Steganography: Hiding Data in Plain Sight

Dan Johnson

Stanford Splash

December 1, 2018

About me

- Grew up in Palatine, IL
- Fremd High School
- Studied Computer Science and Physics at the University of Illinois at Urbana-Champaign
- PhD Student at the Institute for Computational and Mathematical Engineering at Stanford University

Introduction to Steganography

- Steganography is the practice of concealing information within other non-secret information.
- Different than Cryptography. Cryptography is the art of writing and solving codes. Altering information to an equivalent form that it is hard to make sense of without proper key.
- Steganography conceals the message and doesn't attract attention.
- Exercise: What are some ways you would be able to use Steganography?

Binary

- All data on a computer is represented by a sequence of 0s and 1s
- Compare base 10 and base 2 numbers
- Base 10

10^3	10^2	10^1	10^0
6	7	3	4

$(6734)_{10} = 6 * 10^3 + 700 * 10^2 + 30 * 10^1 + 4 * 10^0$

- Base 2 (binary)

2^3	2^2	2^1	2^0
1	0	0	1

$(1001)_2 = 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 1 = 9$

Binary - Text Representation

- Text is translated using ASCII
- Example: dan johnson = 01100100 01100001 01101110 00100000 01101010 01101111 01101000 01101110 01110011 01101111 01101110

Binary - Text Representation

Exercise: write a short secret message (five letters or less) in binary, trade with someone next to you, and have them decipher it

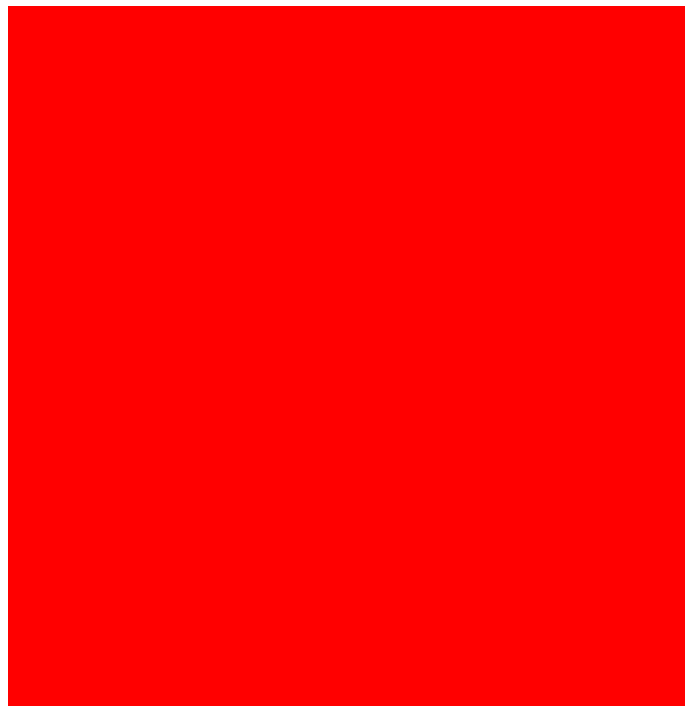
a	097	01100001	n	110	01101110
b	098	01100010	o	111	01101111
c	099	01100011	p	112	01110000
d	100	01100100	q	113	01110001
e	101	01100101	r	114	01110010
f	102	01100110	s	115	01110011
g	103	01100111	t	116	01110100
h	104	01101000	u	117	01110101
i	105	01101001	v	118	01110110
j	106	01101010	w	119	01110111
k	107	01101011	x	120	01111000
l	108	01101100	y	121	01111001
m	109	01101101	z	122	01111010

Binary - Image Representation

- Images are made up by a bunch of pixels in a rectangular array
- Each pixel has three numbers associated with it: a red number, a green number, and a blue number
- Each number is between 0 and 255
- The larger the number is, the more red, green, or blue is in the pixel
- Each of these numbers is represented in binary
- Exercise: how many binary places do we need to represent the numbers 0 to 255

