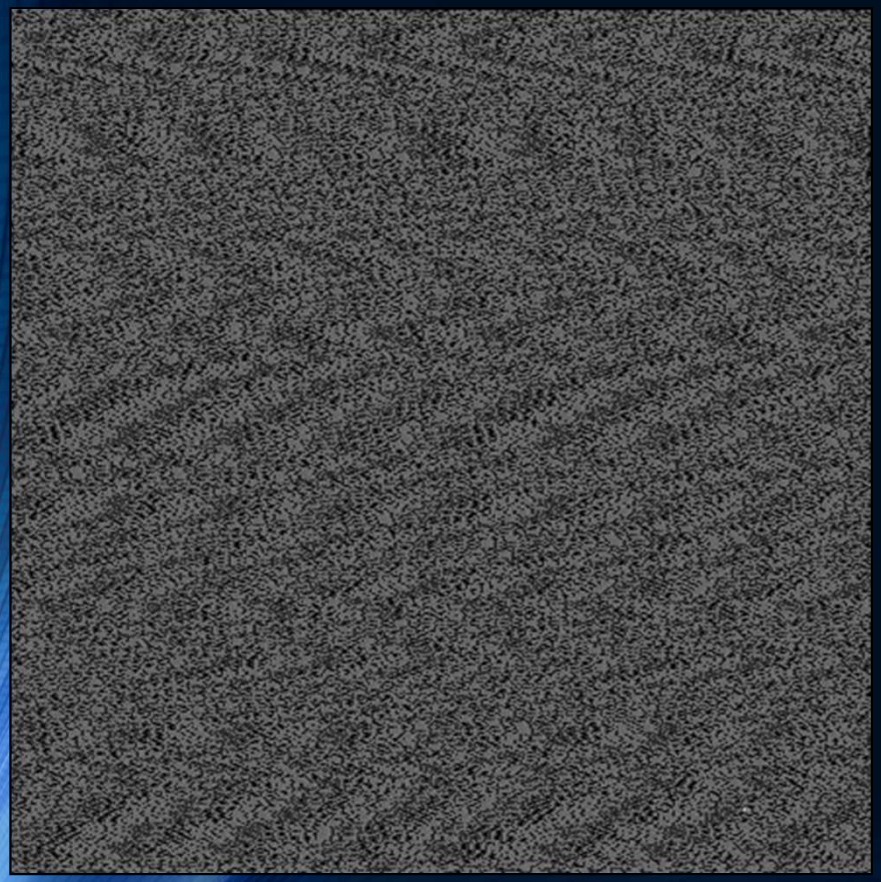
File Type Characteristics

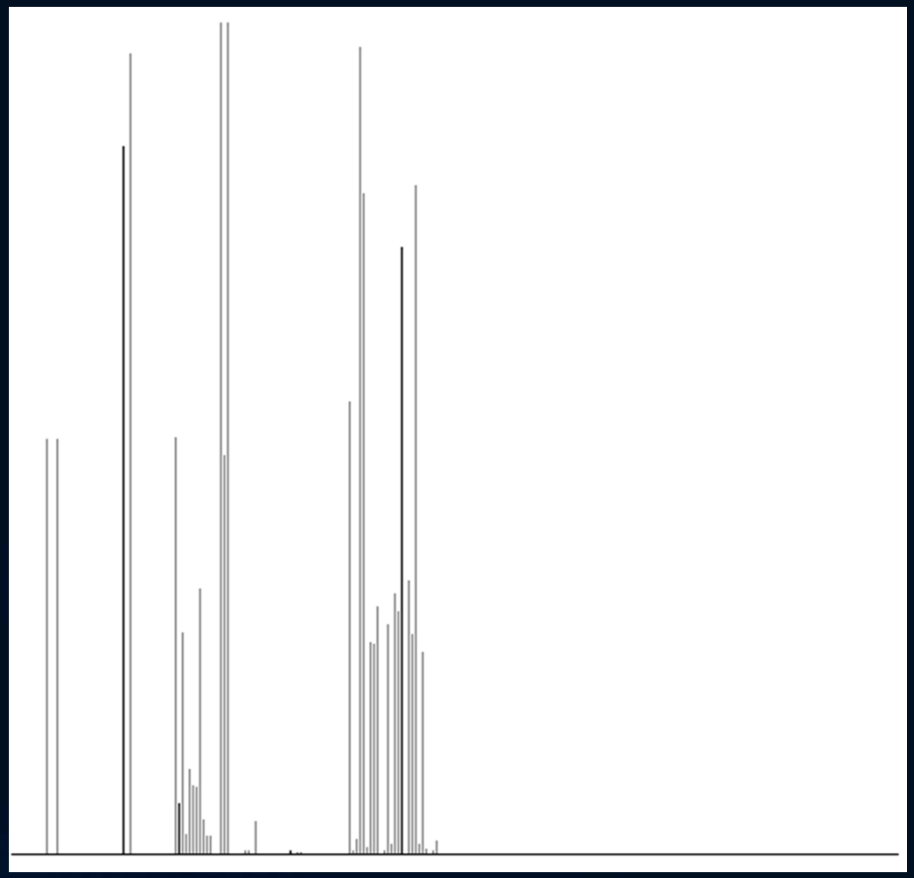
- Partial Texual Histogram of the .txt file

```
a, 097 [61],10631 ( 3.755%)---------+----
b, 098 [62],4117 ( 1.454%)-----
c, 099 [63],4650 ( 1.642%)------
d, 100 [64],3784 ( 1.336%)-----
e, 101 [65],16391 ( 5.789%)---------+----------+-
f, 102 [66],2185 ( 0.772%)--
g, 103 [67],3102 ( 1.096%)----
h, 104 [68],4049 ( 1.430%)-----
i, 105 [69],8865 ( 3.131%)---------+-
j, 106 [6A],211 ( 0.075%)-
```

File Type Characteristics

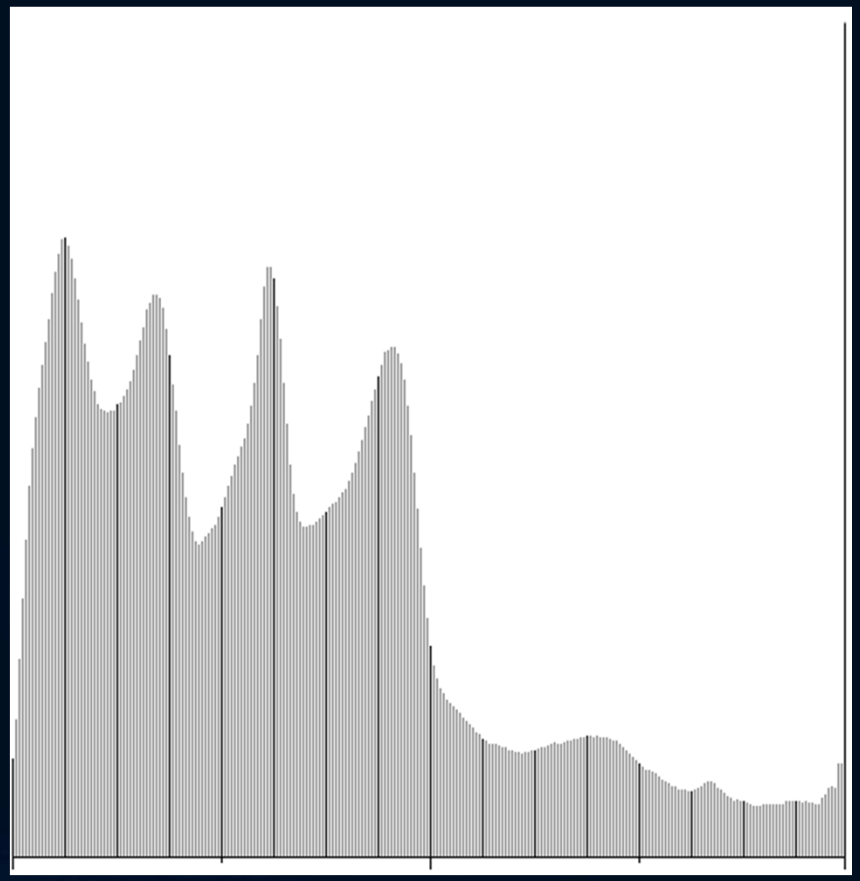
- HTML
  - H=4.70042





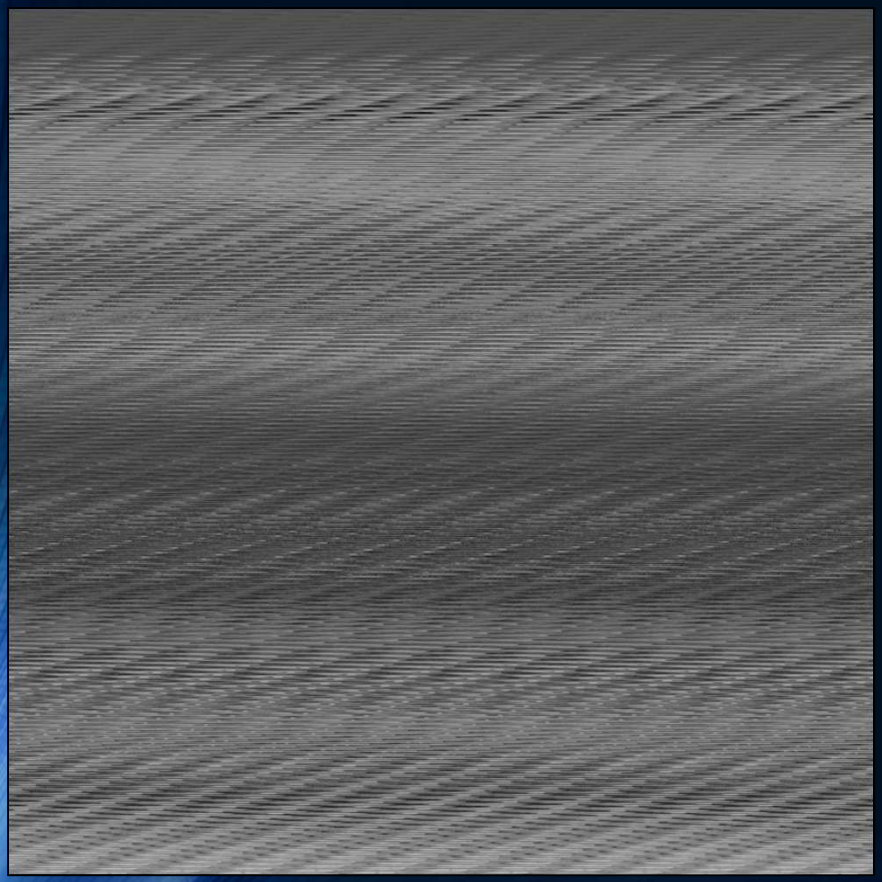
File Type Characteristics

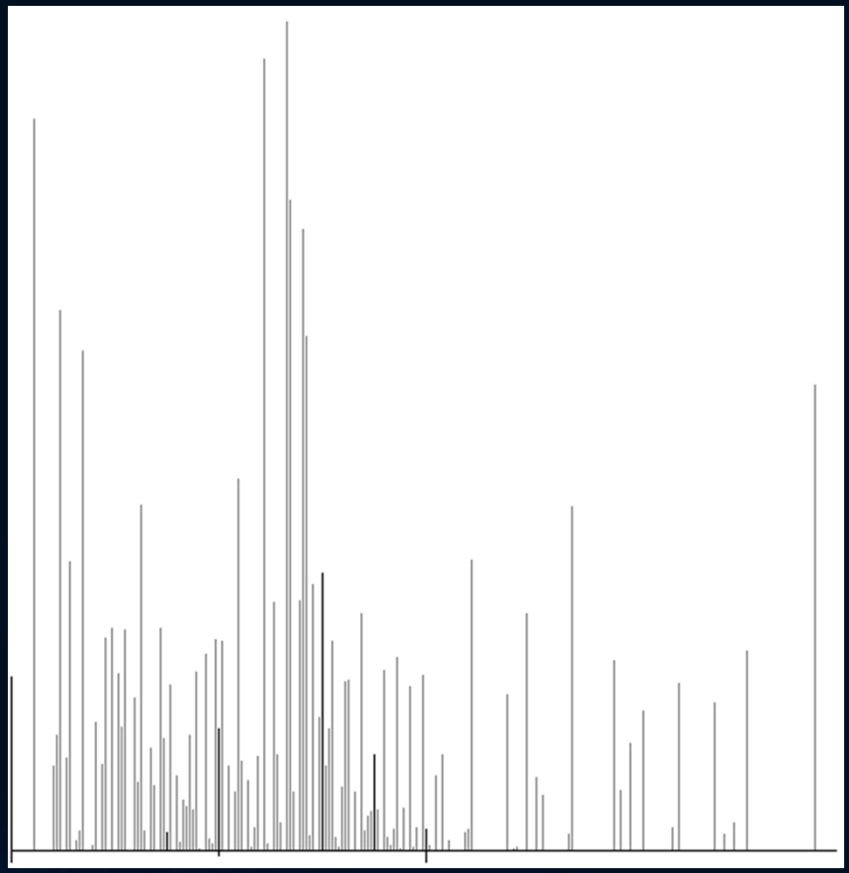
- 24-bit full color RGB bitmap
  - H=7.63054



File Type Characteristics

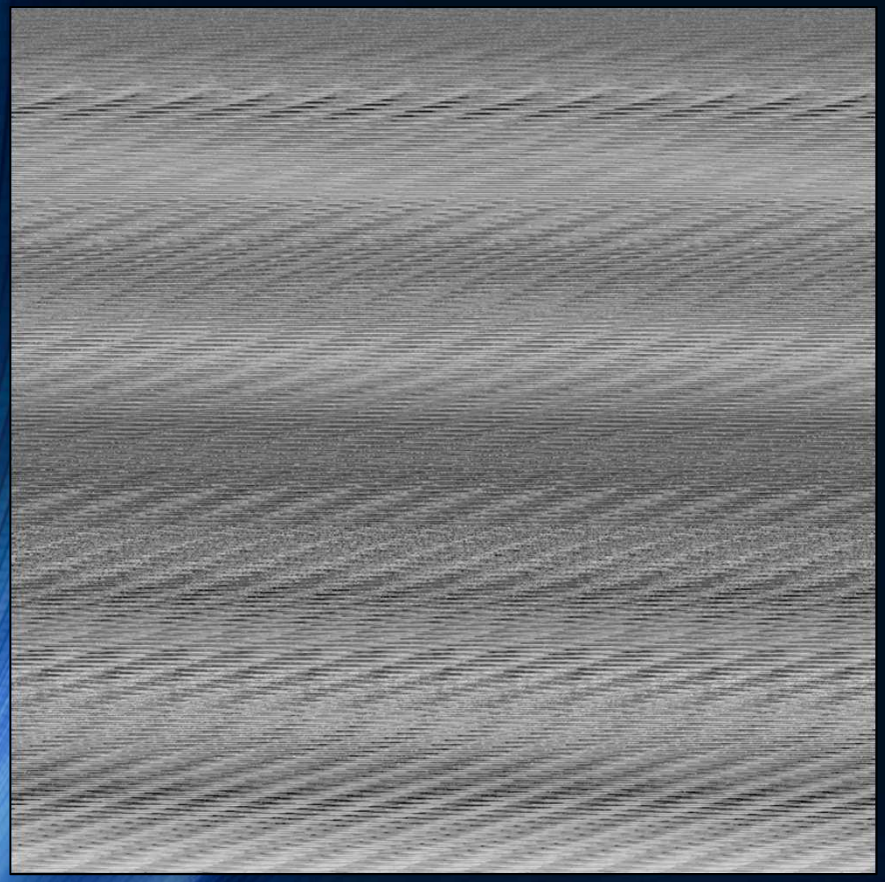
- 8-bit Grayscale Paletted Bitmap
  - H=6.14182

