Steganography & Steganalysis

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

BASIC HIDING - SUBSTITUTION

Substitution Overview

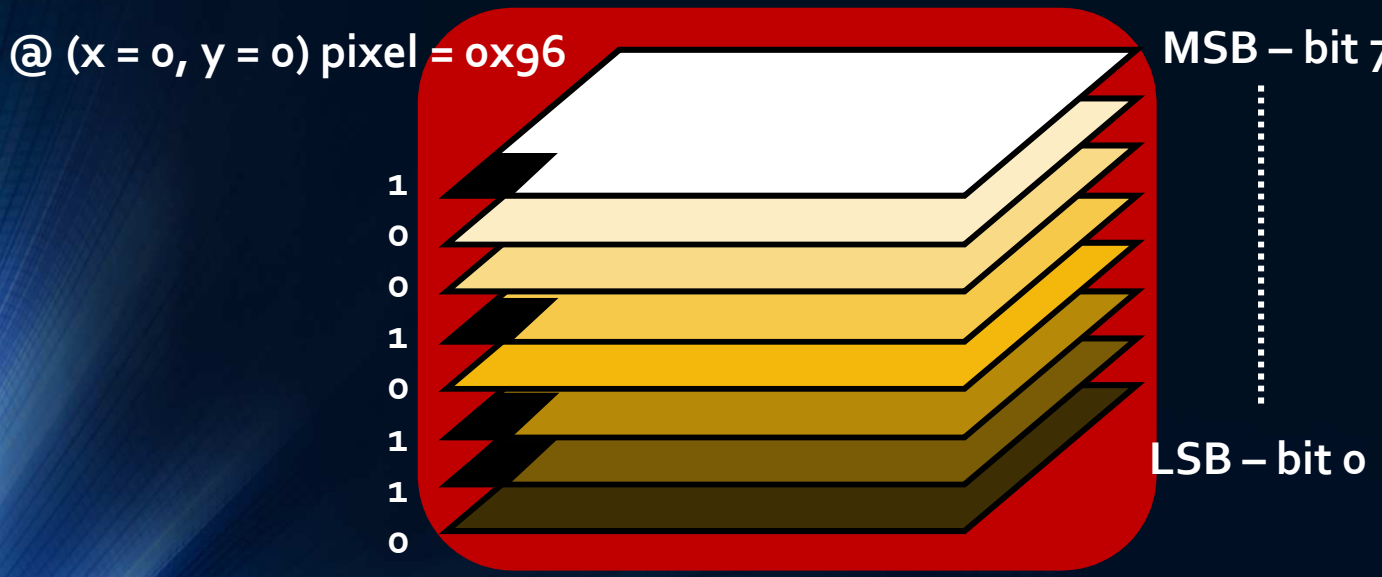
- Picture-in-Picture
- Flying the Bit Planes
- LSB for BMP
- LSB in Wave

Substitution – Least Significant Bit

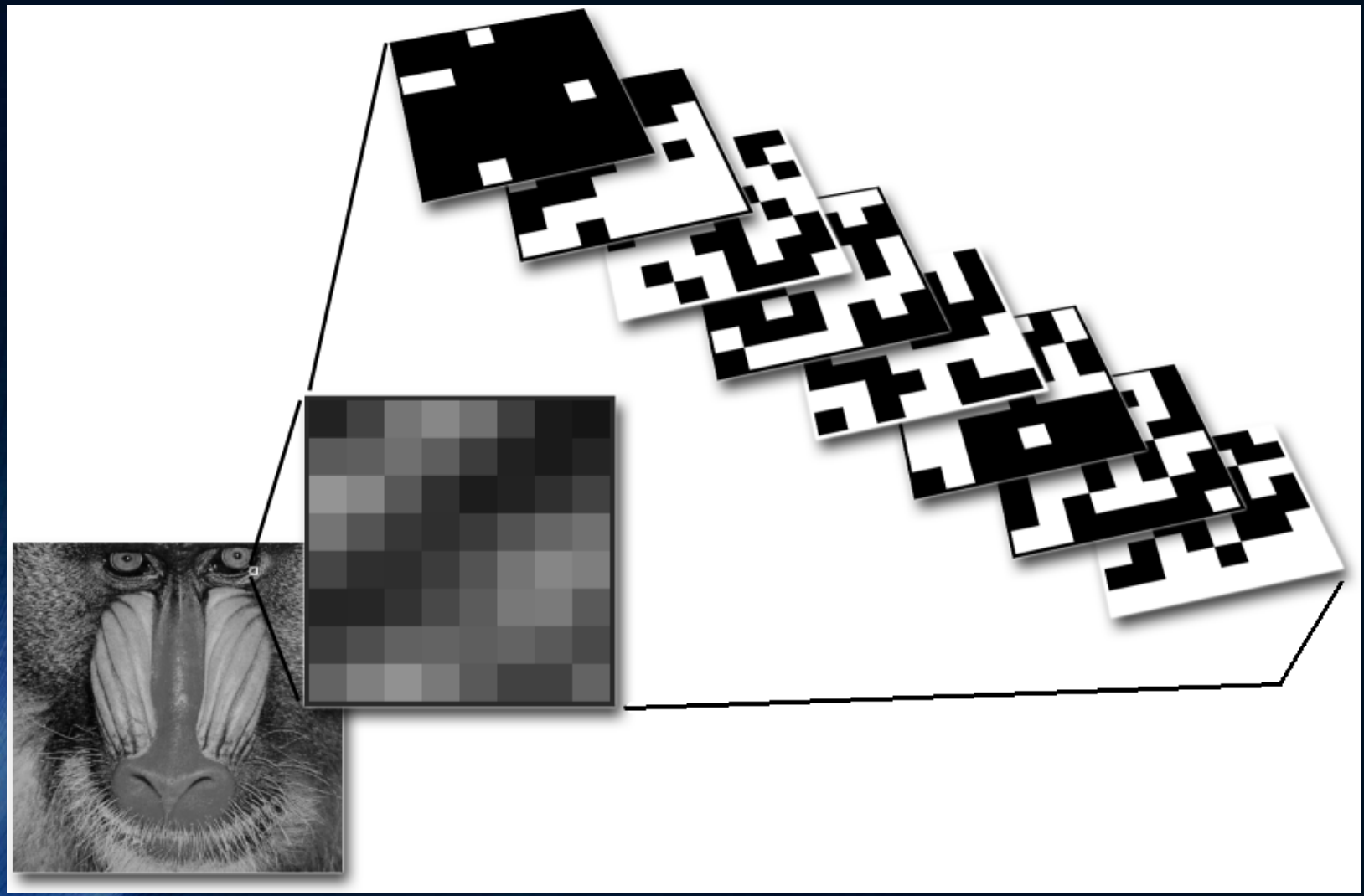
- Substitution techniques replace original data with message data
- Generally, substitution is done on data insignificant to the quality of an image or audio sample
- The LSB approach is common and easily understood
- LSB works with both images and audio
- It does not have to be only the least significant bit, it can be the least 2 or 3 or more LSBs

Substitution - LSB

- Consider a 24-bit image using the RGB model
  - It could be viewed as having 8 bit planes, one for each color
  - Below is a graphical representation of the 8 bit planes, showing which bits are set at the coordinates x = 0, y = 0



