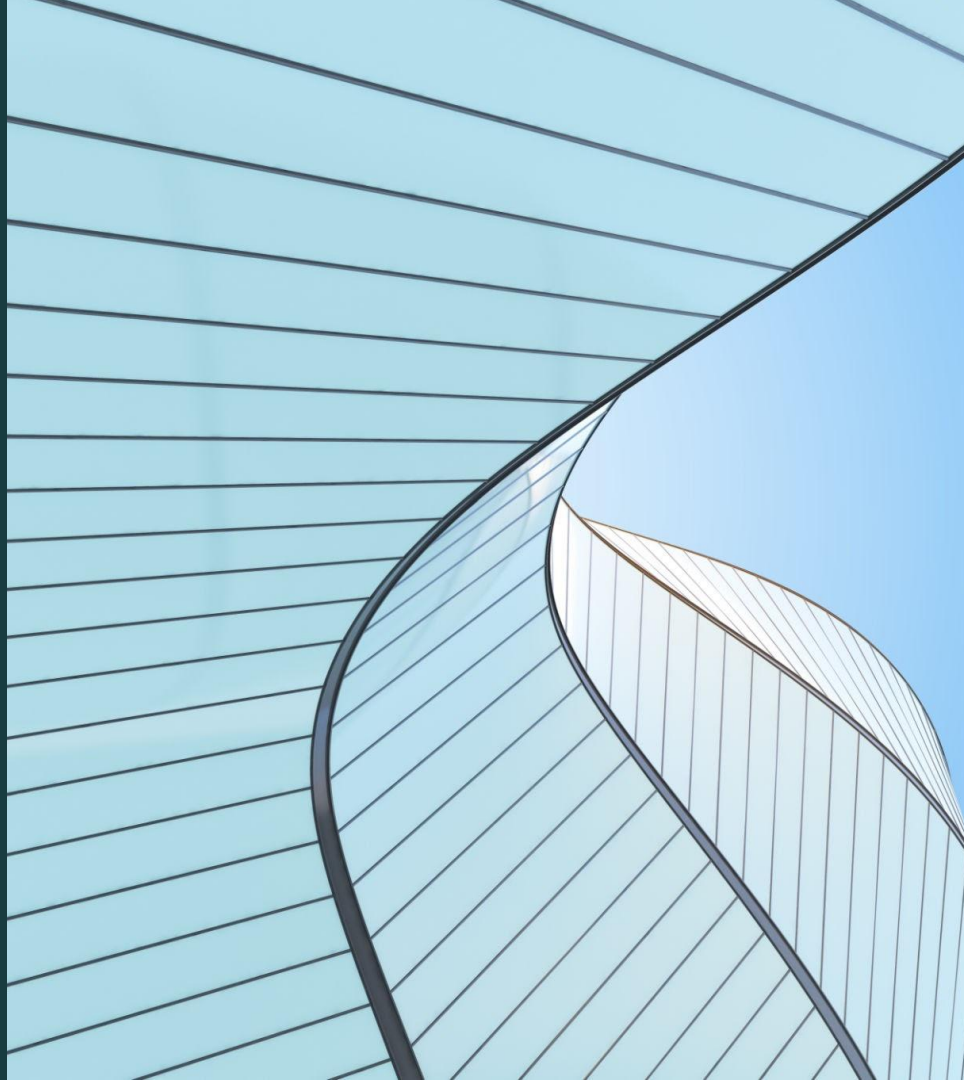
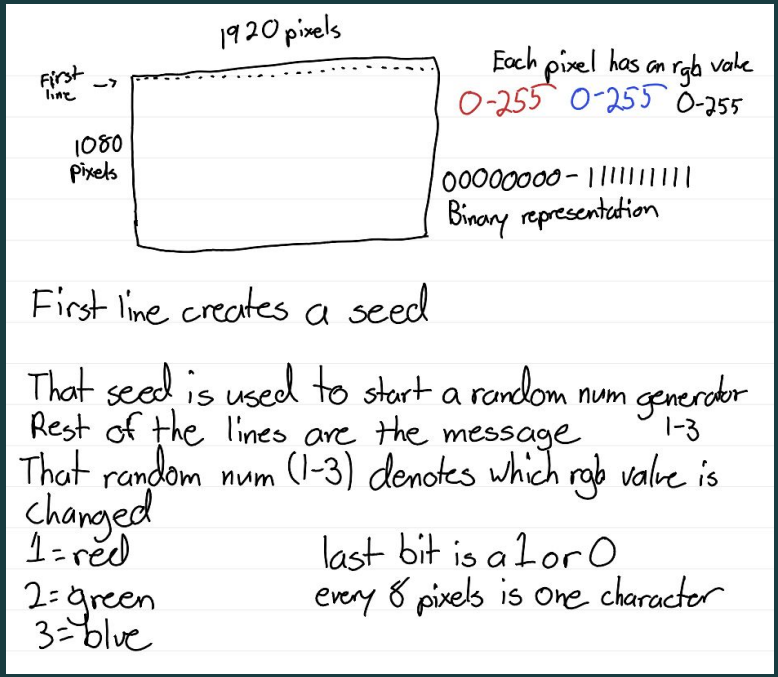
Image Steganography Using LSB

Group 6:
Danilo Sosa, Stacey Zheng, Roberto Reade, Andrew Ioanoviciu



Covert Channel Overview

1. First row generates seed for PRNG (**P**seudo **R**andom **N**umber **G**enerator)
2. PRNG generates 1 to 3 for each pixel's RGB (**R**ed **G**reen **B**lue) value
3. LSB (**L**east **S**ignificant **B**it) for each RGB is changed or read
4. Remaining LSB randomly changed for obfuscation