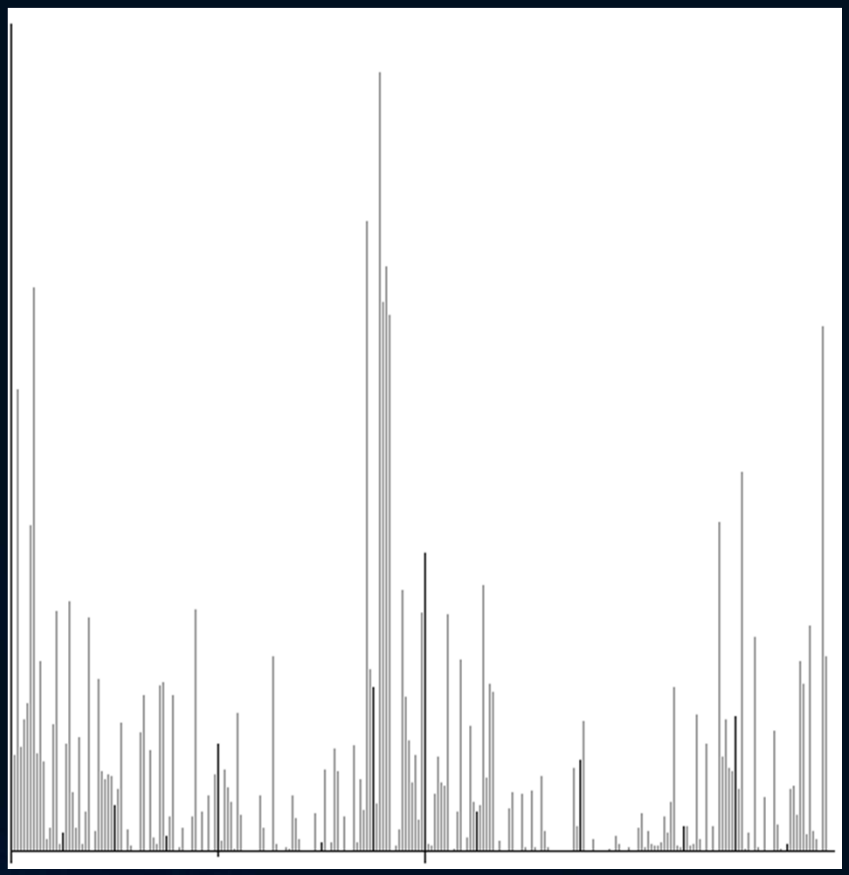




File Type Characteristics

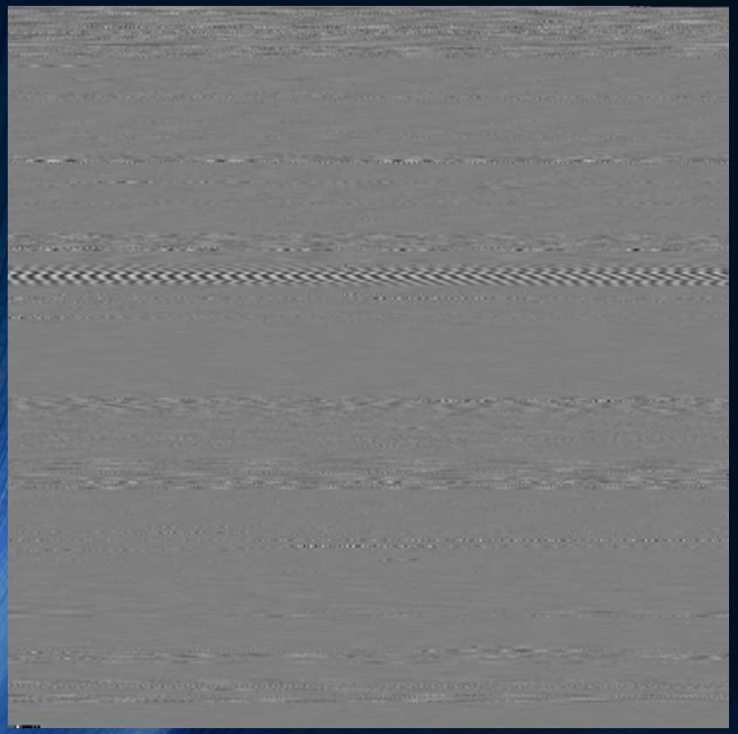
- 8-bit Color Paletted bitmap
  - H=6.68248

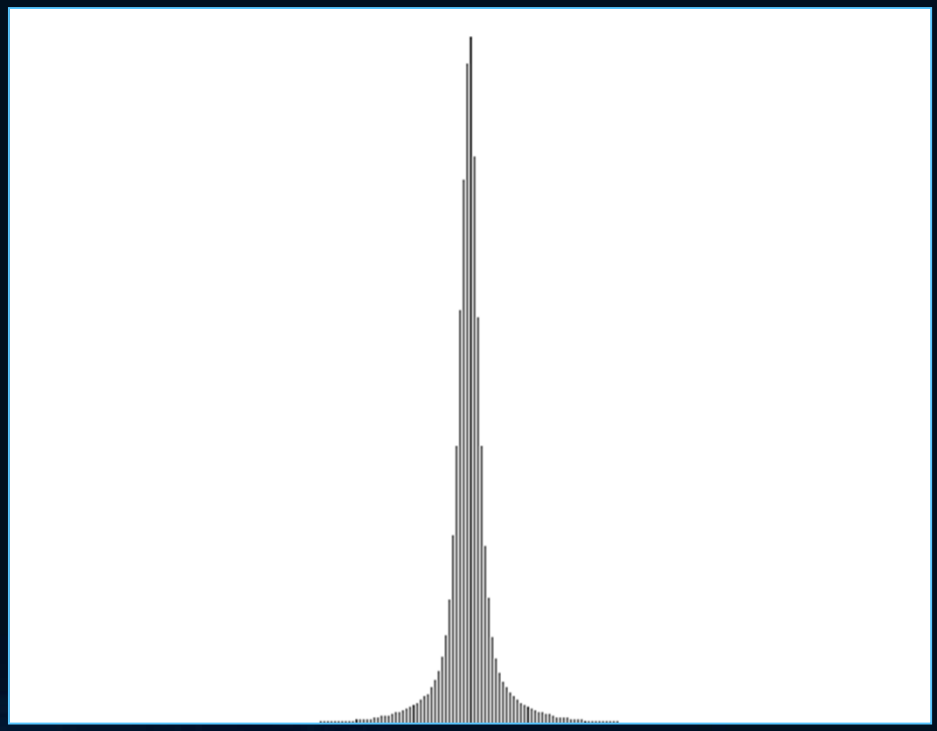




File Type Characteristics

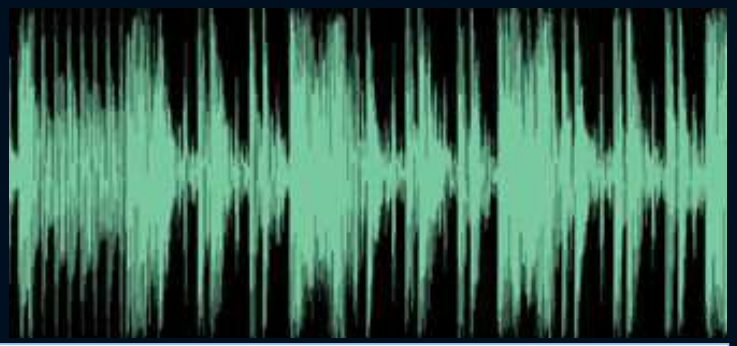
- 8-Bit Wave (Speech)

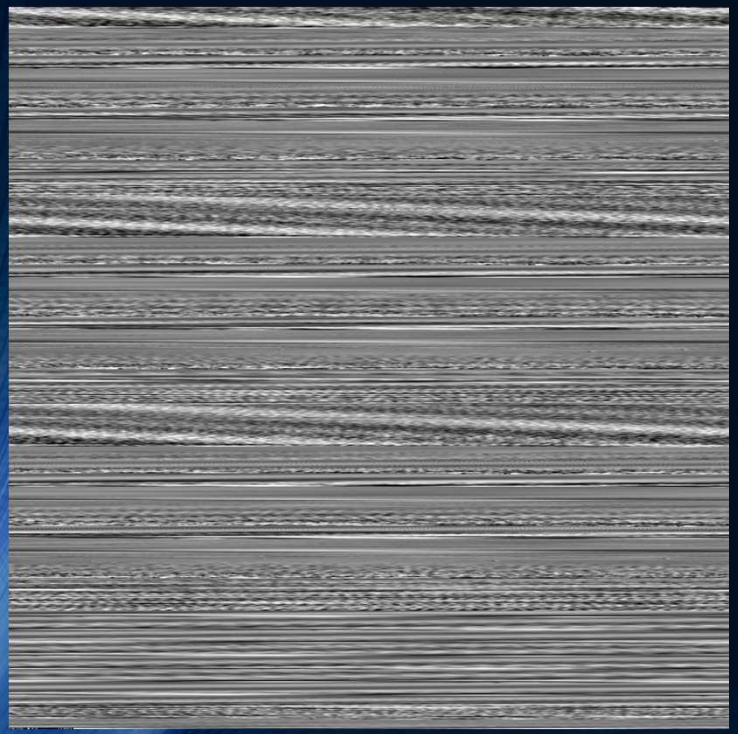


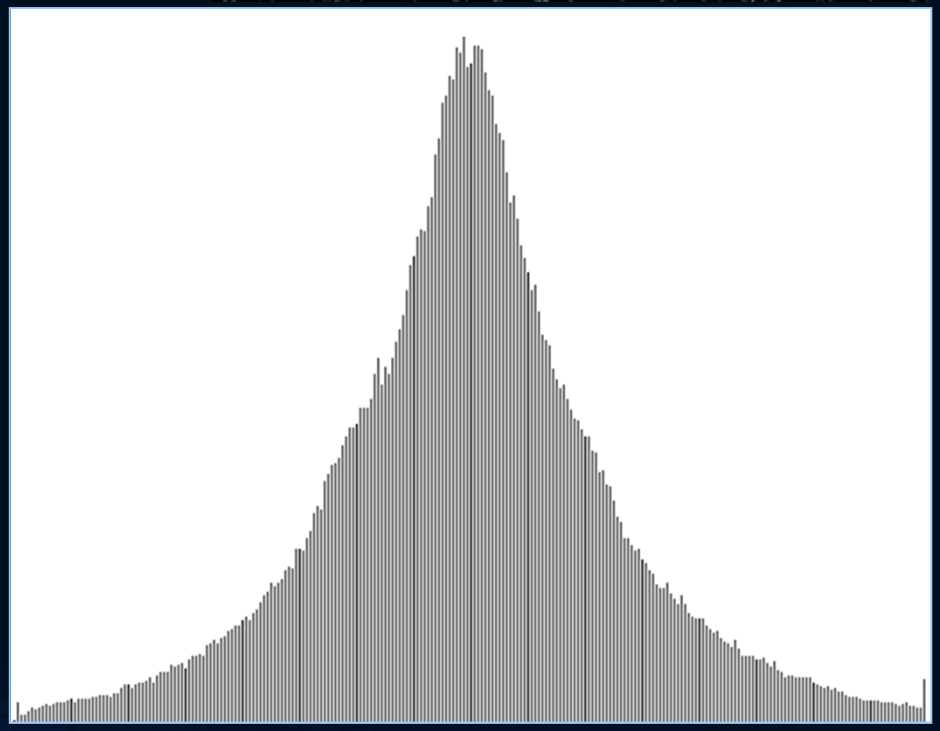


File Type Characteristics

- 8-Bit Wave (Music)

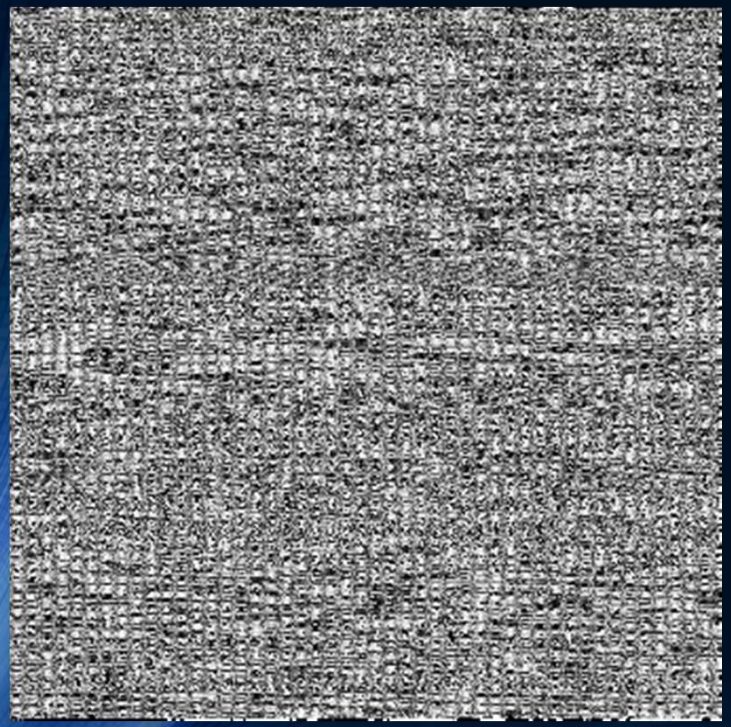


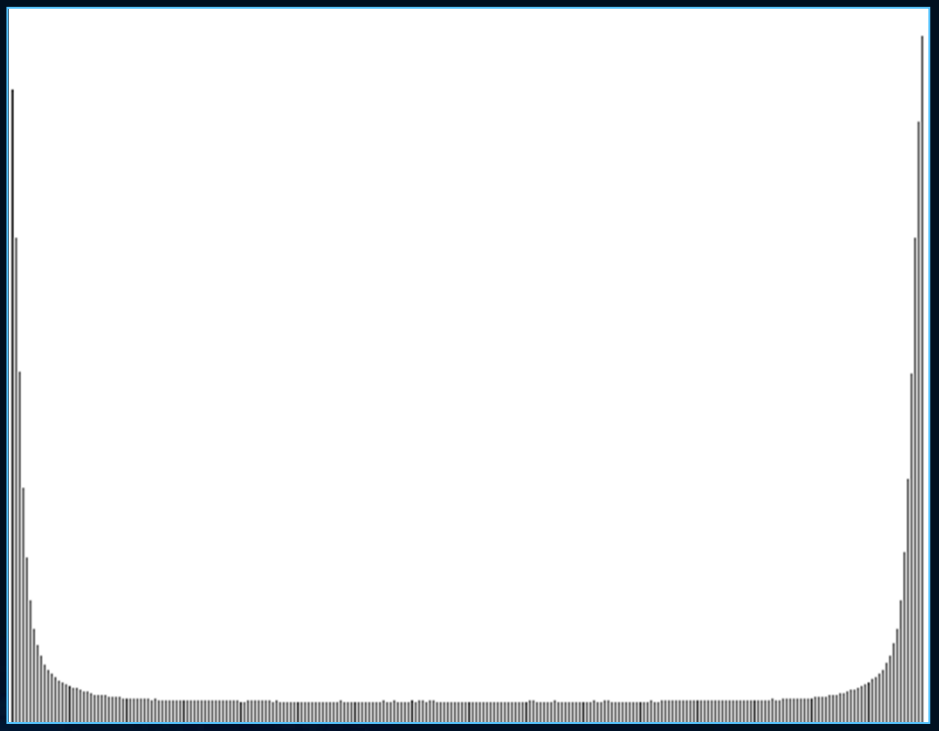




File Type Characteristics

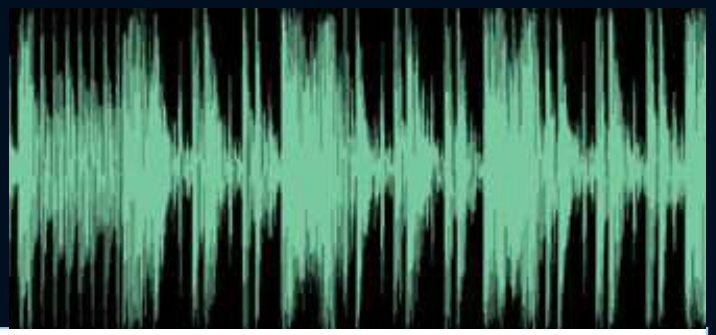
- 16-Bit Wave (Speech)

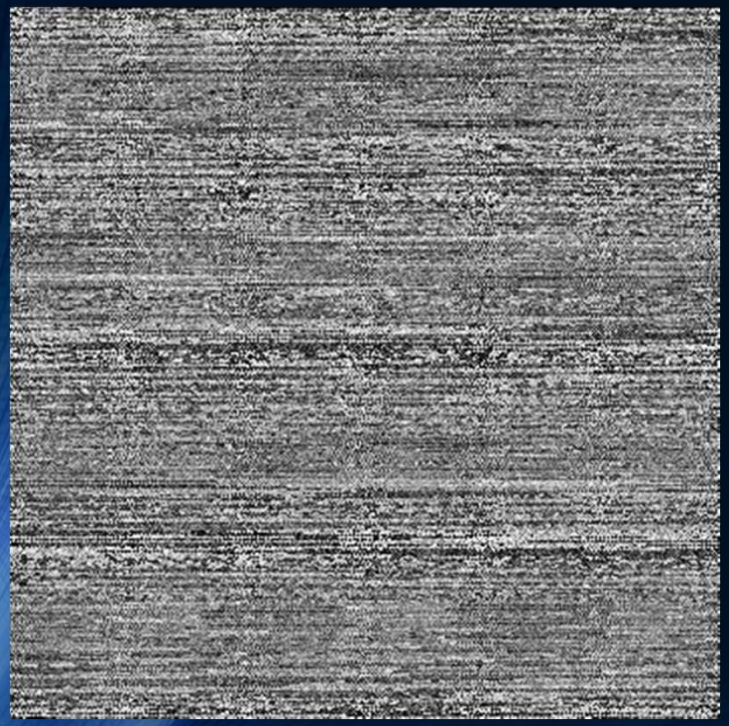


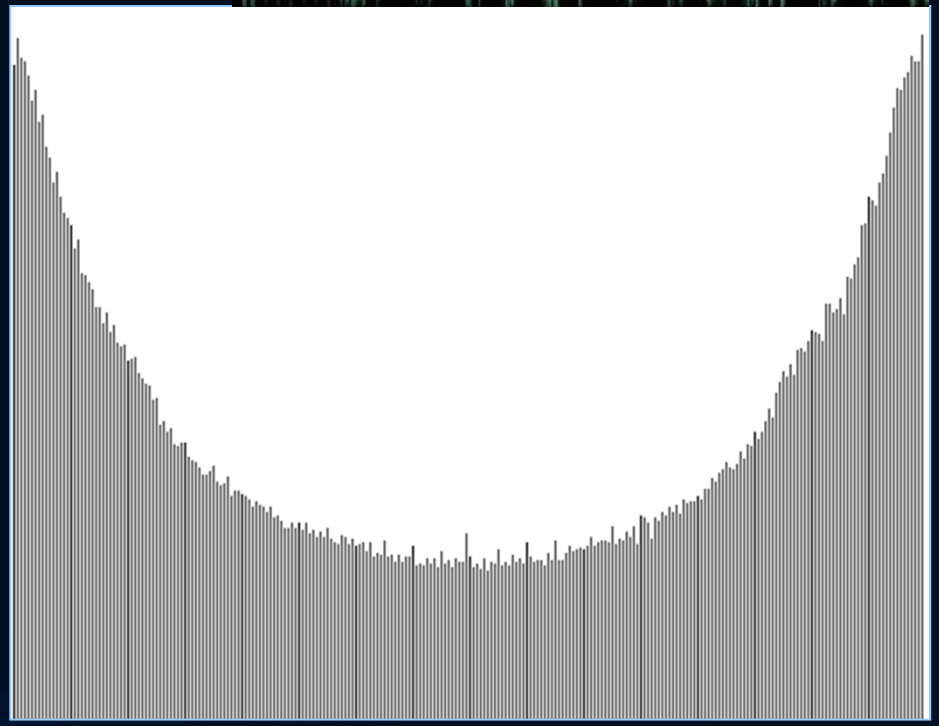


File Type Characteristics

- 16-Bit Wave (Music)