Substitution – 8x8 Block



Substitution - LSB

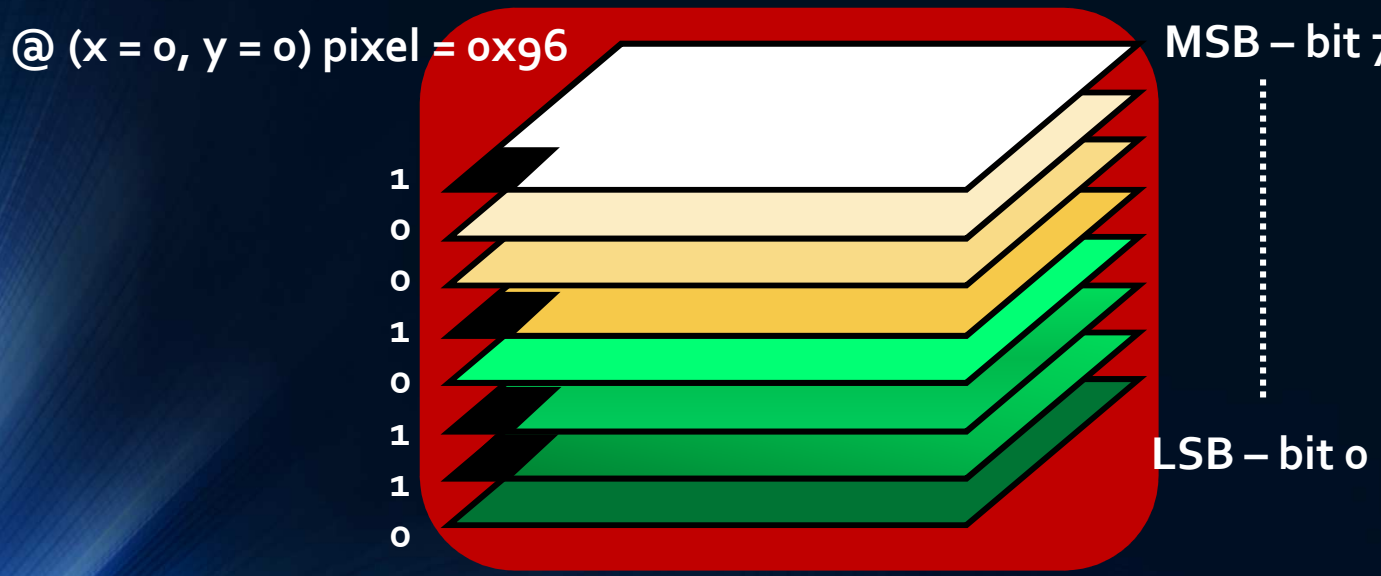
- When the one least significant bit is altered, is that noticeable?
  - How about using the 2 or 3 or 4 bits?
- At some point it becomes perceptible
- Comparing the original and the stego-image side by side , it will be more noticeable, sooner
- Each pixel is one of 256 shades of that color (2^8=256)
- Our eyes cannot distinguish the difference between 0x96 or 0x97
  - In most cases, it takes 4 or 5 bits to be easily perceived

Substitution - LSB

- Another consideration is characteristics of the cover image (or audio sample)
  - Check the variance of surrounding pixels
    - if very low (monochrome color) a change to the LSB may easily be visible
    - if the variance is very high (on a boundary) it may also be visible
  - The rate of change of a color in an image is considered a frequency
    - Black (0, 0, 0) to white (255, 255, 255) would be the max
  - For audio, the rate of variance in the amplitude is also called frequency
- A complex cover is generally better than a very uniform cover

Substitution - LSB

- We can literally hide one image behind another image – just like painting over another painting
- This could be called Picture-in-Picture (PiP)
  - It is not a serious technique, but great for illustration



We will Hide This …

Inside this …

Then That in This …

Can YOU See a Difference?

- The Dalmatian is hiding in 4 bits of Mandrill

Substitution - PiP

- Other images with more solid backgrounds DO NOT provide the same level of imperceptibility
  - To maximize capacity while maintaining imperceptibility, the cover image is a consideration

You CAN See a Difference!

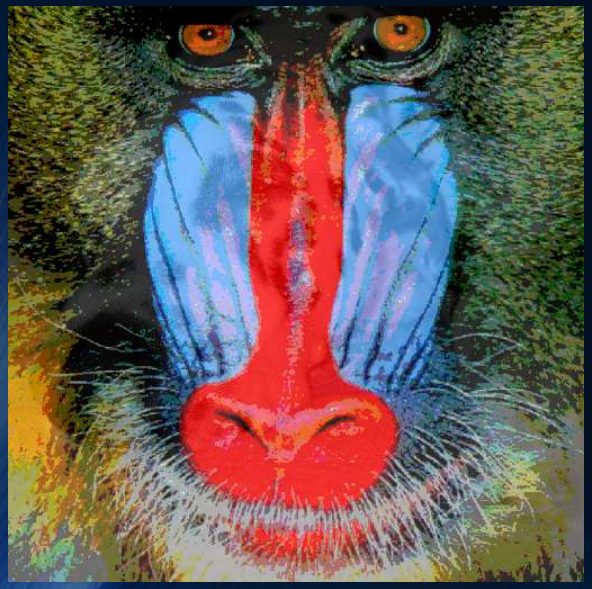
- More uniform colors in cover is NOT as effective

Limitations

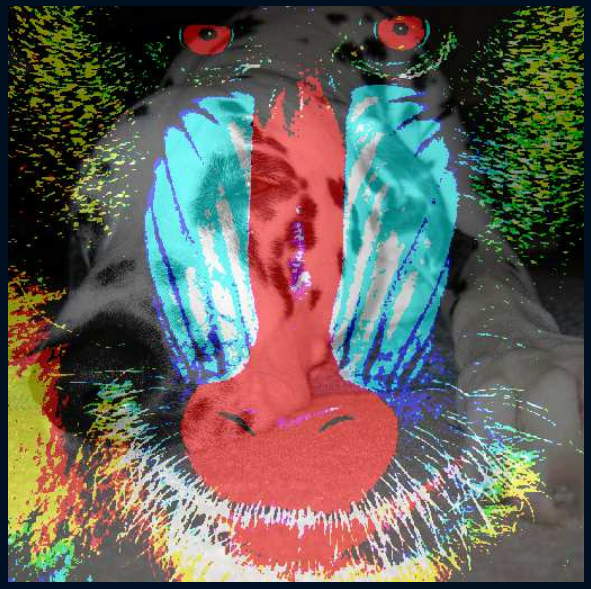
4

5

6



7



Limitations

4

5

6

7

Substitution - PiP

- This particular technique substitutes image bits of one picture into another
- Both pictures must be the same size
- More typical is to substitute bits from the message one by one
  - The message can be anything