Related Works

Concepts Learned

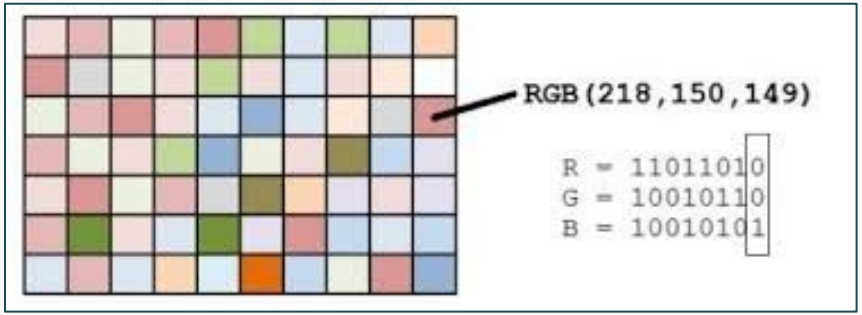
- PRNG (**P**seudo **R**andom **N**umber **G**enerator)
- LSB (**L**east **S**ignificant **B**it) substitution
- RGB (**R**ed **G**reen **B**lue) channels

Idea Origination

- Combining image stego and obfuscation

Topic Understanding Contribution

- Novelty of a channel-only randomization and seed row
- Additional obfuscation of remaining LSBs



Encoder Implementation

- Retrieves PNG image to be used
- Creates a seed and embeds it into the first row of pixels in the red channel of the image
- Uses the seed to initialize a PRNG that gives out a string of digits 1-3
- Prompts user for a message, converts it to binary, and appends an exit string
- Inserts each bit into the LSB of a pixel's corresponding RGB channel according to the PRNG until after the exit string is reached

Original

Decoder Implementation

Encoded

- Retrieves encoded PNG image to be used
- Retrieves the encoded seed from the first row in the red channel of the image
- Uses the seed to initialize a PRNG that gives out a string of digits 1-3
- Extract the bits from the LSB of the pixel's corresponding RGB channel according to the PRNG until the exit code is reached
- Convert the binary to ascii characters and display the message to the user

Metrics Implementation

- Compares the original and encoded image
- Uses scikit-image to calculate the Peak Signal-to-Noise Ratio (PSNR) and Structural similarity index measure (SSIM)
    - PSNR - Used to measure the noise between two images
    - SSIM - Used for measuring the perceived similarity between two images
- Outputs the results to the user

scikit-image logo

Channel Capacity

- Depends on the size of the image
- Can be as small or as large as the sender wants
- Recommended minimum for the image 512x512 pixels
    - Can be as low as two pixels tall and one pixel wide
- First row of pixels is reserved for the seed
- 8 pixels are required per character
- Equation for image capacity:

W: Width of image in pixels

H: Height of image in pixels

$$\frac{(W * H) - W}{8}$$

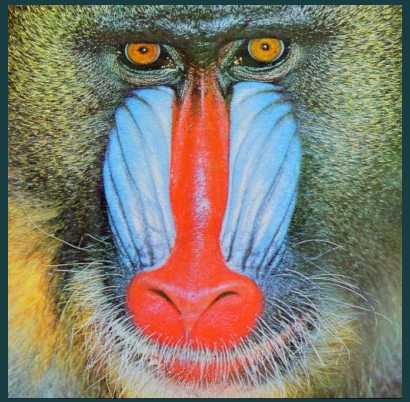
Results Overview

- Even large payloads (up to 784,080 characters) do not introduce visible distortion in the encoded images
- Across all test images PSNR remained above 54 dB
- SSIM values consistently stayed between 0.9985–0.9996
- Results were consistent across different resolutions, and color distributions, demonstrating strong generalization

Comparison to Prior Work

- Rahman et al.’s work served as our baseline for comparison
- The same USC-SIPI images and experimental setup were used
- Our results were added to their comparison table to evaluate our method against all prior techniques
- Classic-LSB, PIT, Karim, and Muhammad

Dataset for Comparison



Baboon

House 1

House 2

Comparison Results

- All methods were evaluated using an 8 KB payload

Image	LSB	PIT	Karim	Muh.	Rahman	Ours
Baboon - 512	61.88	50.19	50.05	59.42	62.85	61.93
Baboon - 1024	67.83	50.20	50.17	65.50	68.96	67.91
House 1 - 512	48.87	50.11	51.56	59.30	74.91	61.60
House 1 - 1024	51.90	50.43	52.45	65.32	75.05	67.63
House 2 - 512	62.74	54.75	54.37	63.34	64.65	61.90
House 2 - 1024	68.83	54.79	54.69	72.47	72.57	67.93

Image	Rahman	Ours
Baboon - 512	0.9999	1.0000
Baboon - 1024	1.0000	1.0000
House 1 - 512	0.9999	0.9994
House 1 - 1024	1.0000	0.9998

Application and Uses

- Hide messages in unique images
- Tested via Email
  - Message recovered successfully
  - Image remained visually identical
- Potential Platforms:
  - Messaging apps
  - Social media
  - Gaming
  - Public forums

Limitations and Future Works

Limitations

- JPEG issues
  - Lossy compression destroys hidden bits
  - Must use PNG which limits platform capability
- Many apps automatically compress or resize images

Future Works

- Test across multiple platforms
  - Social media
  - Messaging apps
  - Public forums
- Strengthen against steganalysis tools
- Develop compression resistant encoding
- Explore covert broadcasts
  - Publicly shared image with secret message

Thank you!