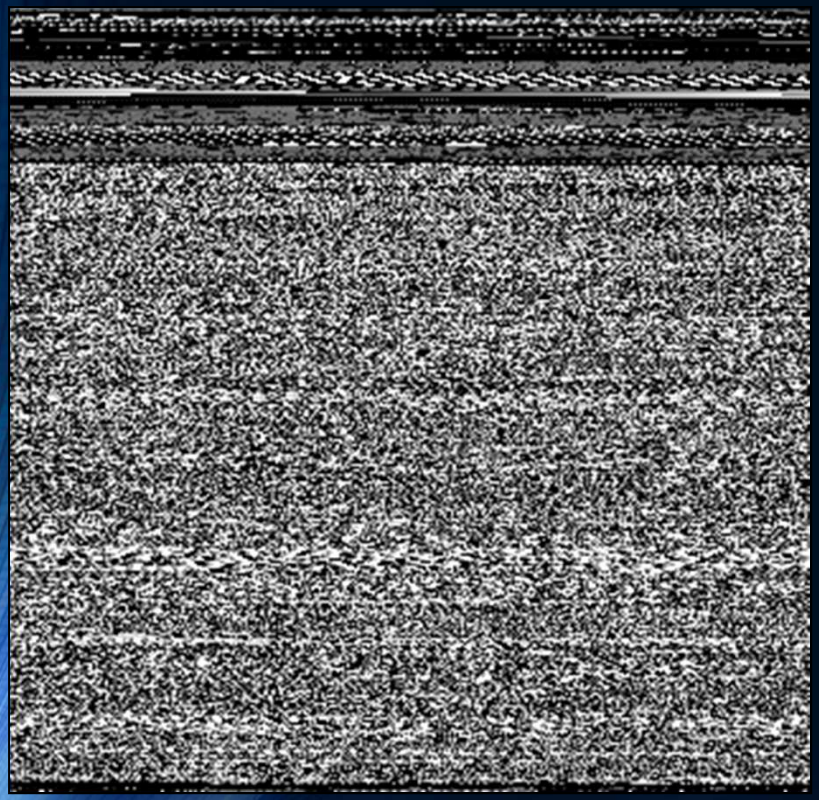


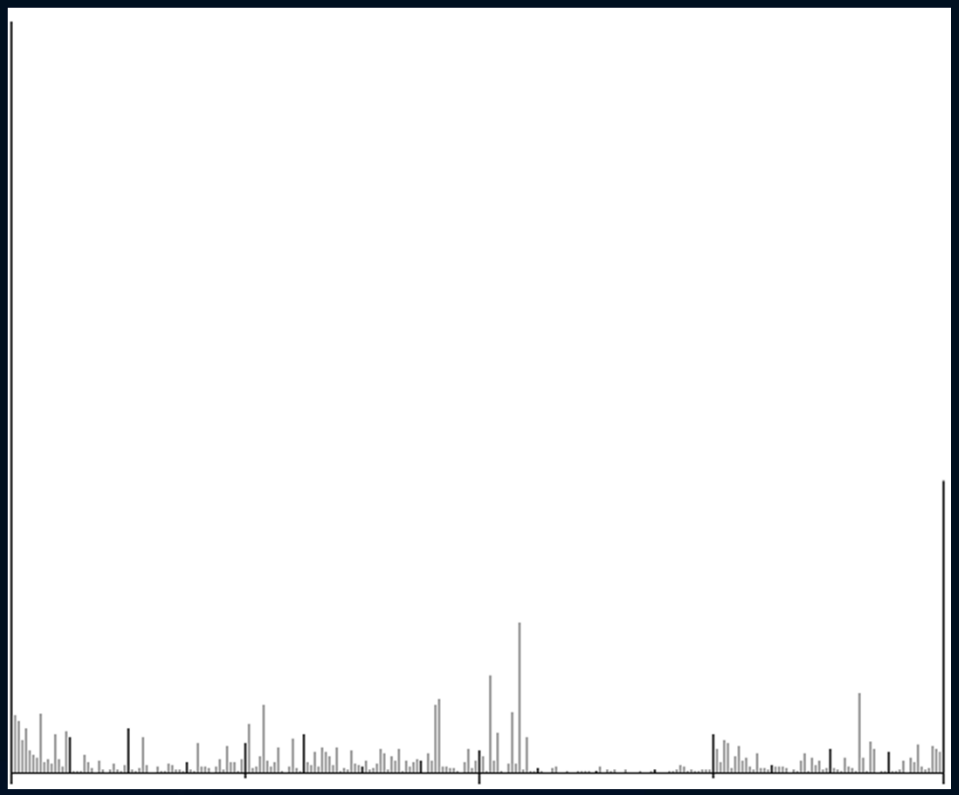




File Type Characteristics

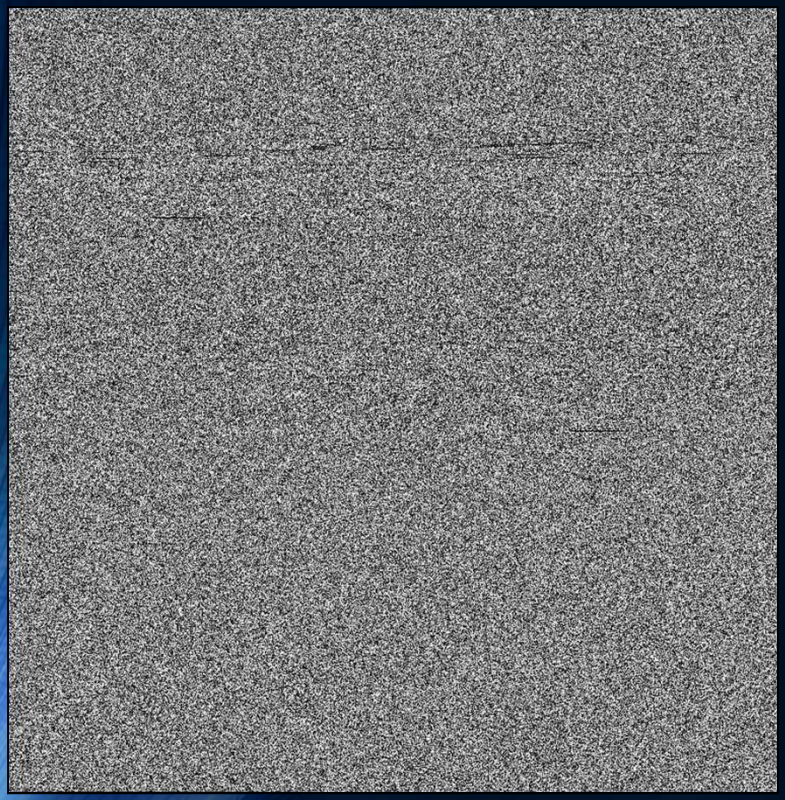
- Portable Executable (PE)
  - H=6.58289

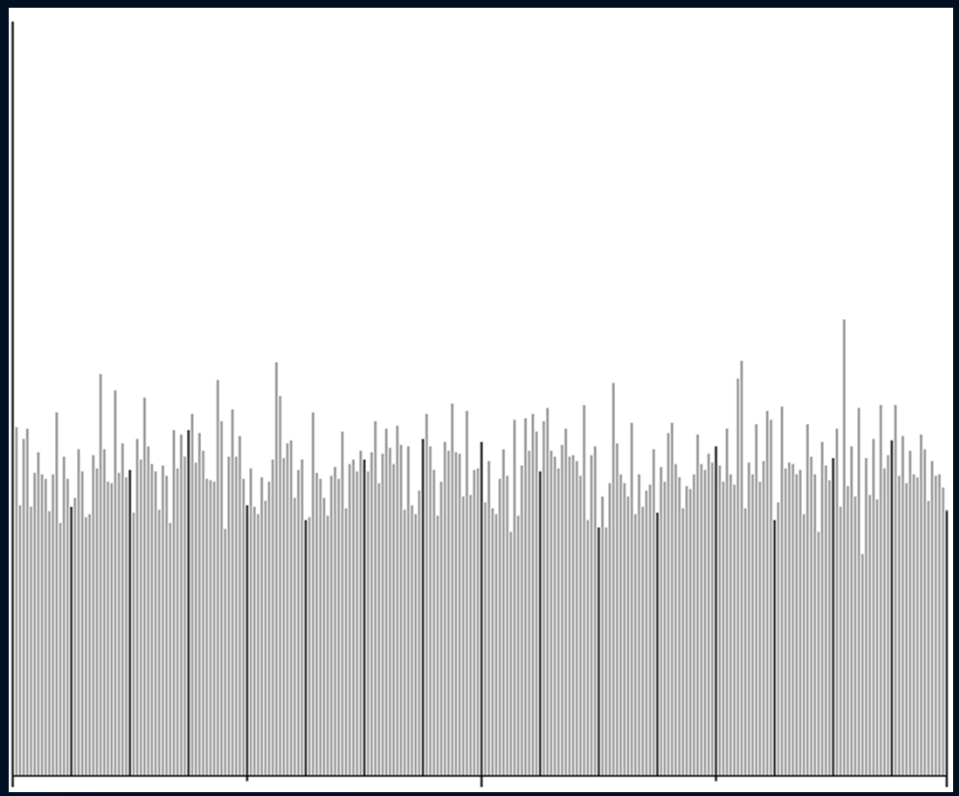




File Type Characteristics

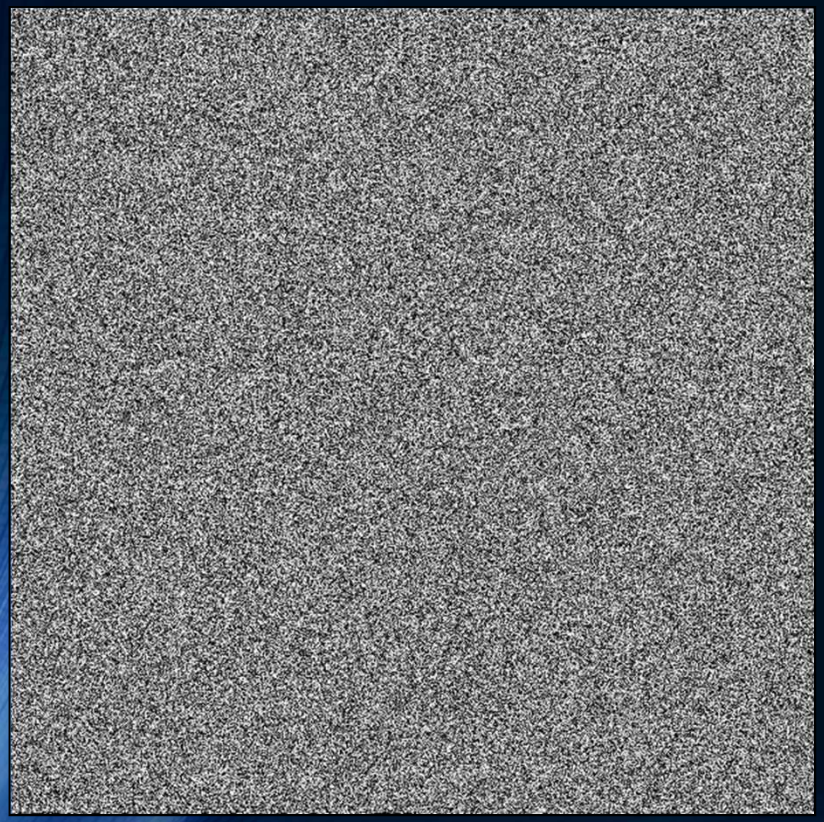
- Jpeg
  - H=7.98698

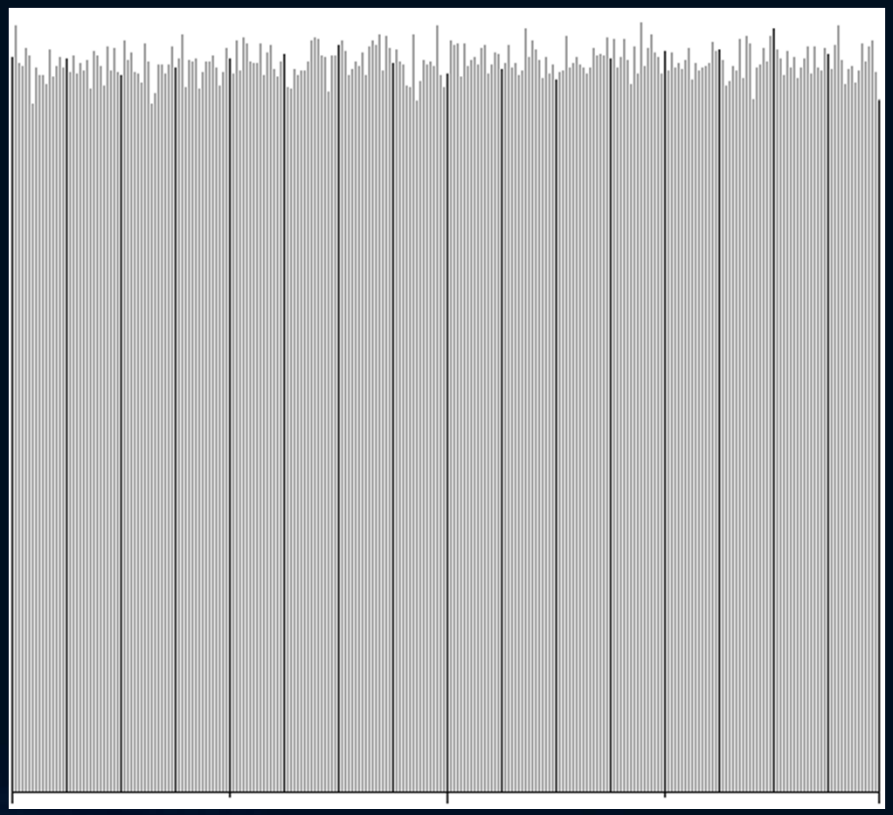




File Type Characteristics

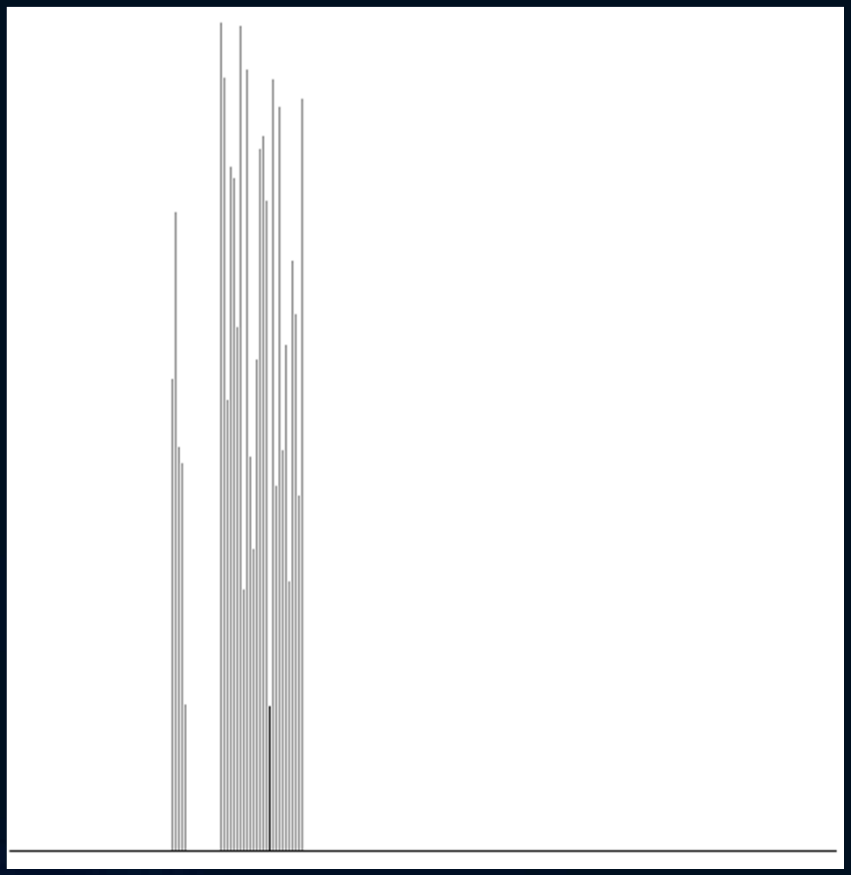
- Encrypted with AES using AxCrypt
  - H=7.99968