Binary - Image Representation

Example:
R: $255 = 11111111_2$
G: $0 = 00000000_2$
B: $0 = 00000000_2$

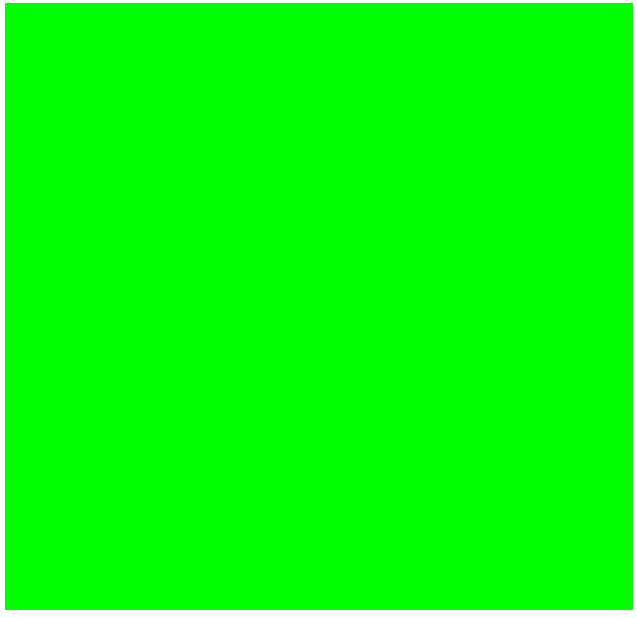


Binary - Image Representation

- Exercise: What will this color be?
  R: $0 = 00000000_2$
  G: $255 = 11111111_2$
  B: $0 = 00000000_2$

Binary - Image Representation

- Exercise: What will this color be?
  R: $0 = 00000000_2$
  G: $255 = 11111111_2$
  B: $0 = 00000000_2$



Binary - Image Representation

- Exercise: What will this color be?
  R: $255 = 11111111_2$
  G: $255 = 11111111_2$
  B: $0 = 00000000_2$

Binary - Image Representation

- Exercise: What will this color be?
  R: $255 = 11111111_2$
  G: $255 = 11111111_2$
  B: $0 = 00000000_2$
-

Binary - Image Representation

- Exercise: What will this color be?
  R: $0 = 00000000_2$
  G: $0 = 00000000_2$
  B: $0 = 00000000_2$

Binary - Image Representation

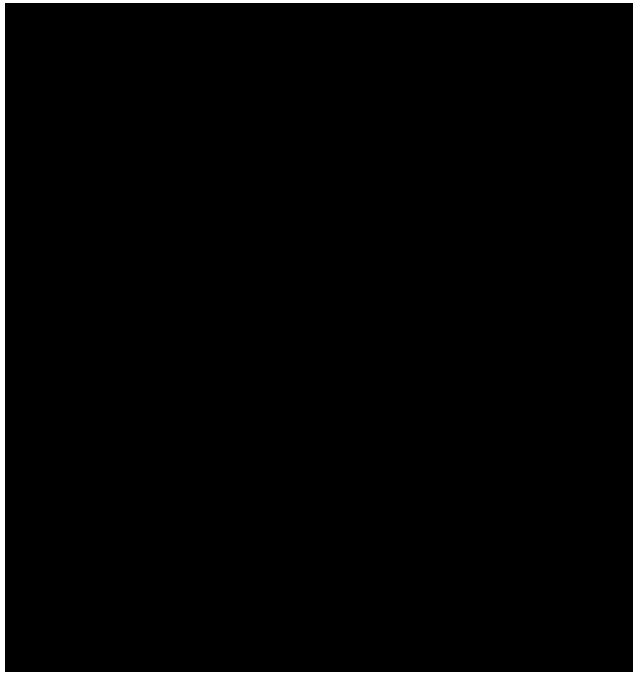
- Exercise: What will this color be?
  R: $0 = 00000000_2$
  G: $0 = 00000000_2$
  B: $0 = 00000000_2$
- 

Image Steganography

- Now we combine the two representations to hide the message in the image
- Idea: take each bit in the message and store it in the least significant bit of each subsequent RGB value

Encoder

Go to https://repl.it/@DanielJohnson6/splash2018

Encoder - Backup Plan

- Open up repl.it/languages/python3
- Copy/Paste code from github.com/dsjohns2/splash2018/blob/master/main.py
- Copy/Paste code from github.com/dsjohns2/splash2018/blob/master/encode.py
- Copy/Paste code from github.com/dsjohns2/splash2018/blob/master/decode.py
- Upload file: raw.githubusercontent.com/dsjohns2/splash2018/master/in.jpg

Exercise

Help me finish encode.py

Exercise

Finish decode.py