When the Message is Encrypted

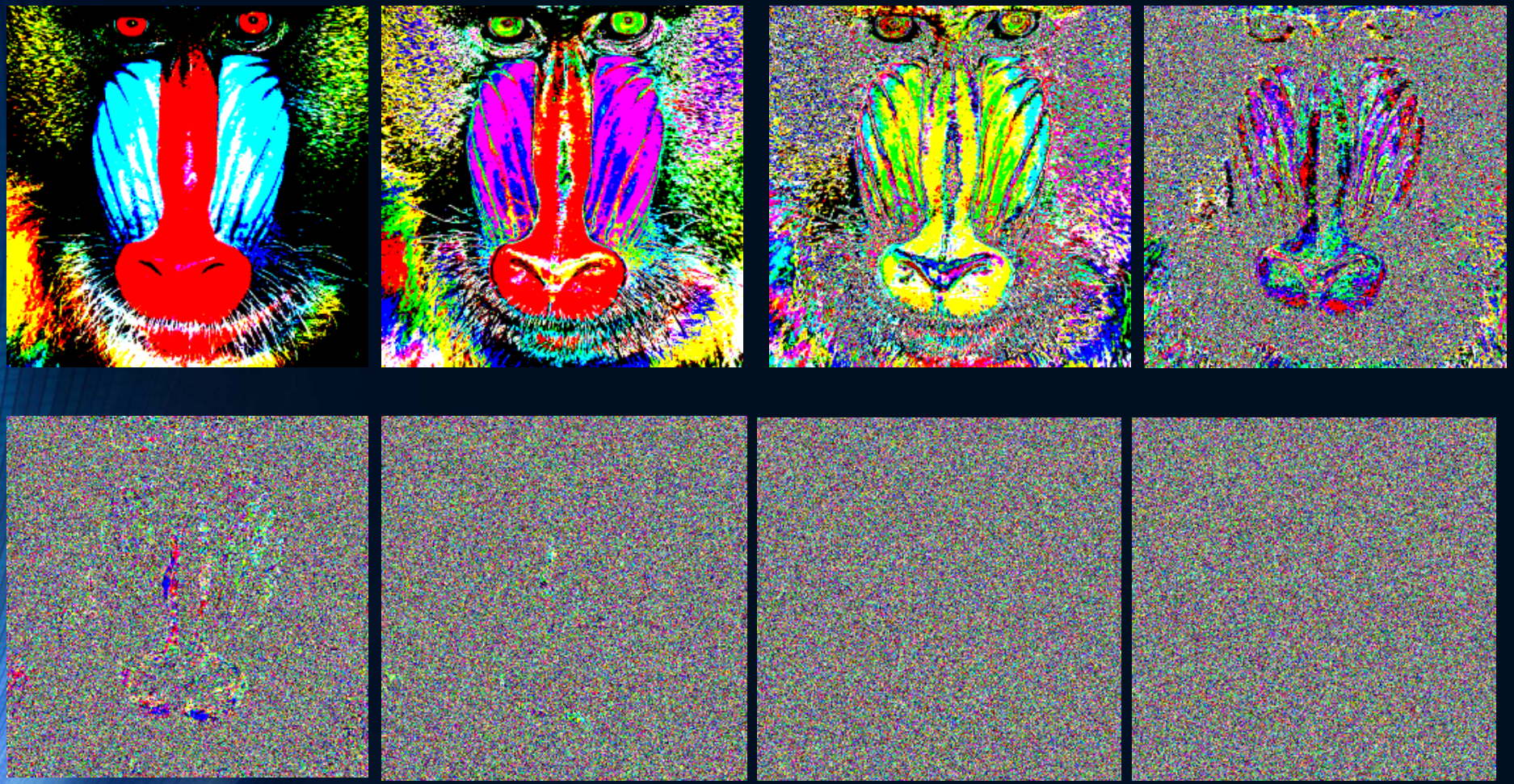
4

5

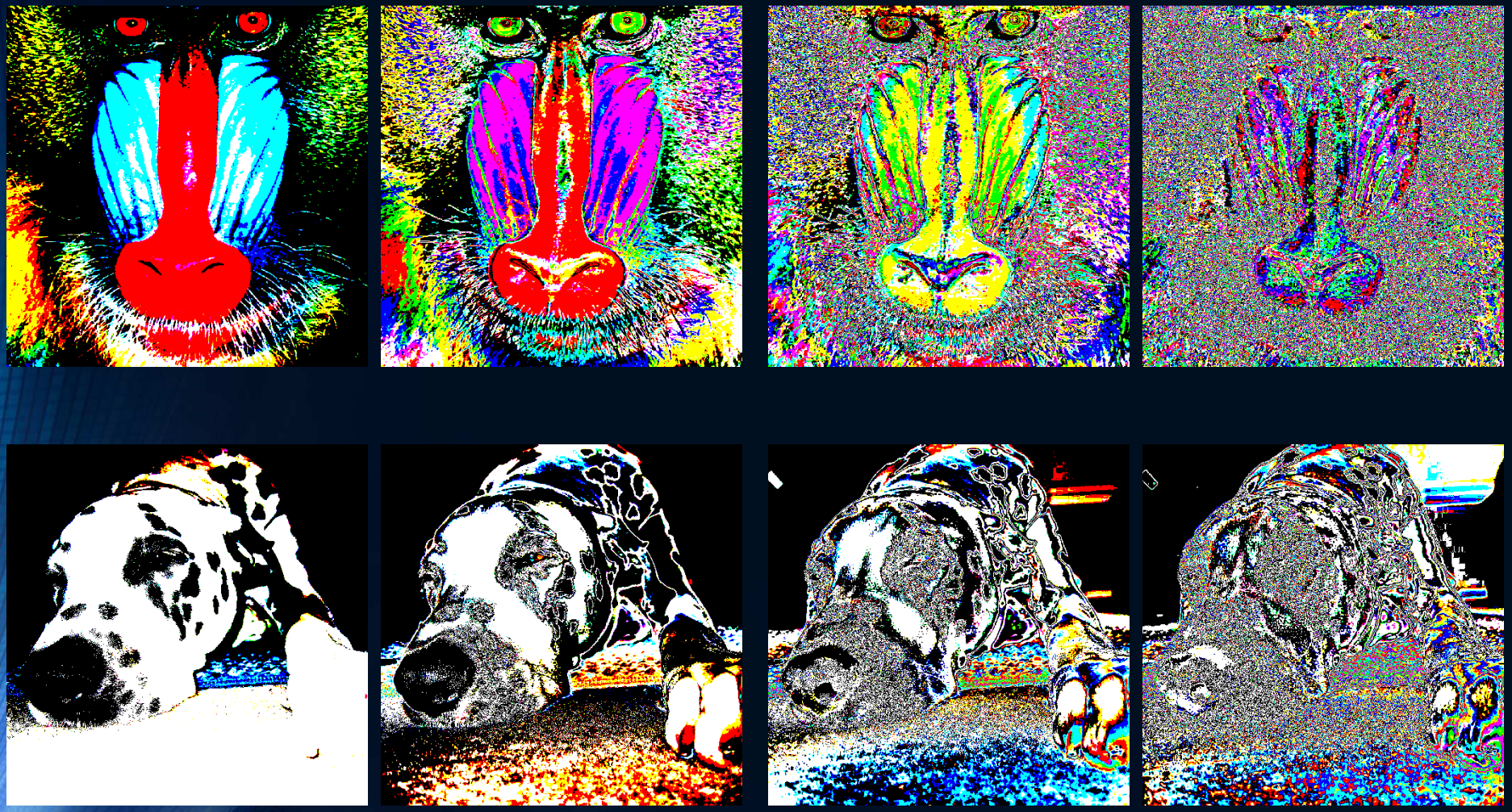
6

7

Sliced Into Bit Planes (Clean Image)



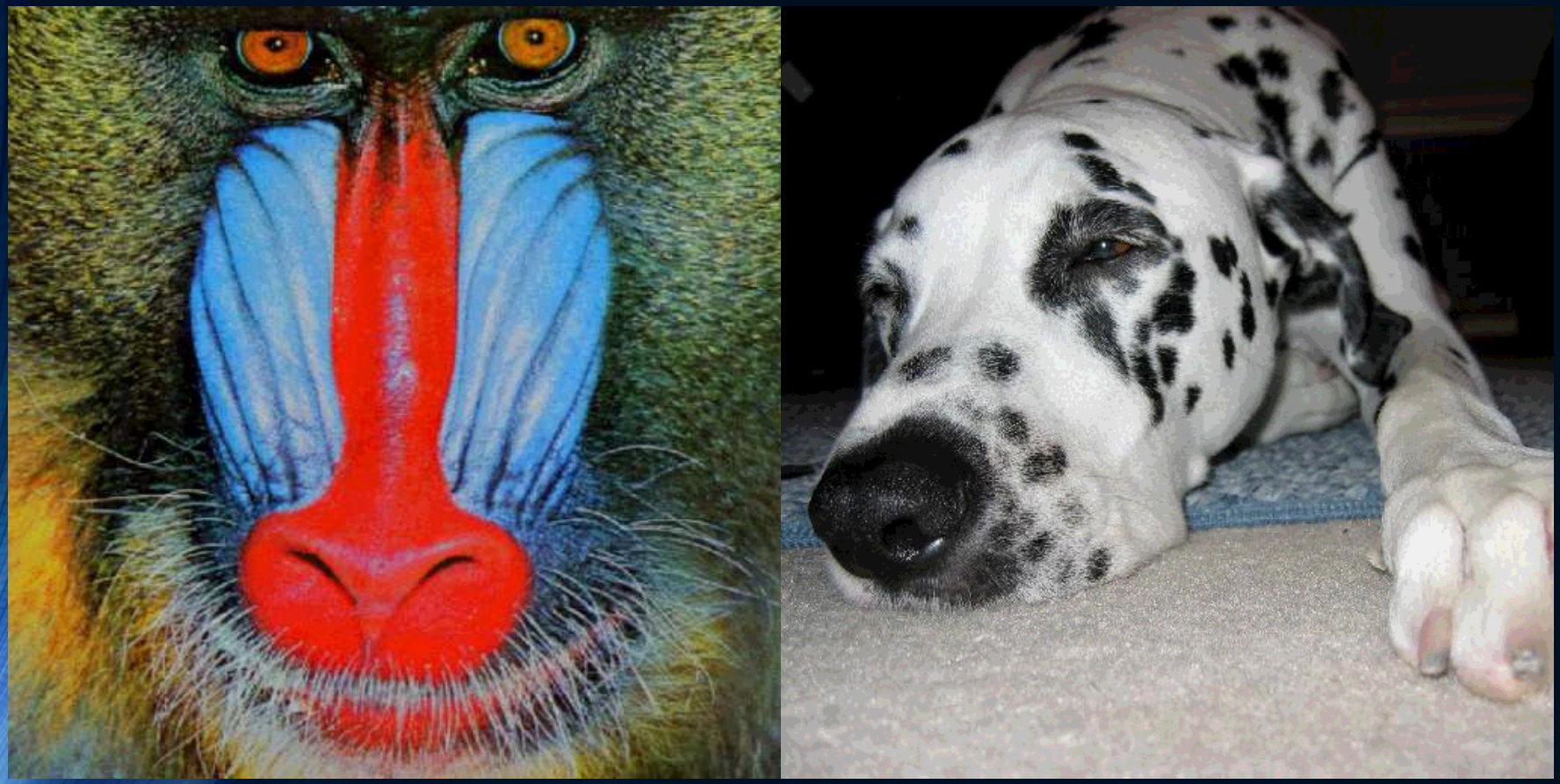
Sliced Into Bit Planes (Hidden Image)