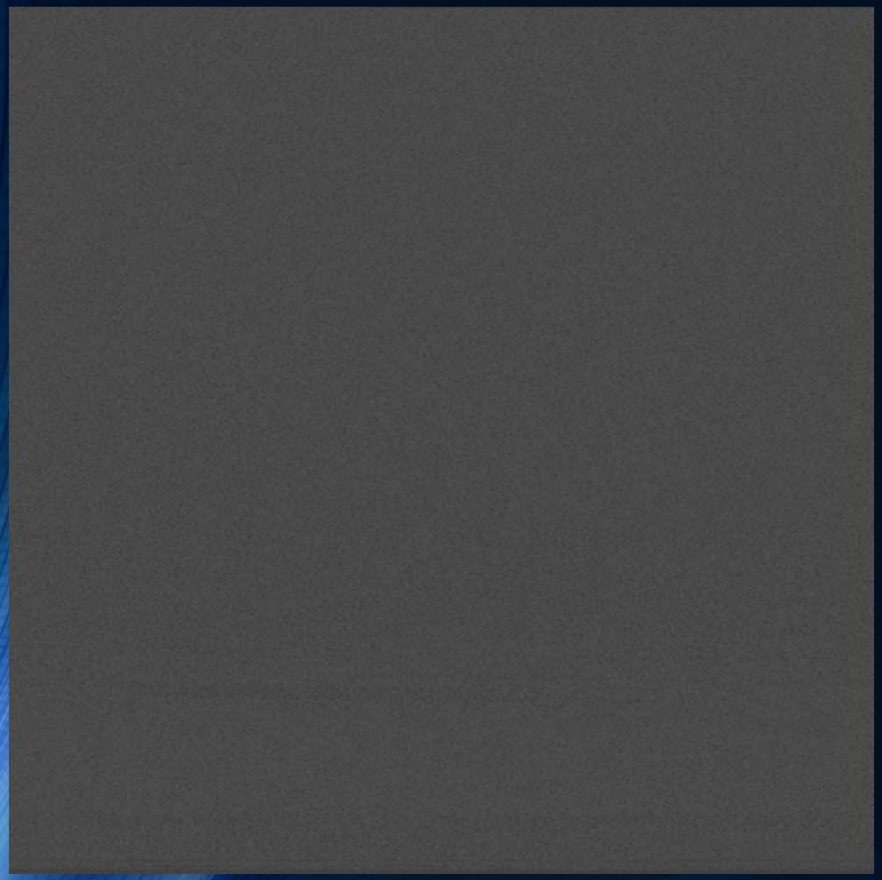
File Type Characteristics

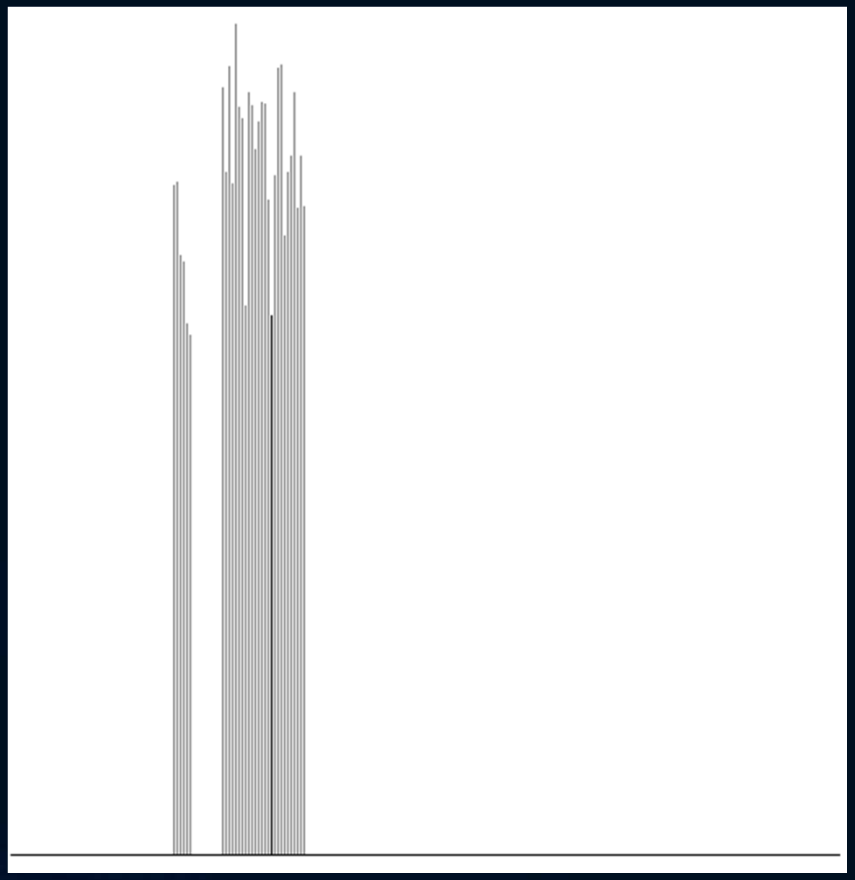
- Base32 Encoded (Text File)
  - H=4.84784 (max possible is 5)



File Type Characteristics

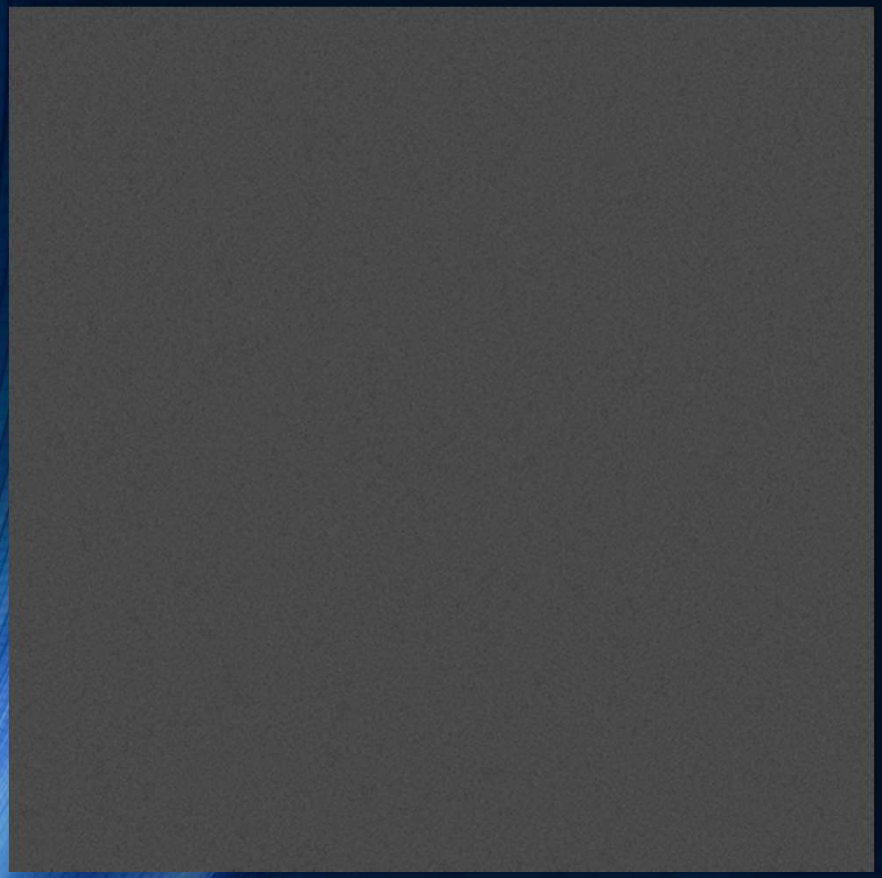
- Base32 Encoded (Compressed File)
  - H=4.98979 (max possible is 5)

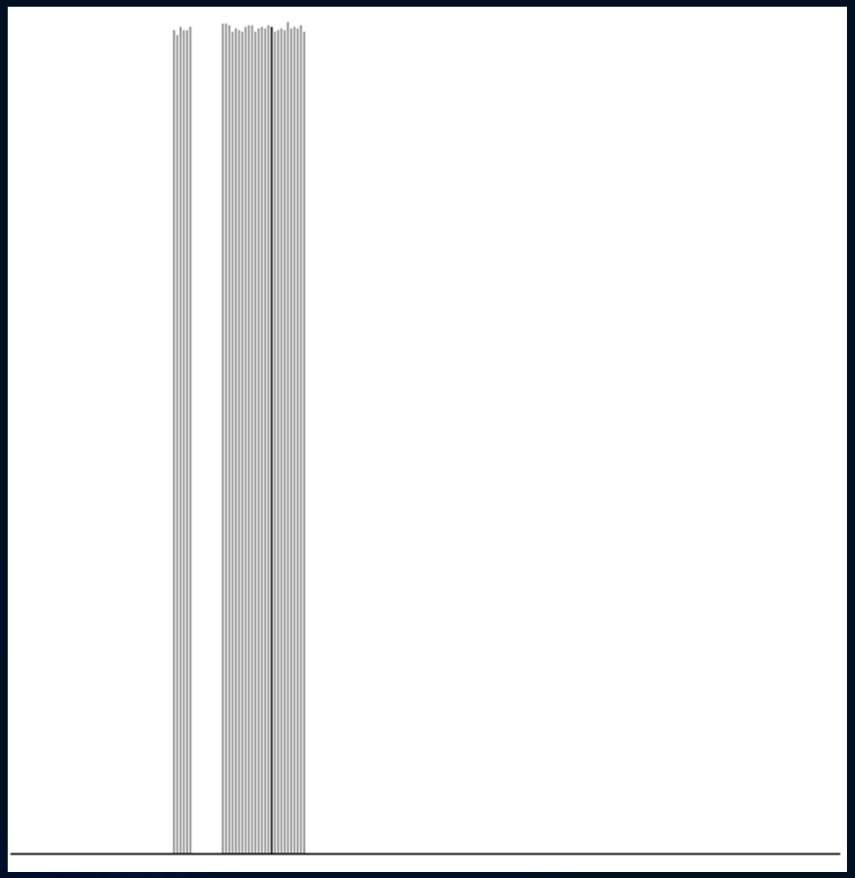




File Type Characteristics

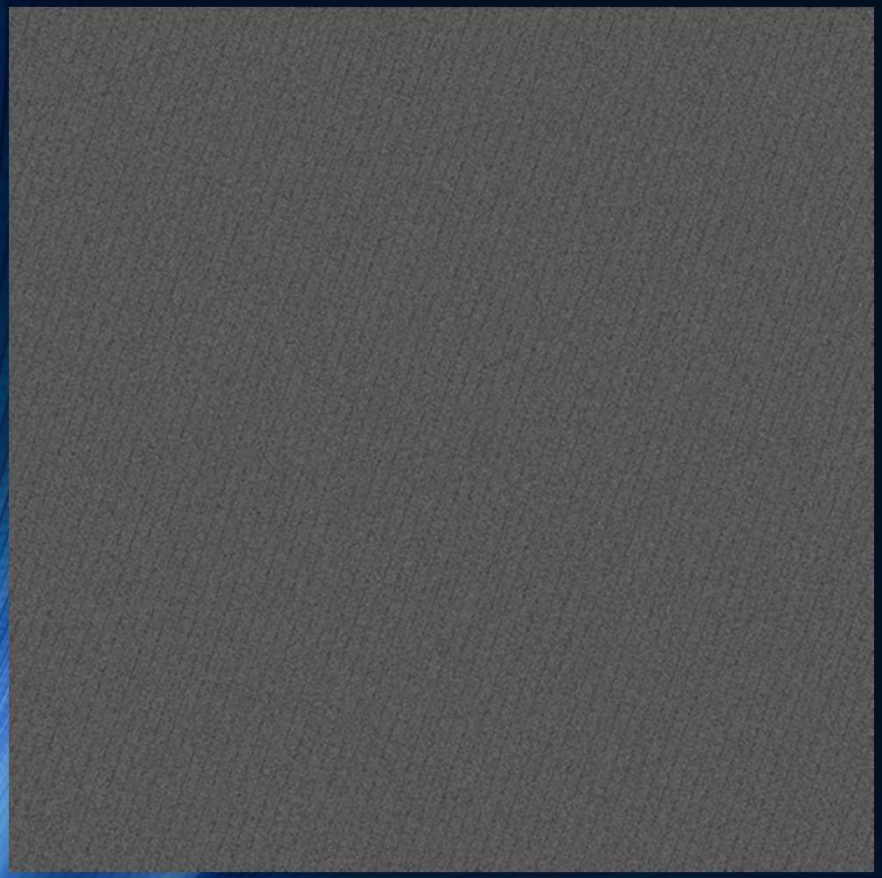
- Base32 Encoded (Encrypted File)
  - H=4.99999 (max possible is 5)

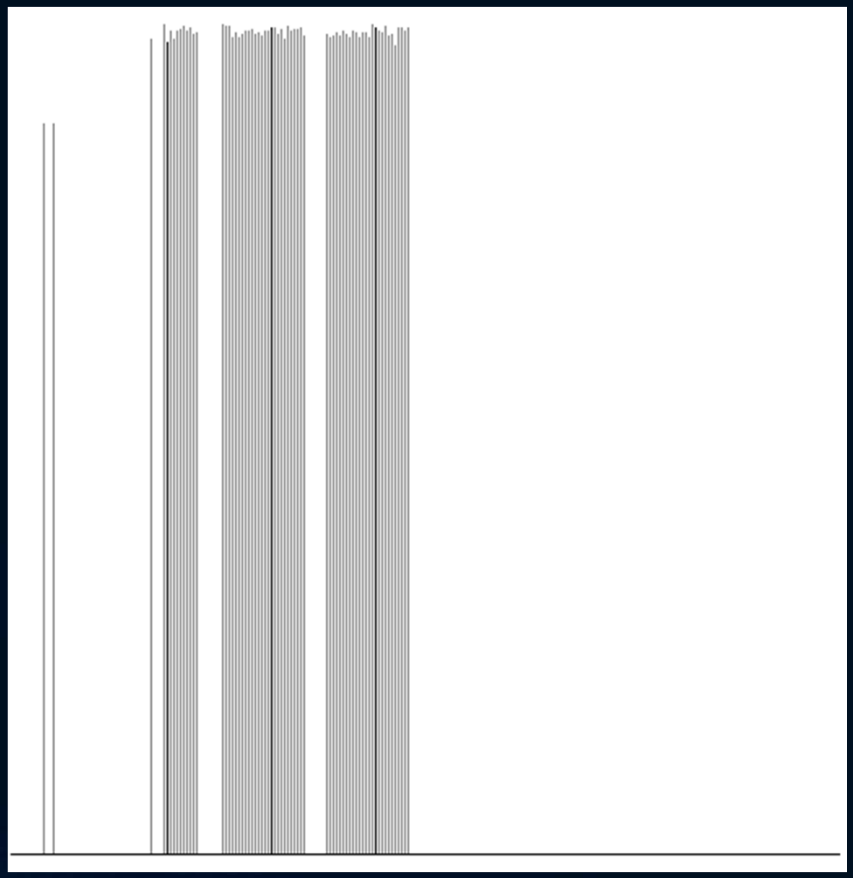




File Type Characteristics

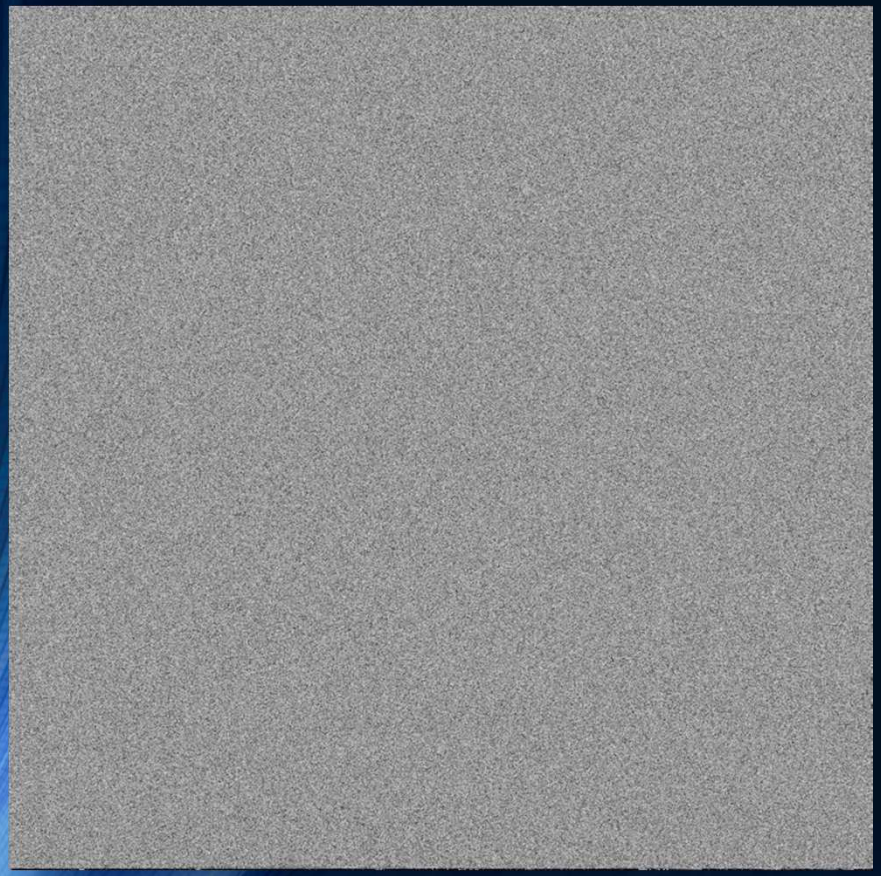
- Base64 Encoded (Encrypted File)
  - H= 6.04411 (max possible is 6.044394119) { CR/LF added }

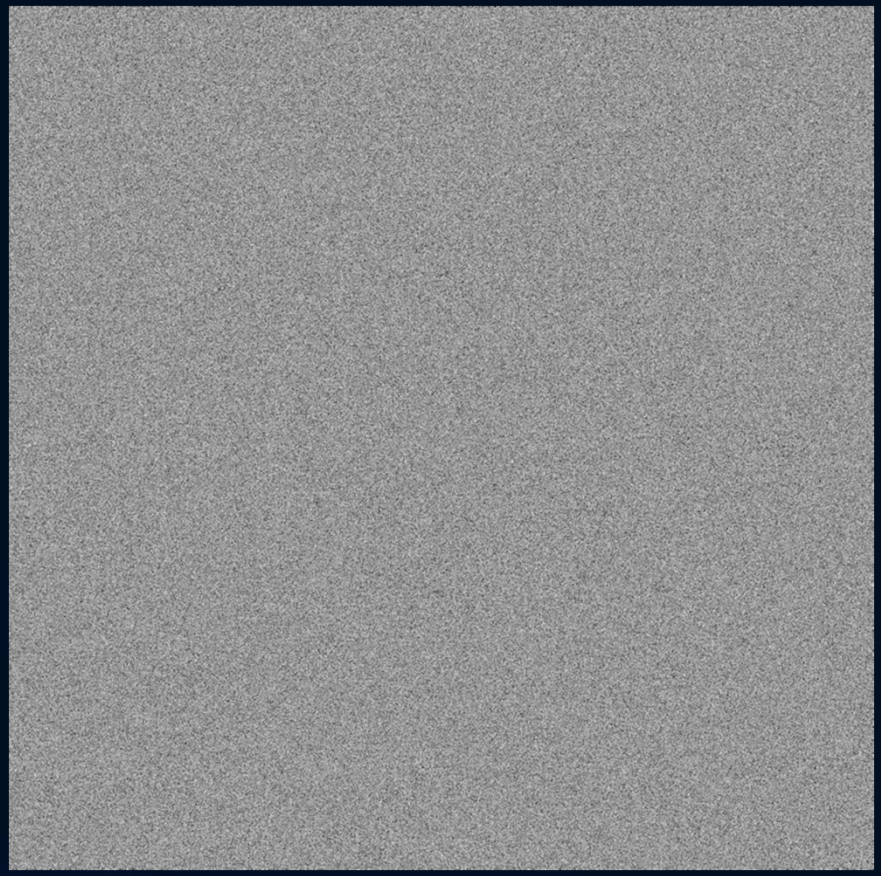




Steganalysis

- Compressed or Encrypted?
  - Can't tell from visualization

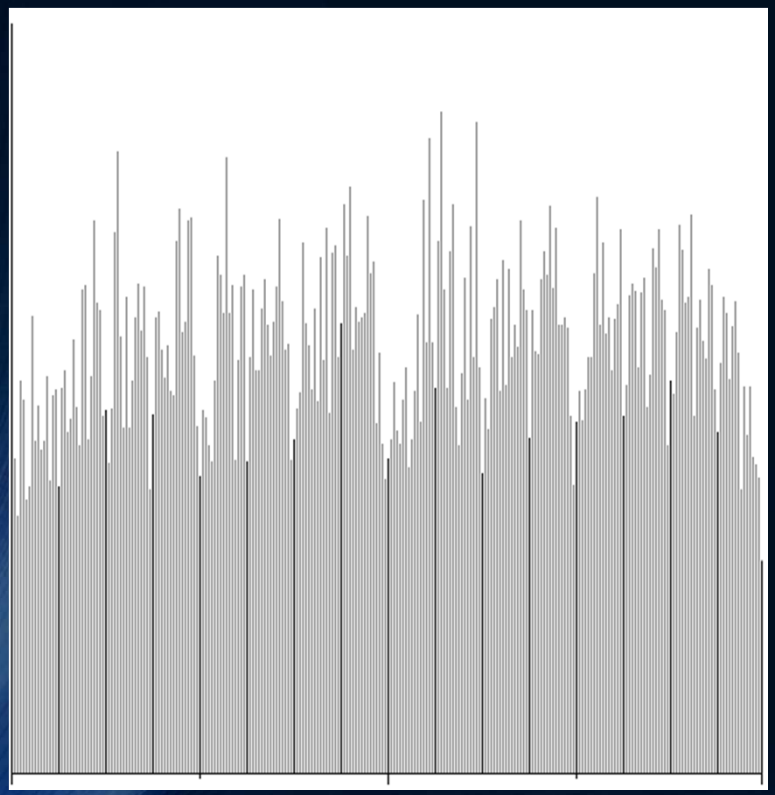




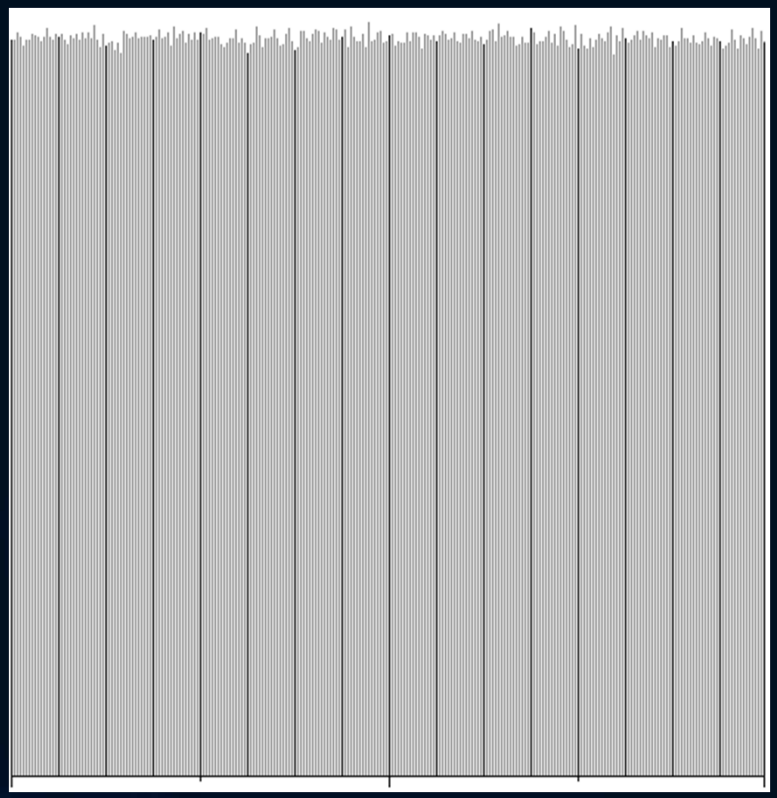
Steganalysis

- Compressed or Encrypted?

H= 7.97085

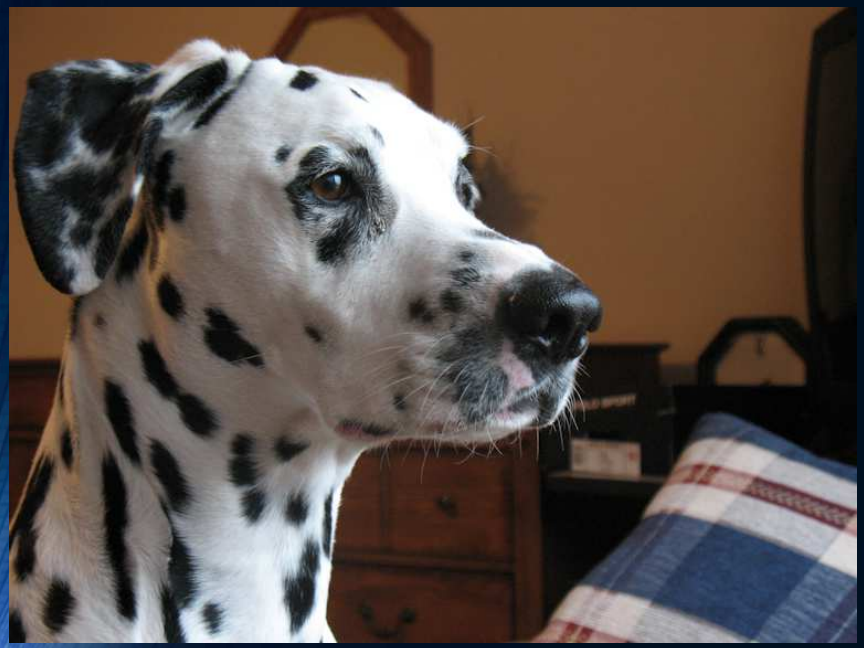


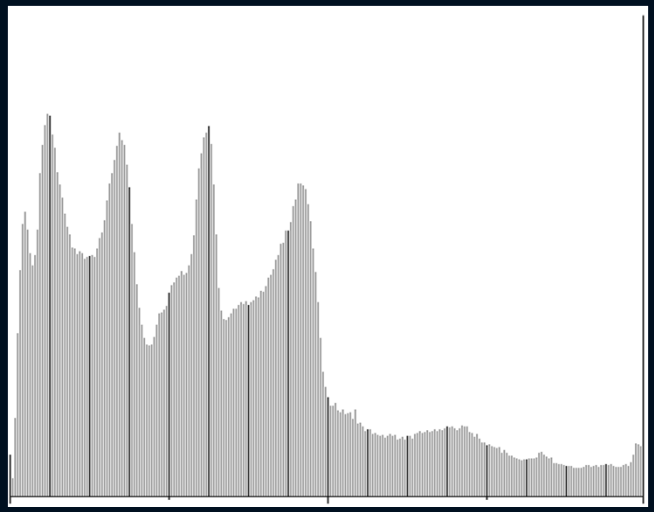
H= 7.99997



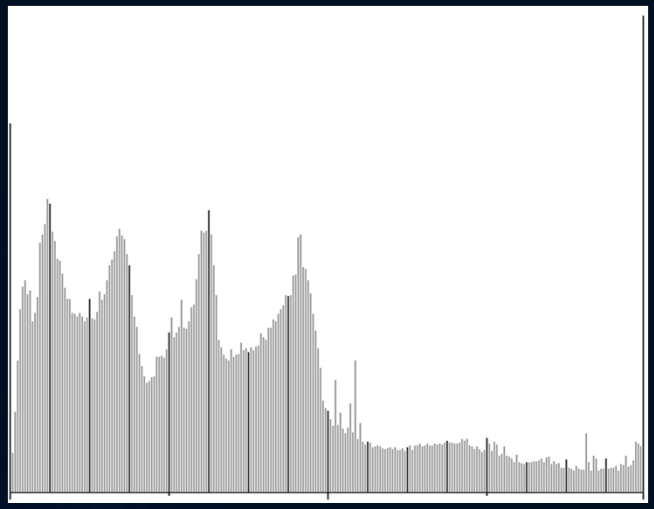
Steganalysis

Before appending data:
H= 7.61037





After appending data:
H= 7.63532



Steganalysis - Visualization

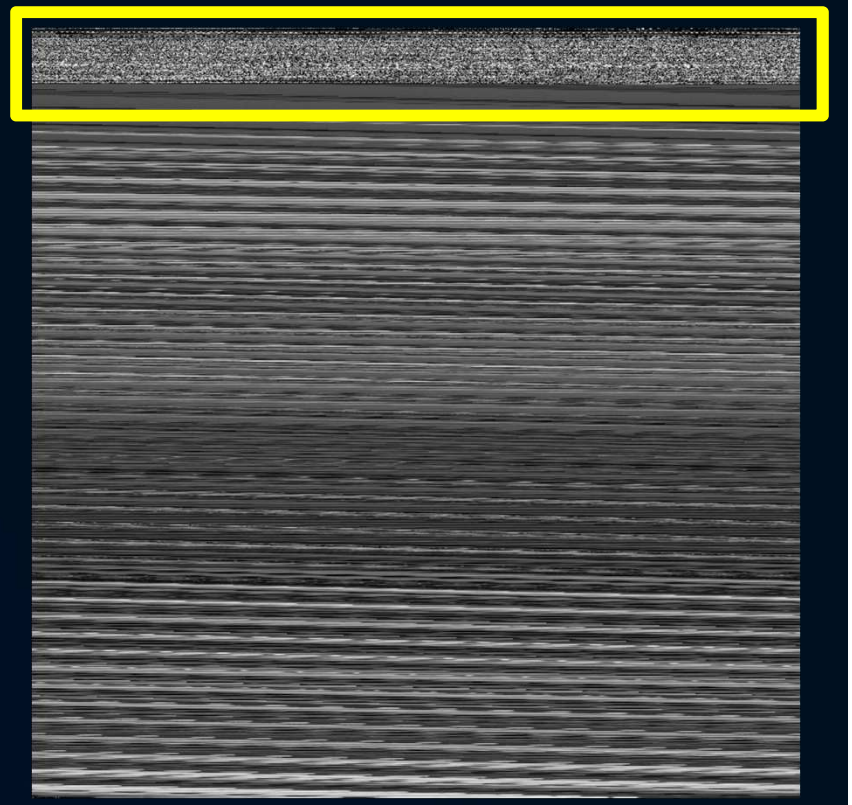
- Histogram & entropy not very effective in that case

Image of the file reveals appended data at end

NOTE: bitmaps start from bottom up

Entropy of original image already fairly high

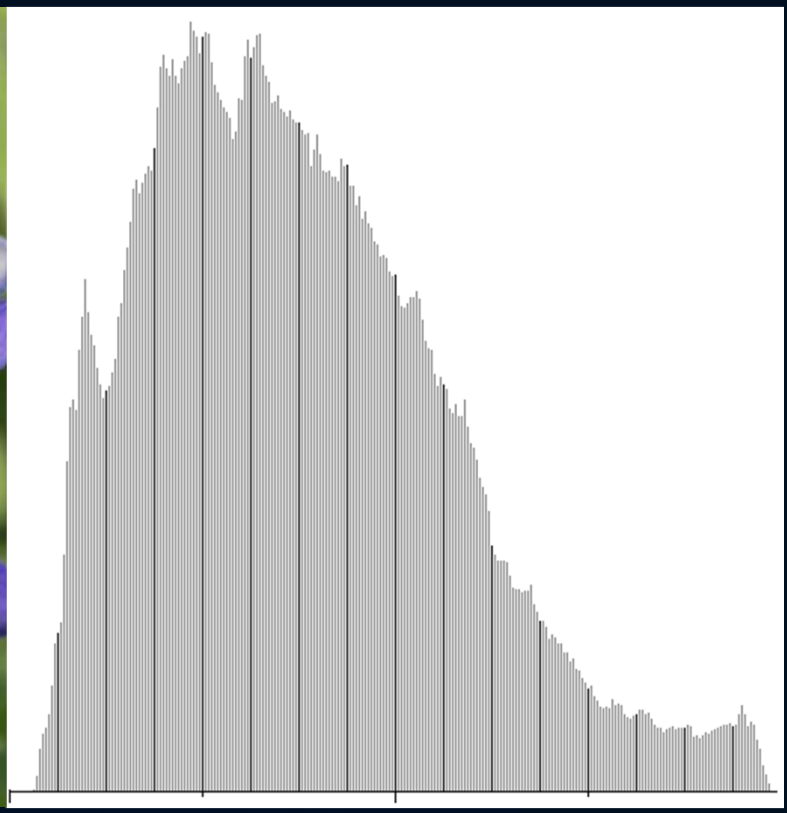
The larger the appended data, the more its entropy characteristics show