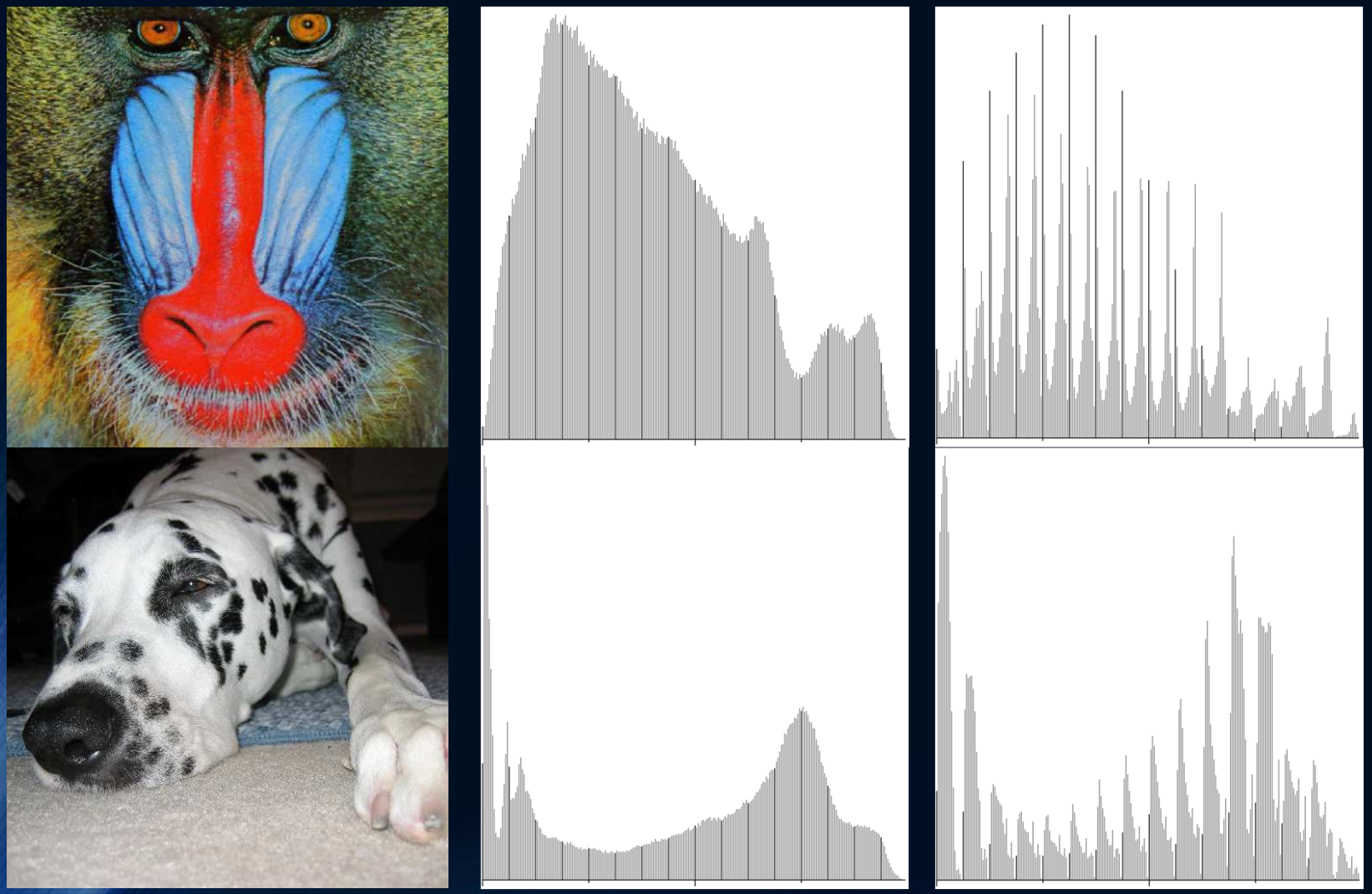
Sliced Into Bit Planes (hidden image)

Substitution - LSB

- A closer look at the 4-bit images



Easily Detected with a Histogram



Substitution - LSB

- Typically, an arbitrary message would be read as a bit stream and embedded
- Basic LSB Algorithm:
  - Read message bit(s)
  - Read cover bit(s)
  - If cover bit(s) == message bit(s), loop
  - If cover bit(s) != message bit(s), replace cover bit(s) with message bit(s), loop
- Loop until there are no more message bits or cover bits, whichever comes first!

Substitution - LSB

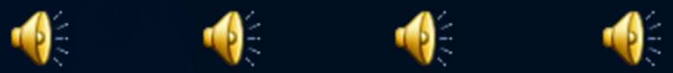
- Hiding data with characteristic statistical properties results in the cover file taking on some of those same statistical characteristics
- In some cases, hiding plain text is worse (statistically) than hiding encrypted or compressed text

Substitution – LSB Audio

- LSB with uncompressed audio works well just like with images
- Capacity is fairly high, around 50%, with minimal perceptual distortion
- It is easily detectible statistically too
- Generally, a "noisy" cover file is better to minimize audible perception
  - This means hard rock or live microphone recordings

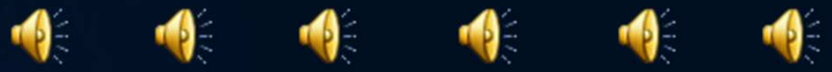
Substitution – LSB Audio

- Jazz (original), 8, 9, 12 bits



Can you tell when the stego message ends?

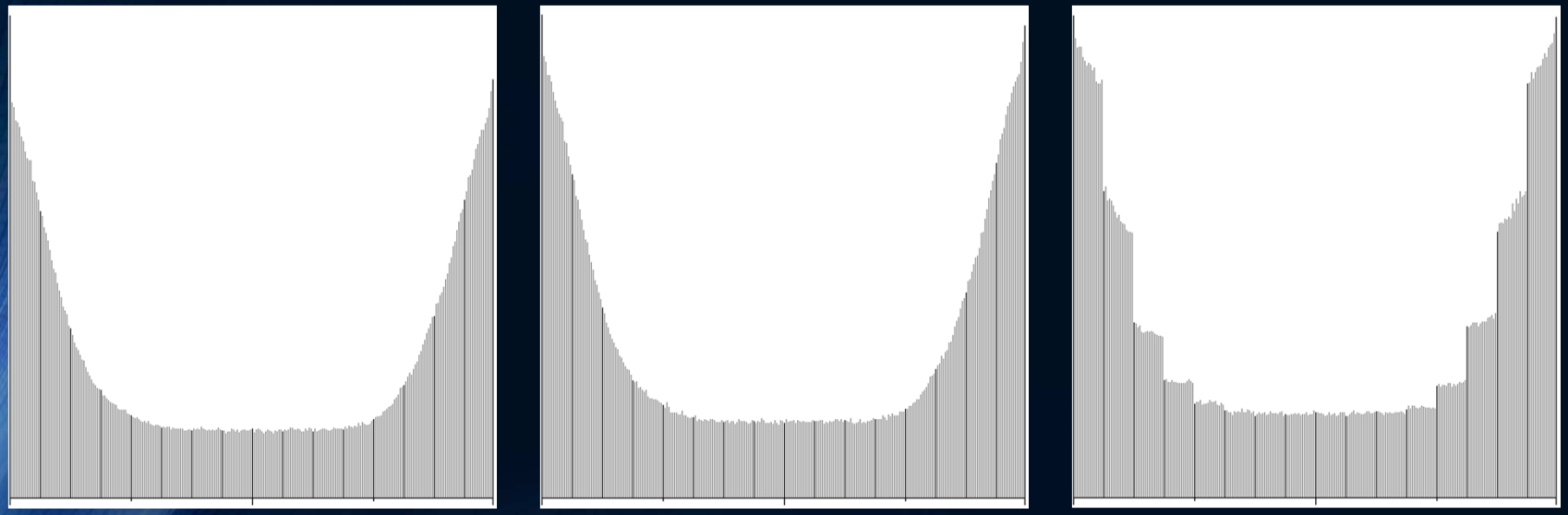
- Rock (original), 8, 9, 10, 11 and 12 bits