Steganalysis - LSB

- Original, zero bits altered

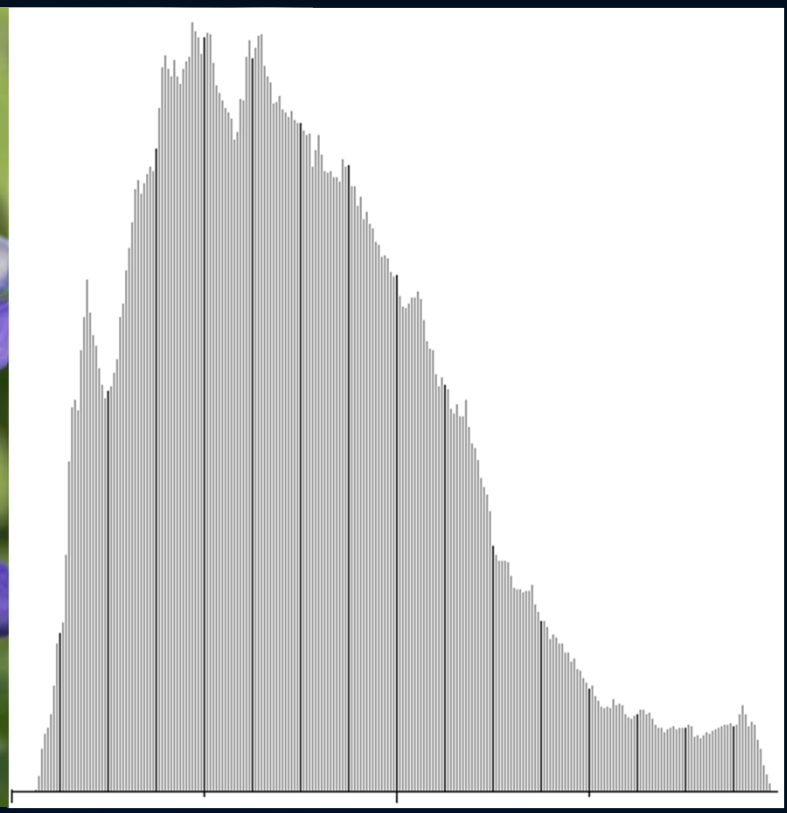
H= 7.55730



Steganalysis - LSB

- 1 bit of randomized data

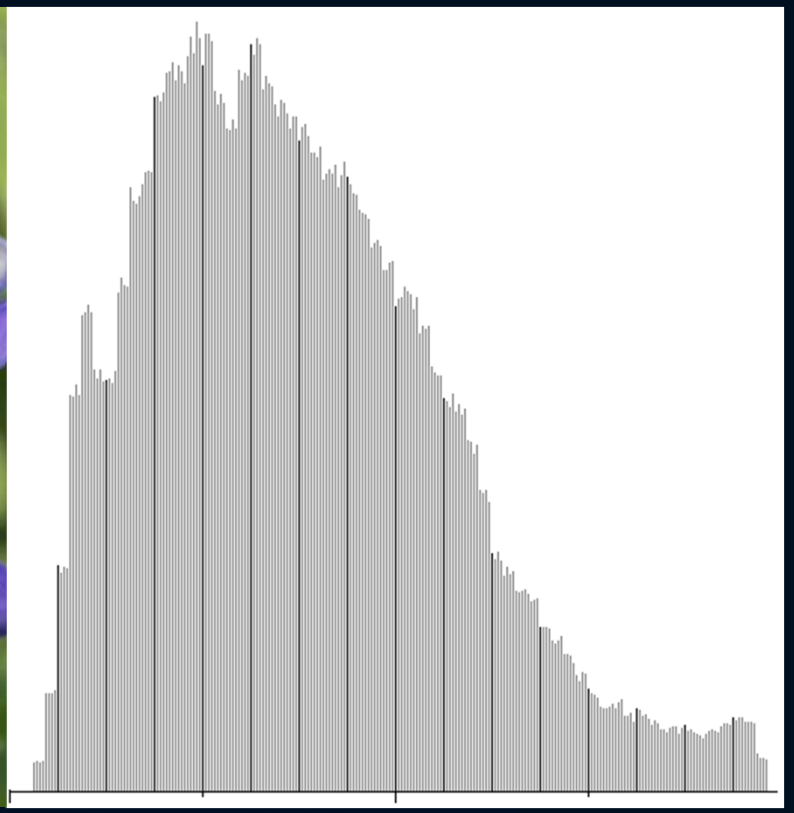
H= 7.55782



Steganalysis - LSB

- 2 bits of randomized data

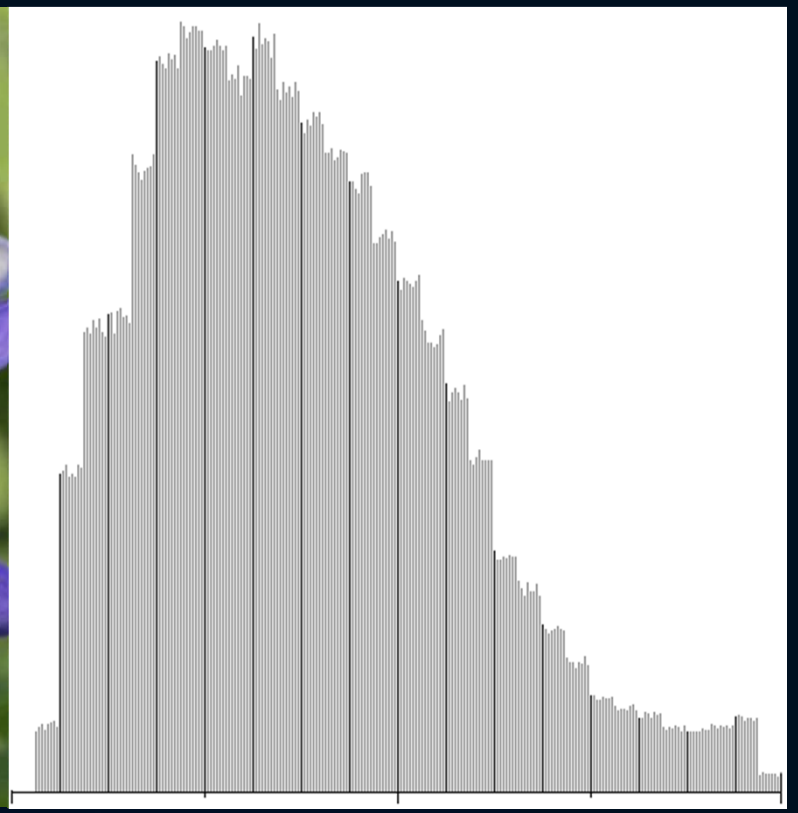
H= 7.55962



Steganalysis - LSB

- 3 bits of randomized data

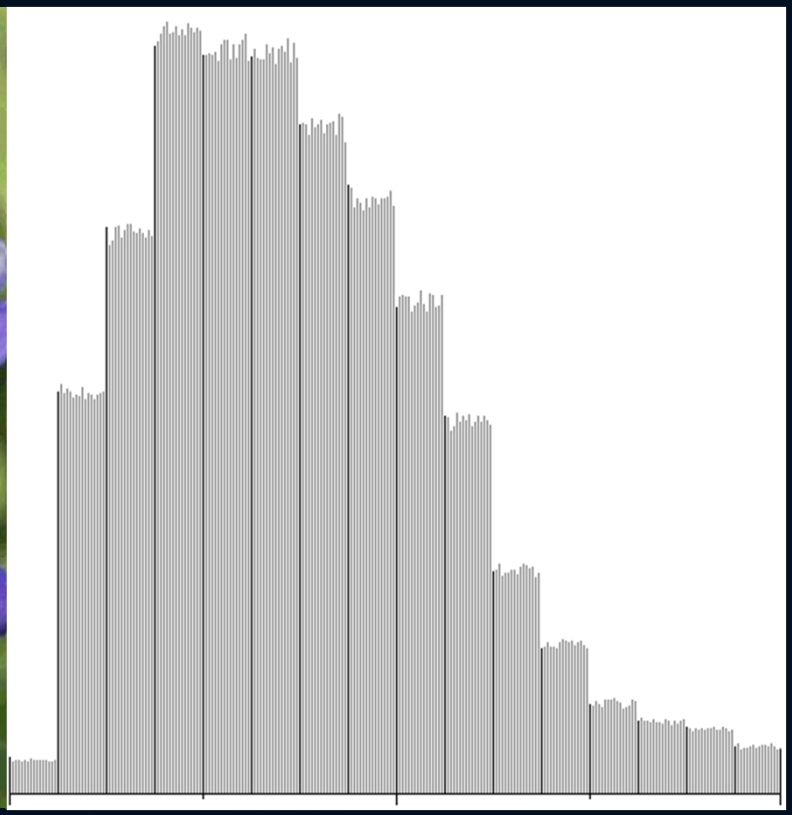
H= 7.56456



Steganalysis - LSB

- 4 bits of randomized data

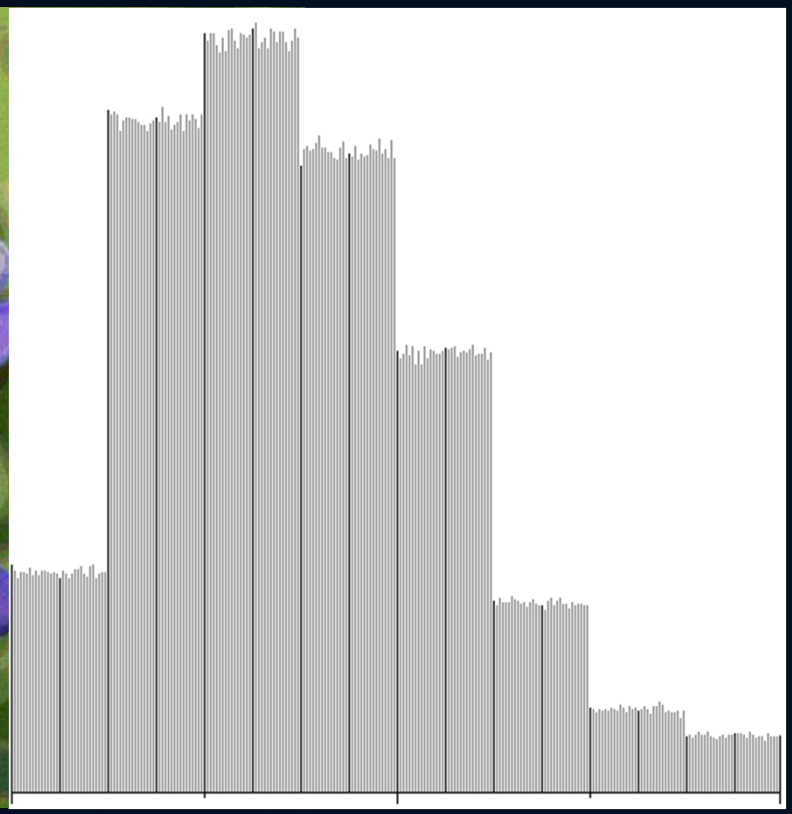
H= 7.57645



Steganalysis - LSB

- 5 bits of randomized data

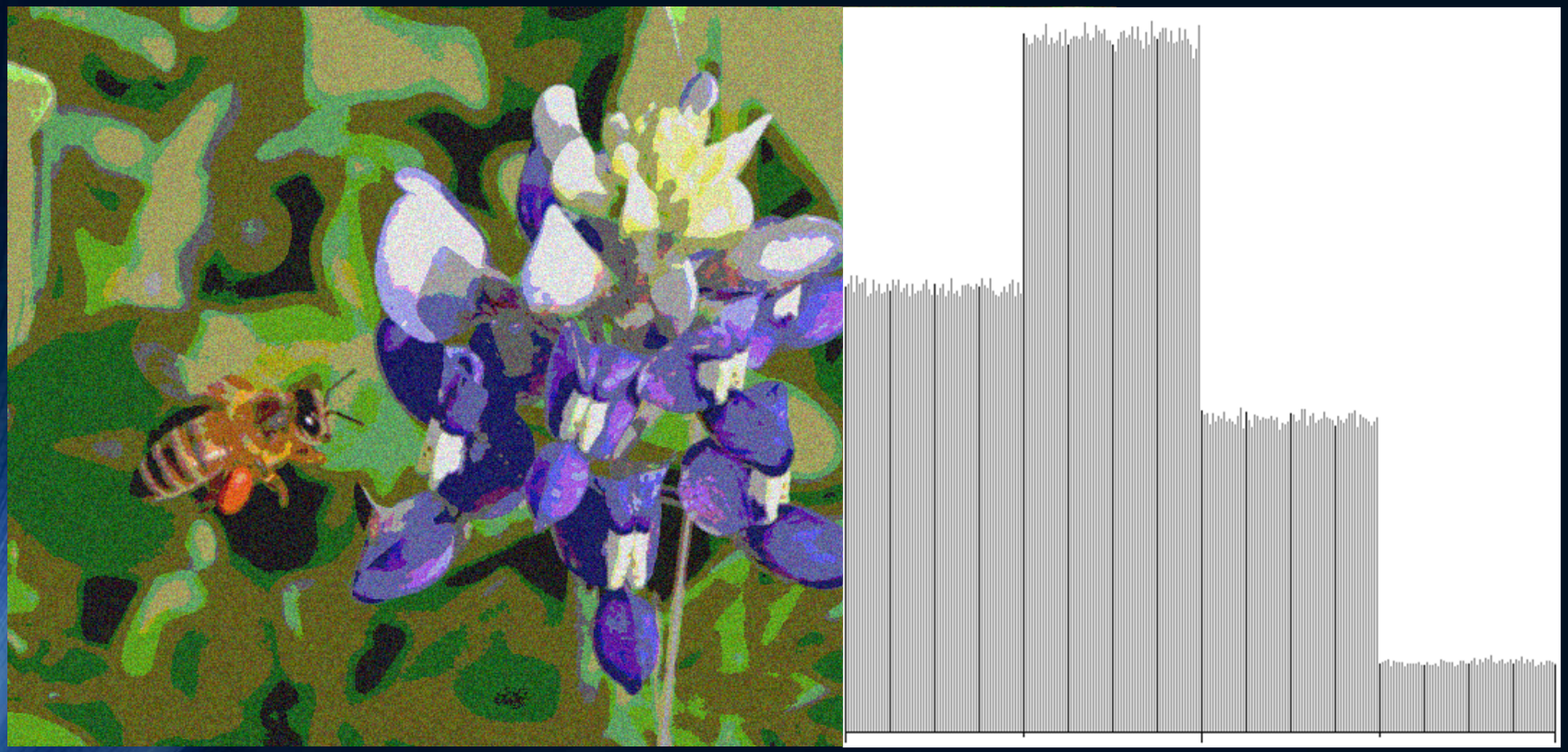
H= 7.62805



Steganalysis - LSB

- 6 bits of randomized data

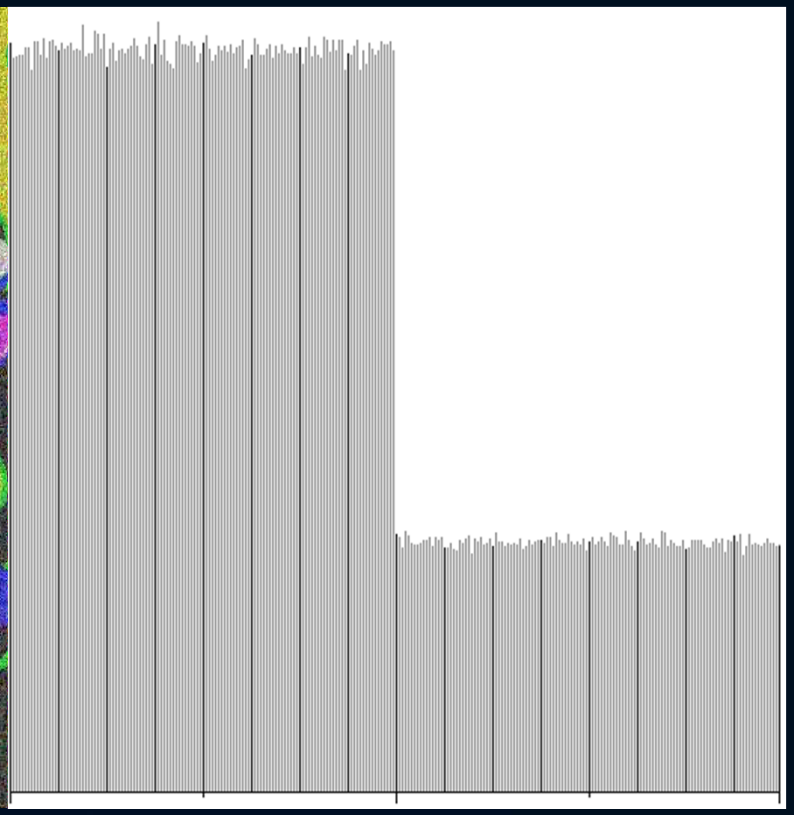
H= 7.71131



Steganalysis - LSB

- 7 bits of randomized data

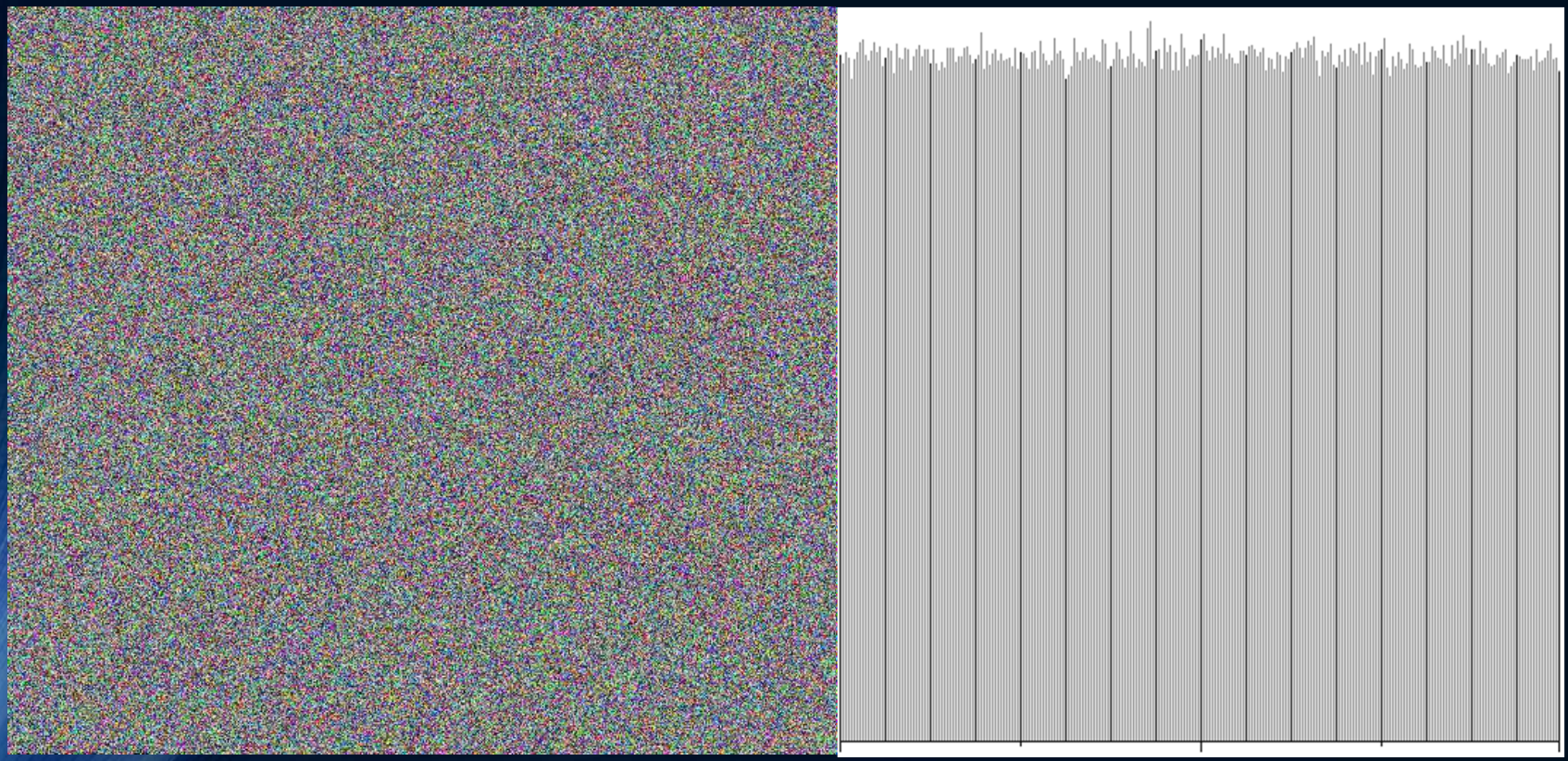
H= 7.81565



Steganalysis - LSB

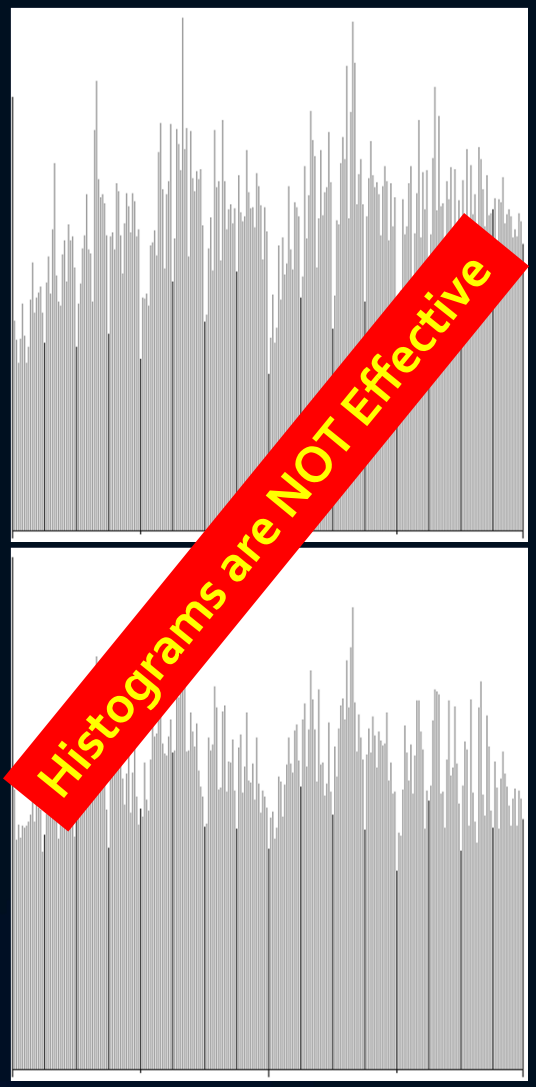
- 8 bits of randomized data

H= 7.99986



Steganalysis - Jpeg

48854 bytes hidden -H= 7.96941

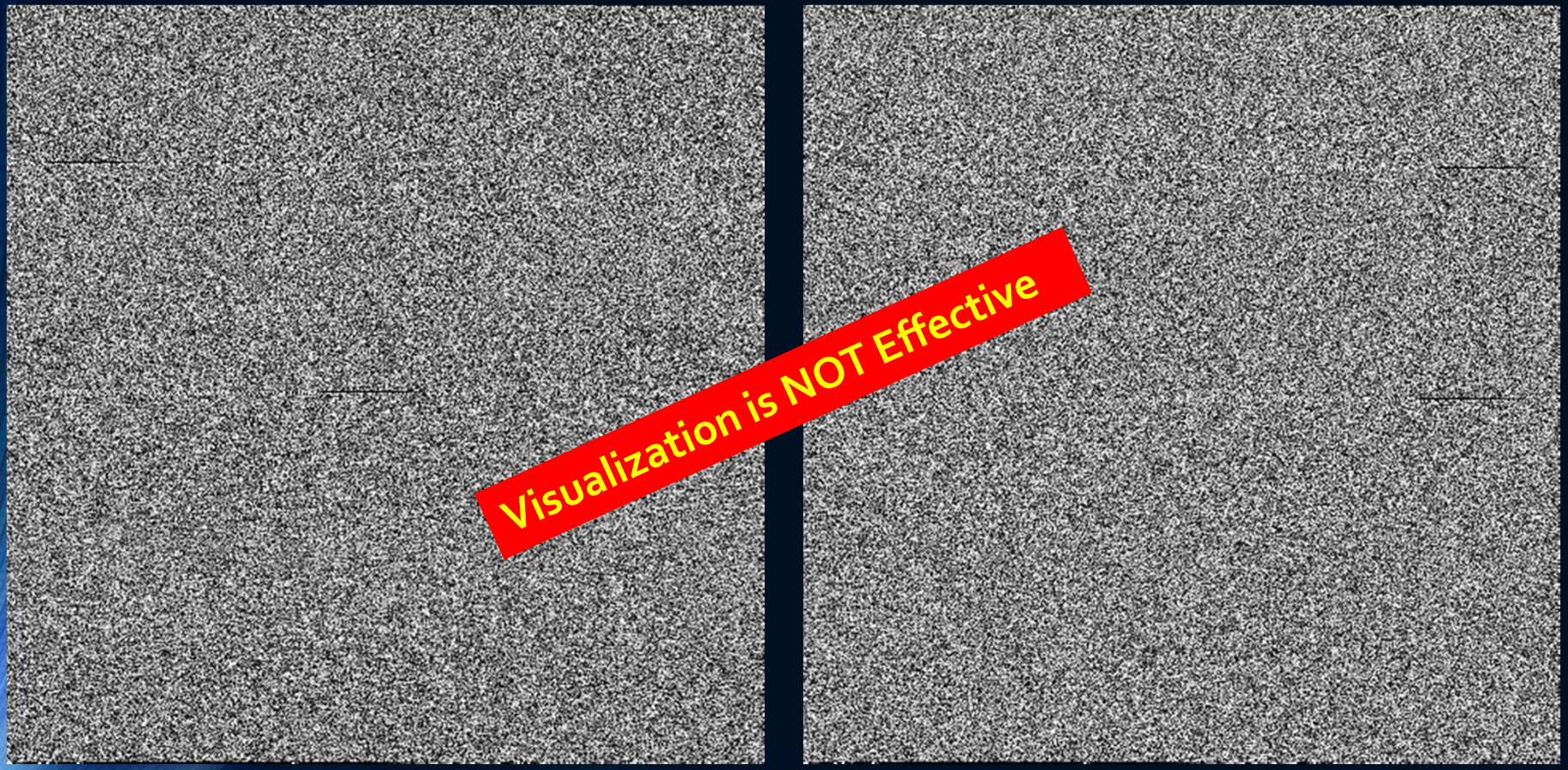


2 bytes hidden - H= 7.98241

Stego Image: 146,256 bytes of hidden data out of 967,442

Steganalysis - Jpeg

- Visualization? One has 2 hidden bytes, other 48,854



Steganalysis - Jpeg

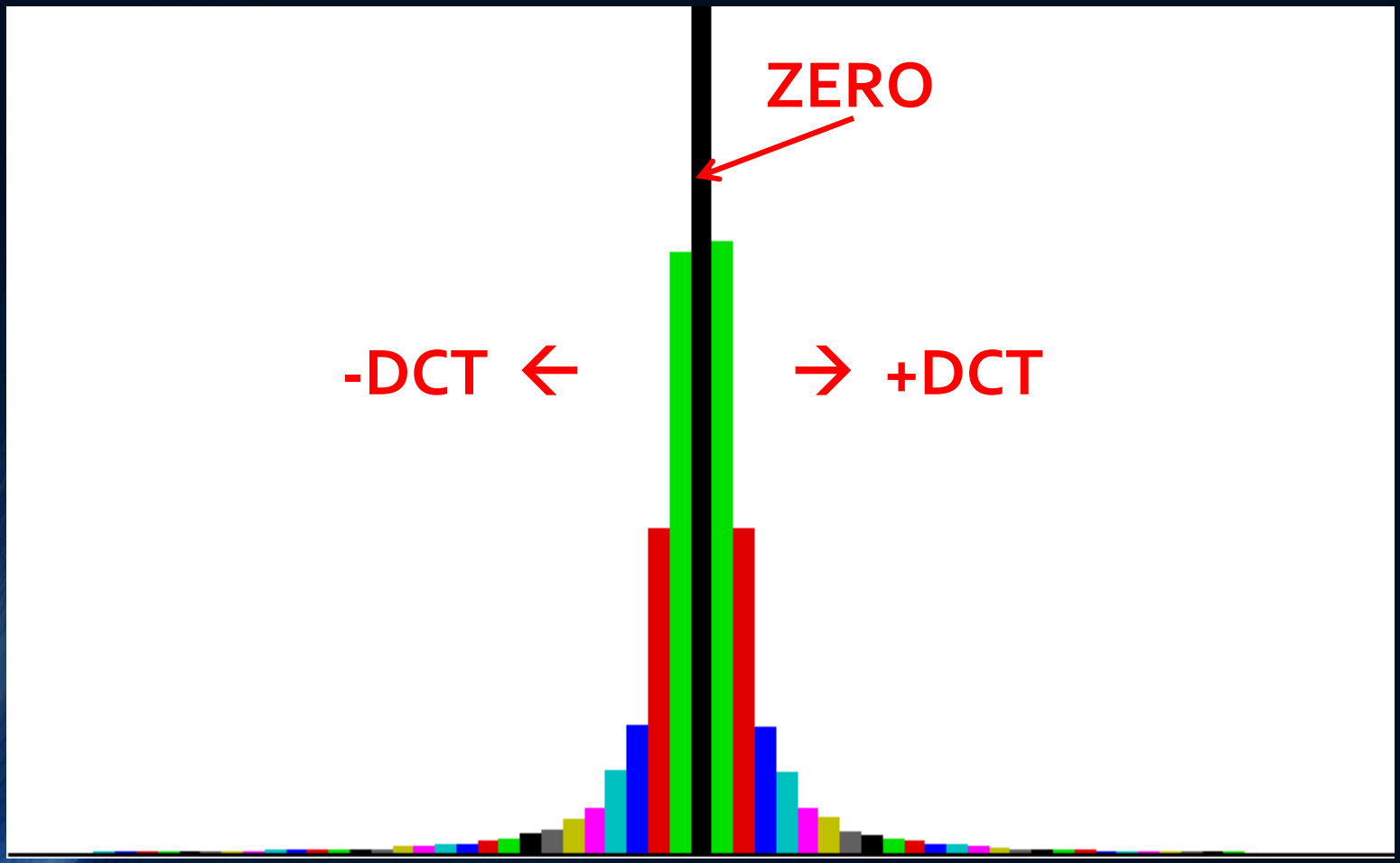
- Solution:

Histogram of DCT coefficients

- DCT Coefficients are generally balanced in a natural image
  - Roughly same number of (+) values as (-) values
- When substituting a bit into the coefficients
  - A "2" becomes a "3", but a "-2" becomes a "-1"
  - A "3" becomes a "2", but a "-3" becomes a "-4"