- Rock and Jazz with Text, 8 bits

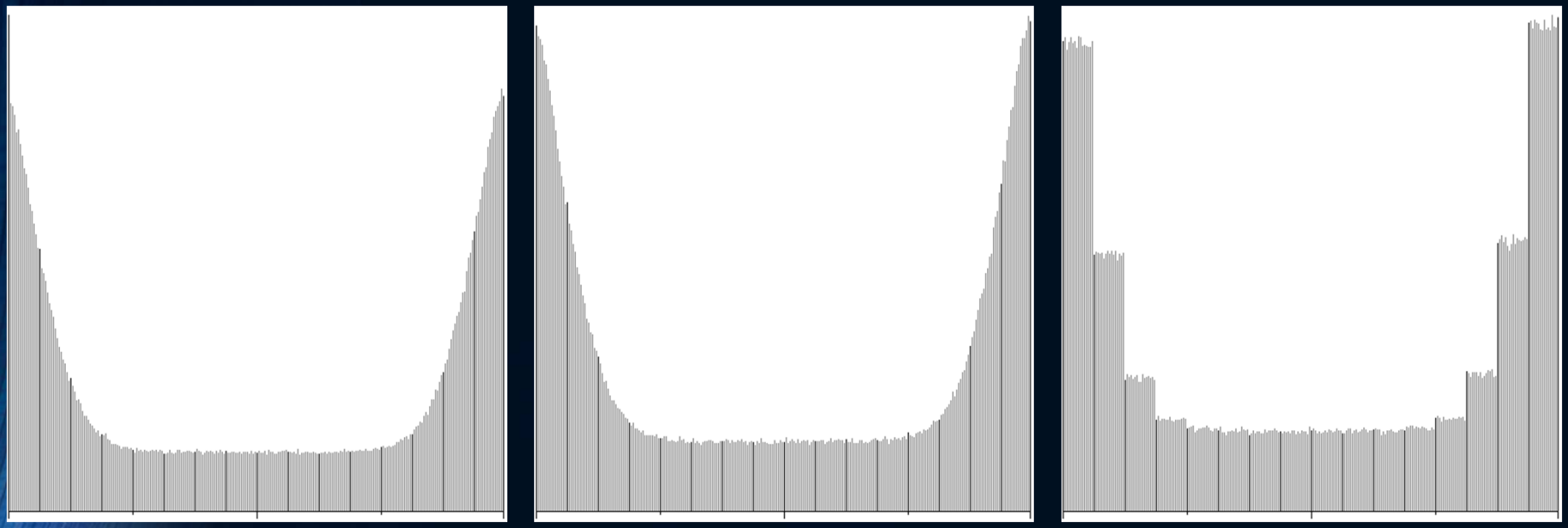
Substitution – LSB Audio

- Audio Histograms are effective too
  - For encrypted data, requires higher embedding rate to be discernible
  - Hiding 8 bits of English text is completely obvious
- Jazz (original), 8, 12 bits



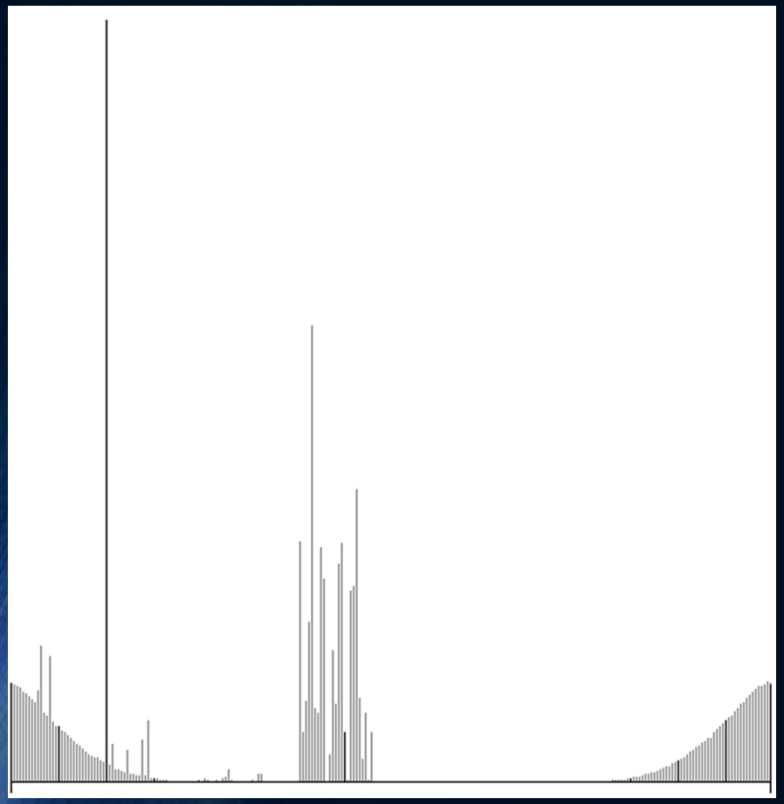
Substitution – LSB Audio

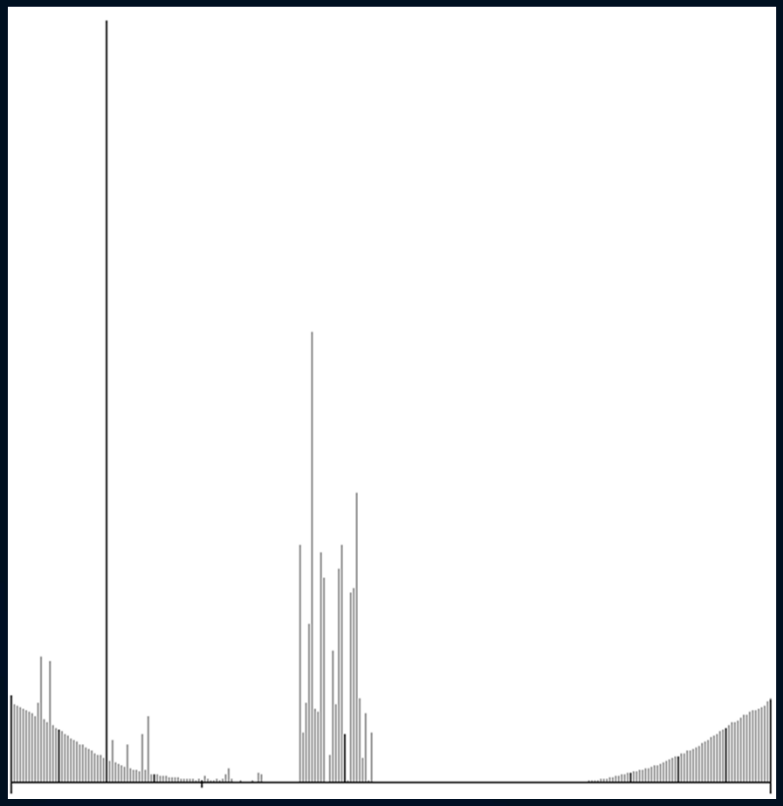
- Rock (original), 8, 12 bits



Substitution – LSB Audio

- Rock and Jazz with 8 bits of English Text





LSB Security

- Simple LSB alterations are trivial to detect in images
- If the data is hidden sequentially and the message does not use the full capacity, statistical characteristics will differ sharply between used/unused sections
  - This can even be clearly visible in the image
- Hiding too many bits results in fuzziness and discolorations
- For audio, it results in audible noise
  - If your source is a cassette tape, you can hide more!

LSB Security – Sequential 3 bits

LSB Security – Sequential 4 bits

LSB Variations – Random Interval

- The *random interval method* complicates detection
  - A secret key seeds a pseudo-random number generator
  - The PRNG must be the same at both ends
  - from each pixel, a pseudo-random distance to the next pixel is chosen
- More difficult to extract and visually detect message
- Still may have non-uniform statistical properties in first part of cover
- Capacity is reduced
- Can't predict exact capacity beforehand

LSB – PRNG Pixel Selection

- Use a Pseudo-Random Number Generator (PRNG) to determine each pixel location
- Stego-image will have more uniform statistical properties
- Length of message must be much, much less than the length of the cover to avoid collisions
  - Probability of collision increases exponentially with increasing message length
  - Ignoring collisions likely to cause message loss since a colliding message bit may overwrite original message bit
- Could select pixel from a list, then delete that element from the list

LSB Variations – Group Pixel Selection

- Use a group of pixels for a single message bit
- Get the parity, if it does not match the message bit, then change one bit out of the group
  - can choose the "best bit" this way to reduce perceptibility
- Could average the pixels and have that result contain the message bit(s)
- Capacity is reduced

LSB - Capacity

- Capacity is based on the number of pixels, the number of bytes per pixel, and the number of bits to be embedded per byte
- For a 24-bit image, there are 3 bytes per pixel
  - Note: Each line in a 24-bit bmp file MUST align to a 4-byte boundary
  - A width of 5 pixels would require 15 bytes, so a padding byte is added to make the line 16 bytes
  - The padding byte is NOT displayed
- For an 8-bit grayscale or paletted color image, there is 1 byte per pixel

LSB - Robustness

- Robustness is the measure of how well a stego file will retain its message after manipulation
- LSB techniques are not robust
- It is trivial to "sterilize" the message without destroying the image
  - Set all LSB's to zero
  - Set all LSB's to one
  - Set all LSB's to random values

Palette Based Images

- A palette-based image is one in which each pixel value is an index into a table of colors
- Each table entry consists of three 8-bit values for an RGB color
- If each pixel is 8 bits, there are up to 256 table entries
- If each pixel is 16 bits (as with some bmp files) there are 65536 table entries
  - change a 24 bit bmp file to 16 bit
  - no matter how small the image, the file will be at least 64 kbytes
- Could hide in the pixels or in the palette

Paletted Image Hiding

- The palette does not need to be sorted in any particular order
  - Resorting the palette would not alter the perception of the image at all
- If the palette were to be sorted by luminance value, the LSB technique could be used
- A grayscale image is nothing more than a palette where all entries have equal RGB values
- Typically, it's palette is sorted in ascending order
  - black: 0,0,0
  - dark gray: 32,32,32
  - light gray: 192,192,192
  - white: 255,255,255
- Altering the LSB here is the same as using a palette sorted by luminance

Paletted Image Hiding- EzStego

- Copy the palette from the image
- Sort a temporary copy of the palette by luminance
  - Original: middle, black, white, dark, light
  - Sorted: black, dark, middle, light, white
- Find index of a pixel's RGB color
  - Ex: pixel's index was 0 (middle)
  - Now in sorted palette, index is 2 (middle still)
- Replace the LSB of the index with the message bit
  - Ex: lsb was zero (010_2 – lsb is zero)
  - Replaced by a one, now index is 3 (light)
- Now index *may* point to new RGB color
  - if LSB the same as message bit, points to same color
  - if LSB different, likely similar in color, though not always

Paletted Image Hiding- ExStego

- Find the index of the new RGB color in the original palette
  - Original: middle, black, white, dark, light
  - Sorted: black, dark, middle, light, white
  - Ex: new color is light, so index of that in original palette is 4
- Change the pixel to the index of the new color
  - So index in stegged pixel is 4
  - Original palette is left unchanged
- Now palette is not reordered, and nearest luminance is chosen
- Recovery is done by getting LSBs of pixels

Paletted Image Hiding

- Sorting by luminance and doing LSB manipulation on a color image has drawbacks
- These two colors below have the same luminance

Paletted Image Hiding

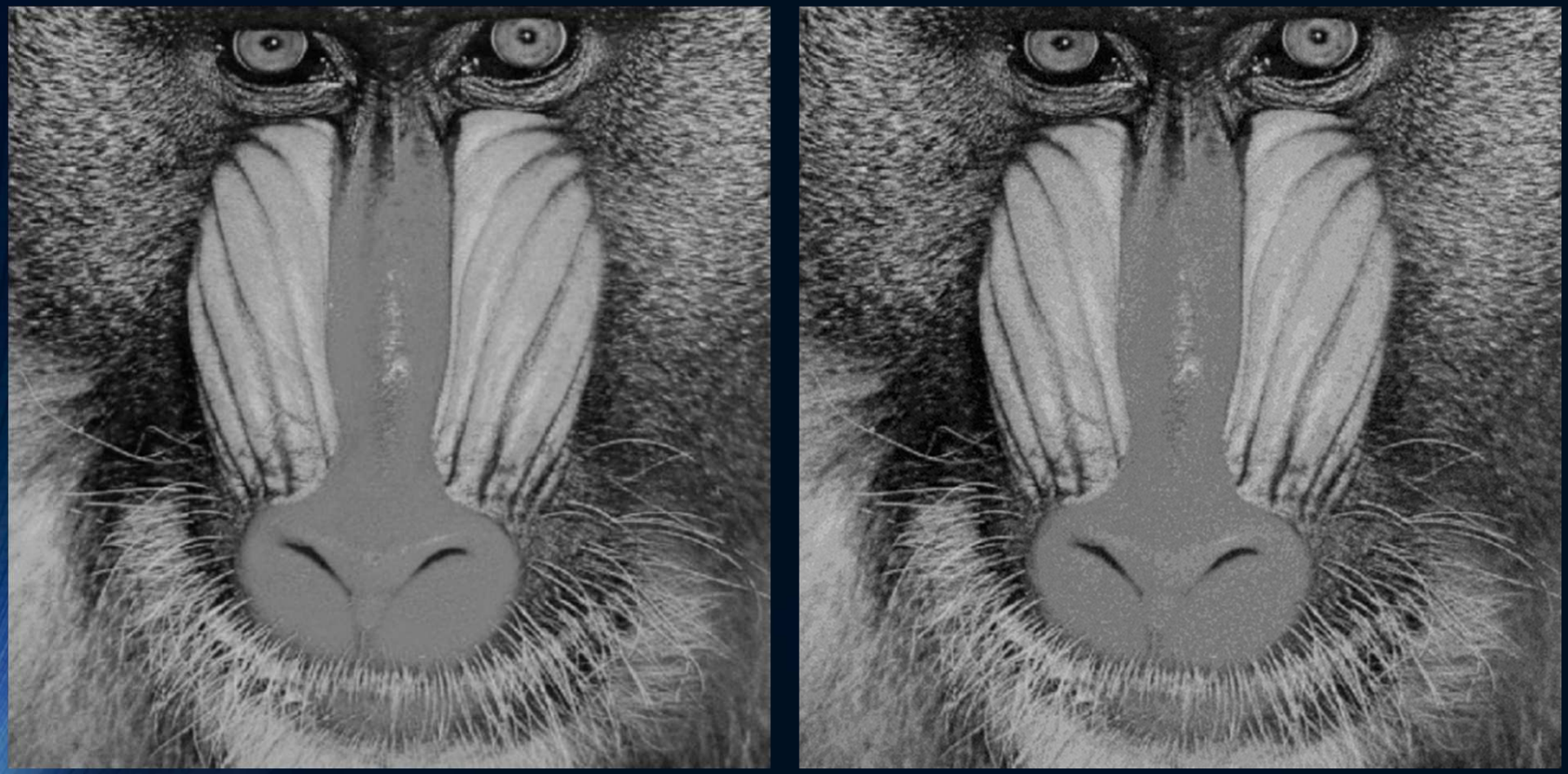
- If an image has fewer than 128 colors, each palette entry could be duplicated
- Message bits are encoded such that
  - m=0 uses the original palette entry
  - m=1 uses the duplicate
  - visual perception will not be altered since the exact same color is displayed regardless of the value of the message bit
- Easily detected by observing duplicate colors in palette
- Could make some colors only close rather than a duplicate
  - harder to detect
  - visual perception possible if colors not close enough
- Could hide in the LSB of the palette colors
  - Very limited capacity
  - In 16-bit color bitmap, could have up to 64k