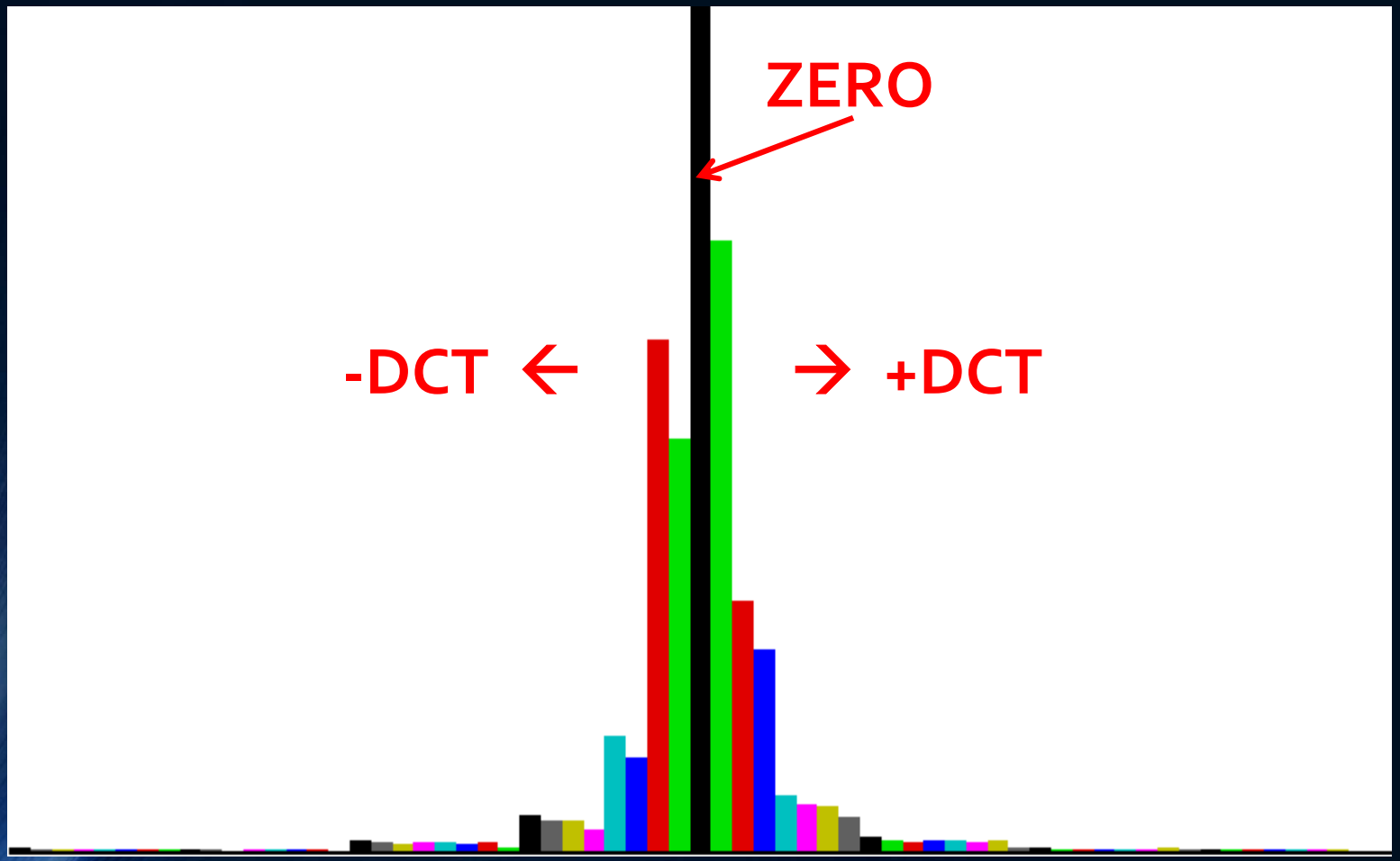
Steganalysis – Jpeg

- Histogram of DCT Coefficients in Natural Image



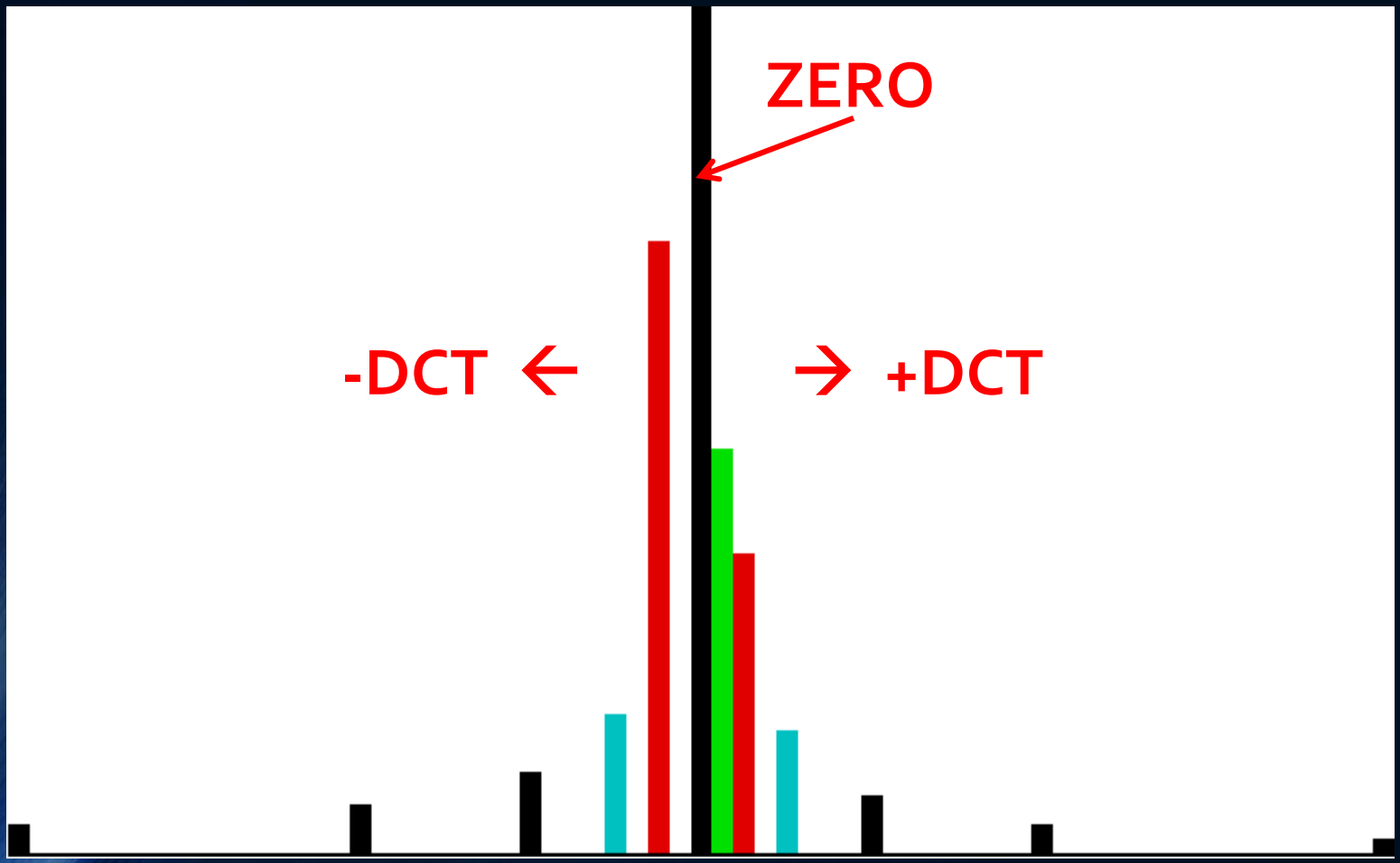
Steganalysis – Jpeg

- Histogram of image with 48,854 bytes hidden



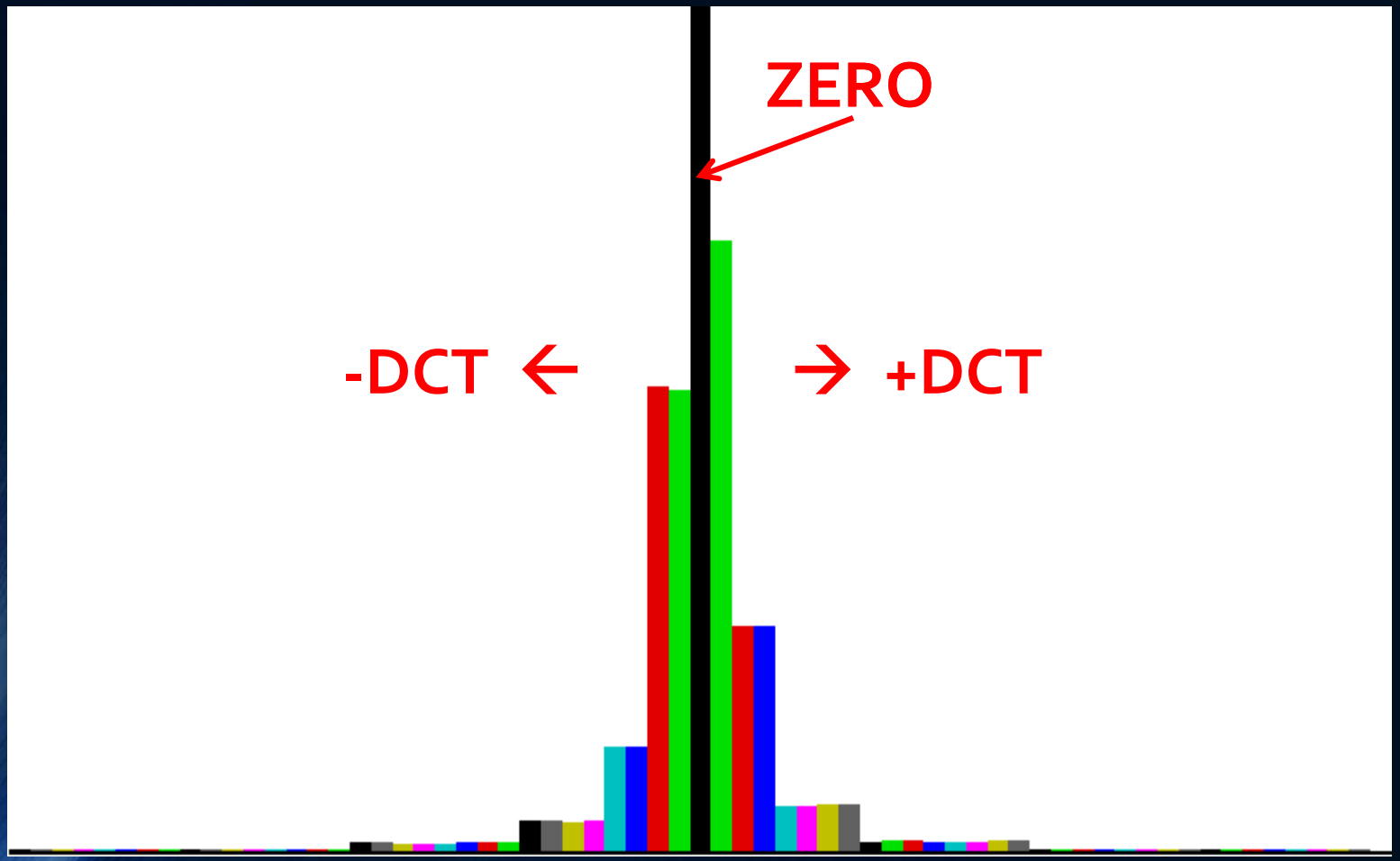
Steganalysis – Jpeg

- Histogram when all hideable bits are set to ZERO



Steganalysis – Jpeg

- Histogram when all hideable bits are randomized



Steganalysis – Jpeg

BEFORE:			AFTER RANDOM		AFTER WIPING		AFTER Hiding	
-0008)	5744	(0.09)	16231	(0.26)	64479	(1.04)	12910	(0.21)
-0007)	19108	(0.31)	16141	(0.26)	0	(0.00)	16482	(0.27)
-0006)	19769	(0.32)	16270	(0.26)	501	(0.01)	16282	(0.26)
-0005)	20359	(0.33)	16338	(0.26)	0	(0.00)	19306	(0.31)
-0004)	29626	(0.48)	18167	(0.29)	36060	(0.58)	24242	(0.39)
-0003)	6434	(0.10)	17893	(0.29)	0	(0.00)	11818	(0.19)
-0002)	15177	(0.25)	21534	(0.35)	43238	(0.70)	17007	(0.28)
-0001)	28061	(0.45)	21704	(0.35)	0	(0.00)	26231	(0.42)
00000)	5689055	(92.04)	5689055	(92.04)	5689055	(92.04)	5689055	(92.04)
00001)	27592	(0.45)	27592	(0.45)	27592	(0.45)	27592	(0.45)
00002)	15188	(0.25)	10798	(0.17)	21629	(0.35)	15192	(0.25)
00003)	6441	(0.10)	10831	(0.18)	0	(0.00)	6437	(0.10)
00004)	29348	(0.47)	22032	(0.36)	87874	(1.42)	29449	(0.48)
00005)	20053	(0.32)	22158	(0.36)	0	(0.00)	20987	(0.34)
00006)	19315	(0.31)	21931	(0.35)	96	(0.00)	18178	(0.29)
00007)	19254	(0.31)	21849	(0.35)	0	(0.00)	19356	(0.31)
00008)	5777	(0.09)	3342	(0.05)	26025	(0.42)	5253	(0.08)

Detection

- The following example illustrates how modification of the DCT coefficients can be detected
- In their unmodified state, the count of coefficients tend to be symmetrical about zero
  - The number of +1 values is roughly equal to the number of -1 values
  - The number of +2 values is approximately the same as the number of -2 values
  - The number of +3 values …

Detection

- WHY do the DCT coefficient values become non-symmetrical?
- When we change a +2 it becomes a +3
  - 00000010_2 → alter LSB → 00000011_2
- And when we change a +3, it becomes a +2
- BUT, when we change a -2, it becomes -1
  - 11111110_2 → alter LSB → 11111111_2
- And when we change a -1, it becomes -2
- So if the number of changes to these coefficients is balanced (i.e. randomized or encrypted data), the +/- balance is destroyed

Detection

- We generally do not use zero for hiding because it would negate a large component of the compression
- PLUS, an unusually small number of ZEROs would be an indication!
- So we cannot use a +1 either, because a change in the LSB results in ZERO
  - The decoder can not tell the difference between an actual zero and a one that was altered to a zero

Detection

- Since +1's are not changing, but -1's are, they become unbalanced too
- For positive numbers, 2's and 3's swap
  - 4's/5's, 6's/7's, etc.
- For negative numbers, it's -2's and -1's
  - -4's/-3's, -6's/-5's, etc.

Steganalysis – Jpeg

- Outguess uses excess capacity to make adjust 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 actual number of changes
  - Does not substitute bits, decrements existing values, maintaining balance
- For these and other techniques, different detection methods needed

Steganalysis - Jpeg

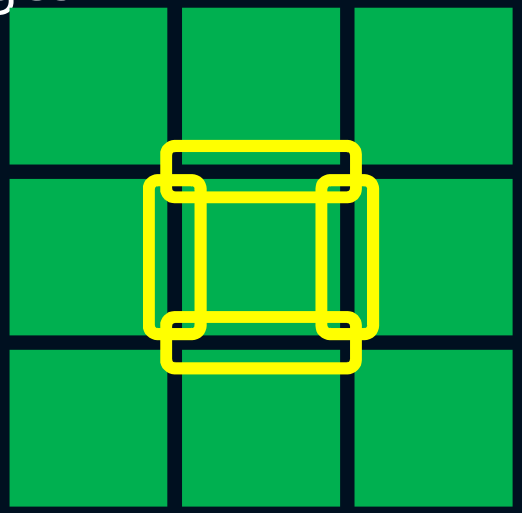
- F5 mitigates this by subtracting from coefficients rather than bit stuffing
  - Proportion is maintained
- Outguess uses unused coefficients to rebalance
- Not 100% effective either
  - Low embedding rates
  - Some techniques do not alter them (Swapping)

Steganalysis – Jpeg

- 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, recompress using same quantization 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 predicted histogram with stego-image histogram
- Able to calculate approximate message length as well

Steganalysis – Jpeg

- When modifying DCT coefficients, spatial discontinuities increase at the 8x8 boundary (in the image not DCT)
  - i.e. "blockiness"
- Measure the discontinuity at the 8x8 edges
  - Most 8x8 blocks have 4 boundary edges
- Use the cropped image as estimate
- Measure blockiness of both and compare



$$B = \sum_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum_{j=1}^{N} \left| g_{8i,j} - g_{8i+1,j} \right| + \sum_{j=1}^{\lfloor (N-1)/8 \rfloor} \sum_{i=1}^{M} \left| g_{i,8j} - g_{i,8j+1} \right|$$

Steganalysis – Jpeg Extraction

- 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

Steganalysis – Jpeg Destruction

- Sterilization of data hidden in a jpeg is easy
- Could ZERO or RANDOMIZE the LSBs of the DCT coefficients
  - But that's too hard
- Could hide another message on top of prior message
  - Similar to randomization
  - Use the same tool if known
- Resize the image – EASY!
  - NOT in multiples of 8!
  - Resize by a single (or 2) horizontal columns and vertical rows
  - Completely changes DCT coefficients

Steganalysis – Jpeg Compatibility

- One example of a specific steganalysis technique
- From the paper "Steganalysis Based Upon Jpeg Compatibility"
- Technique reliably detects spatial-domain steganography (LSB) that has been applied to images *previously* stored as a JPEG
- The JPEG compression algorithm introduces a unique fingerprint that serves as a fragile watermark
- Modifying a single LSB can be detected

Steganalysis – Jpeg Compatibility

- NOT applicable to algorithms that embed in the LSBs of the DCT coefficients (like Chang's algorithm)
- Can estimate the size of the message and identify which pixels carry the message
- Paper describes a technique to recover the quantization matrix used in the JPEG compression
- The paper describes the technique for grayscale images, but it may be extended to color images

Steganalysis – Jpeg Compatibility

a. B = [B_{raw}] (round or truncate value)
b. Take the L^2 norm – it should be <= 16
    1. | | B-B_{raw}| | <= 16
    2. inner bars are magnitude (absolute value)
    3. outer bars indicate L^2 norm which is the sum of the squares
    4. since | B-B_{raw} | <= ½, the sum of 64 squares <= 16
        a. $(½)^2 + (½)^2 + (½)^2 \ldots (½)^2 = 16$ and that is a maximum
        b. this basically calculates the difference due to rounding

$$S = \sum_{i=1}^{64} \left| DCT(B)_i - Q_i * round\left(\frac{DCT(B)_i}{Q_i} \right) \right|$$

Steganalysis – Jpeg Compatibility

- If S > 16, the block is not compatible with JPEG compression using Q
  - May indicate presence of hidden data
  - If S <= 16 can check one additional equation (not shown here for brevity)
- Repeat above steps for all blocks 1 – T
- If all blocks are incompatible, image may not have been stored as a JPEG

Steganalysis – Jpeg Compatibility

- It may have been a JPEG that was modified which destroys the jpeg signature
  - Affine transformations may have been applied
  - Perhaps the image was cropped by a few pixels
  - Could repeat the above steps, offsetting the blocks by 1-7 pixels in both x and y directions (64 possibilities)
- This technique works for all steganographic spatial-domain methods
- Does NOT work for DCT embedded data
- Do NOT use as a cover image, a file that was previously JPEG compressed
  - Cropping the image destroys the JPEG signature

Questions & Comments

References

- Conti, Greg; Grizzard, Julian; Ahamad, Mustaque; Owen, Henry; Visual Exploration of Malicious Network Objects Using Semantic Zoom, Interactive Encoding and Dynamic Queries. Georgia Institute of Technology
- "Steganalysis Based Upon Jpeg Compatibility", Jessica Fridrich
- "An Effective Algorithm for Breaking F5", Hong Cai, Sos Agaian, University of Texas at San Antonio
- "Pairs of Values and the Chi-squared Attack", Christy Stanley, Iowa State University
- "Steganalysis of JPEG Images: Breaking the F5 Algorithm", Jessica Fridrich, Miroslav Goljan, Dorin Hogea, SUNY Binghamton, NY
- "New Methodology for Breaking Steganographic Techniques for JPEGs", Jessica Fridrich, Miroslav Goljan, Dorin Hogea, SUNY Binghamton, NY

References

- http://en.wikipedia.org/wiki/YCbCr
- "Embedding Robust Labels into Images for Copyright Protection", Jian Zhao, Eckhard Koch
- "A Method of Embedding Binary Data into JPEG Bitstreams", Hiroyuki Kobayashi, Yoshihiro Noguchi, Hitoshi Kiya
- "High Capacity Data Hiding in JPEG Compressed Images", Chang, C.C. and Tseng, Hsien-Wen
- Compressed Image File Formats, JPEG, PNG, GIF, XBM, BMP, John Miano, Addison Wesley

References

- "Defending Against Statistical Steganalysis", Niels Provos
- **"**A JPEG-Based Statistically Invisible Steganography**",** Qingzhong Liu, Andrew H. Sung, Zhongxue Chen, Xudong Huang
- "F5 - A Steganographic Algorithm - High Capacity Despite Better Steganalysis", Andreas Westfeld, Technische Universit¨at Dresden
- "Detection of Hiding in the LSB of DCT Coefficients", Mingqiao Wu, Zhongliang Zhu, and Shiyao Jin
- "STEGANALYSIS OF BLOCK-DCT IMAGE STEGANOGRAPHY", *Ying Wang and Pierre Moulin*, University of Illinois at Urbana-Champaign
- "Attacking the OutGuess", Jessica Fridrich, Miroslav Goljan, Dorin Hogea