Paletted Image Hiding - Security

- Particular implementations may leave a "palette signature"
  - palette in a particular order or one having duplicate entries
  - randomized palette order may draw suspicion too
- Perceptibility can increase if color of stego-bit extremely mismatched (using the LSB technique)

Perceptibility – Paletted Images

- Grayscale images are fine cover files (4 bits)



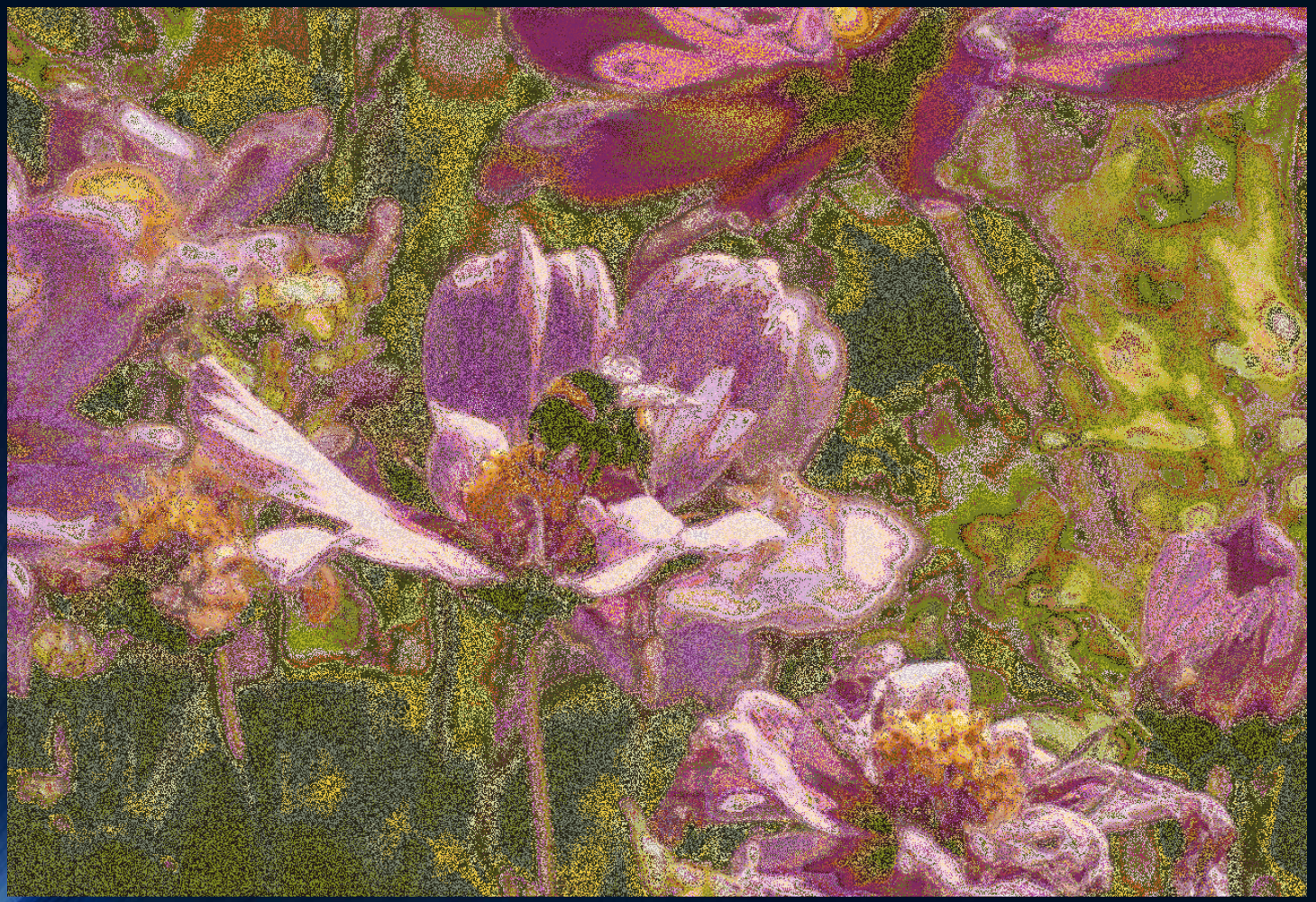
- What is the problem with 8-bit paletted color images?

Perceptibility – RGB Image

- Picture #1: Can you see a difference (1 bit)

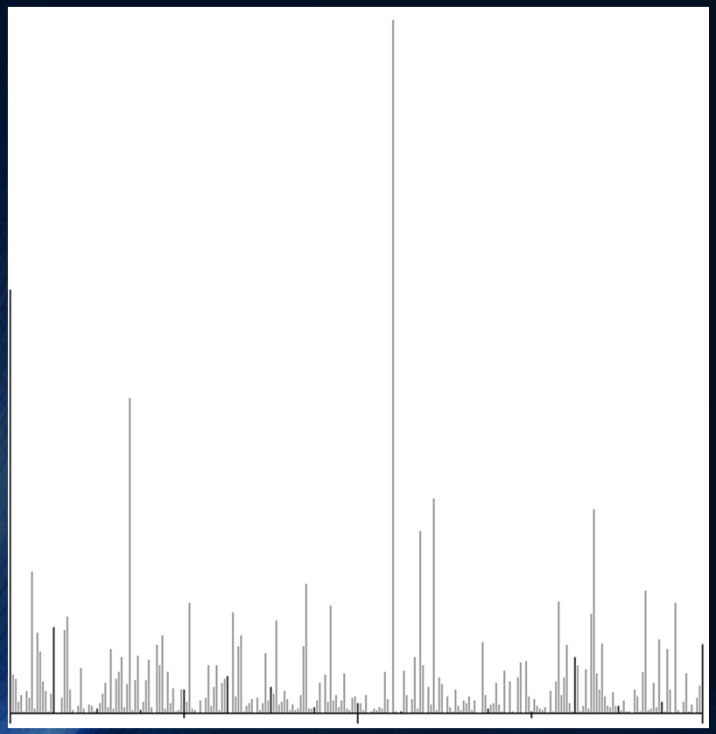
Perceptibility – Paletted Image

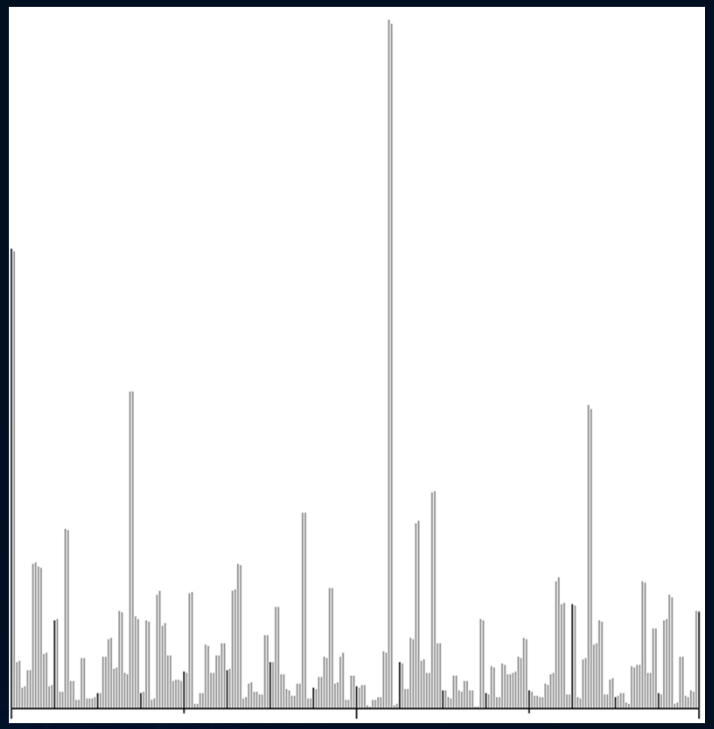
- Picture #2: Can you see a difference (1 bit)



Histograms are Still Effective

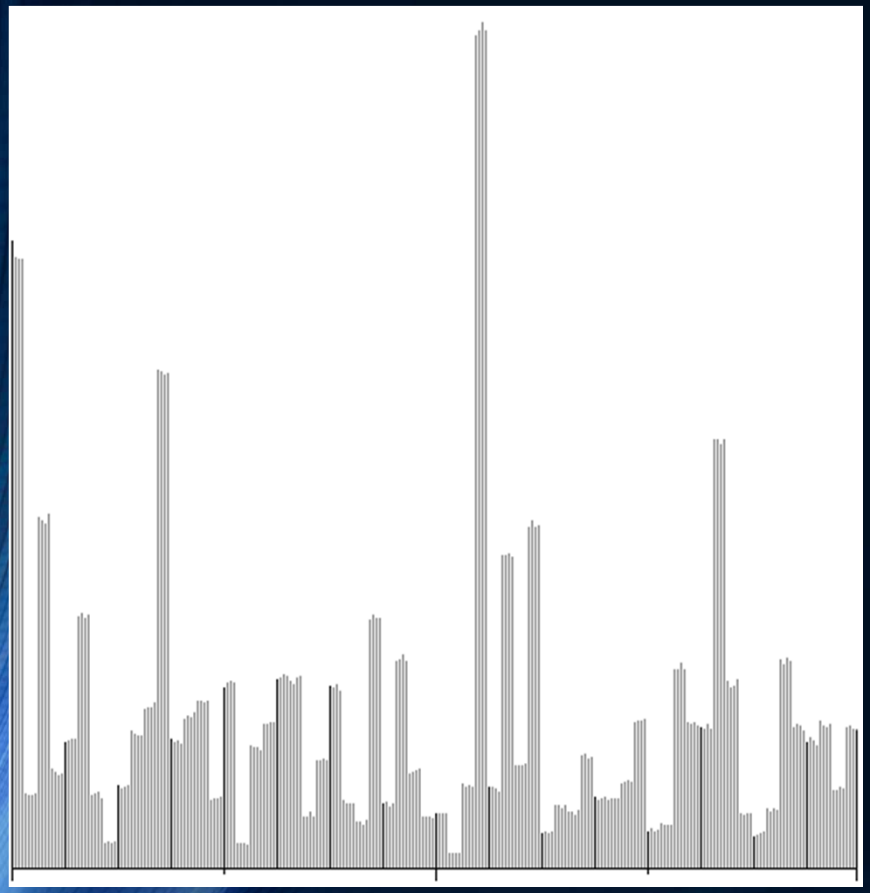
- Note the pairings of values in the histogram with the hidden bits
  - Left one has no data, right one has 1 bit in the indices
  - The histogram is scaled, so they are the same height

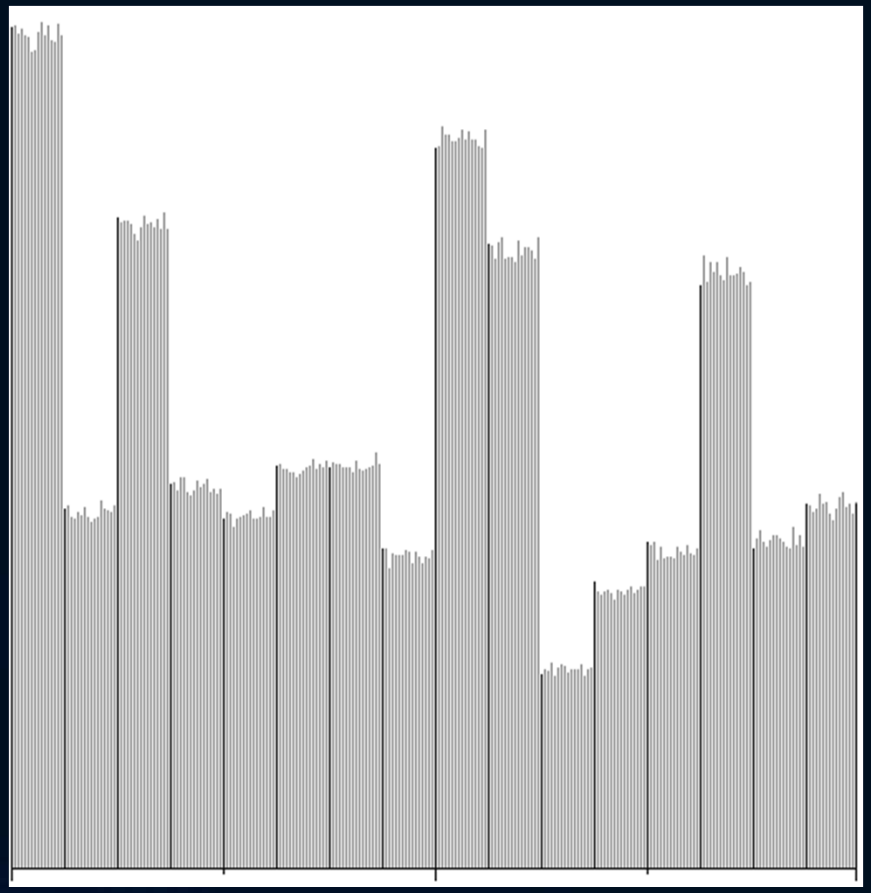




Histograms are Still Effective

- Hiding more bits exacerbates this effect, 2 & 4 shown





Palette-Based Images – A Technique

- A More Advanced technique is presented in the paper "A New Steganographic Method for Palette-Based Images" by Dr. Jiri Fridrich
  - Dr. Fridrich has many publications in this field and it is her specialty
  - check out the website: http://dde.binghamton.edu/
- This technique addresses some of the previous issues
- The secret message M is a stream of bits, m_1, m_2, …m_n of length n
- A user chosen seeds a PRNG to randomly select N pixels in the image

Palette-Based Images – A Technique

- For each pixel, the set of closest colors is calculated using the formula:

$$\text{distance} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

- RGB of pixel is compared to RGB of other palette entries
- Starting with the original pixel color, ask the question: Does the parity match the message bit to be encoded?
  - parity = R+G+B mod 2, or simply add RGB and take the LSB

Palette-Based Images – A Technique

- If no, go to the next closest color, ask the question, repeat
- If yes, change the index of the pixel to match this color
  - The palette is not reordered
  - The problem of similar luminance, different colors, is avoided

Palette-Based Images – A Technique

- Recovery is done by using the PRNG and appropriate key to select the correct pixels in the correct order
- The parity of the RGB color is the corresponding message bit
- Security
  - image colors altered slightly
  - difficult to detect since existing palette is used
  - difficult to extract unless seed is known
- Capacity
  - Message length must be <= cover length
- Robustness
  - If algorithm known, could alter an LSB of one color for each palette entry, so not very robust

Palette Based Images - Capacity

- Capacity is less because there are fewer bytes per pixel
- Generally, can't change the color value too much or the effect is noticeable
  - Translates to fewer bits per pixel
- With duplicate colors in the palette, it's one bit per pixel
  - Can be more if fewer colors – 64 colors be duplicated 4 times --- 2 bits/pixel

Palette Based Images - Robustness

- Messages can be easily destroyed by an attacker since she could resort the palette
- Often, graphics software will reorder the palette in its own way when an image is saved

Hiding in the Palette Itself

- Capacity is very limited
- Palettes are typically no larger than 256 3-byte entries
  - Are often smaller
  - 16-bit color may have more entries
- For a paletted bitmap, there are 4 bytes per palette entry, but if the 4th byte is other than zero, it's suspicious

Questions or Comments