In the Simpleviewer XML file we entered metadata for the class picture
- Added the picture file to images
- Added tiny picture file to thumbs
The Result

Faces
Computing Is Pretty Strange

Steganography: An Amazing Thing To Do with Bits

Lawrence Snyder
University of Washington, Seattle
The process of hiding information

Two Greek roots meaning:
“stego” == “roof”  “stega” == “cover”
Why Hide Information?

- Most common reason to hide information is to avoid being caught with it
  - Military and spy documents
  - Repressive governments restricting news/info
  - Avoid others “snooping” – privacy

- Hiding is different than encryption ... uses the fact that the searcher doesn’t know it’s there
Illustrate A Way To Do It

- The Plan ...
  - hide “subversive” protest picture in “calendar art”
Recall Properties of Pixels

- Pixels are made of RGB color
- Light intensity (brightness) of the 3 lights is given by RGB bytes:
  - R: 1111 1111
  - G: 1111 1111
  - B: 0000 0000
- We can manipulate the bits and change the picture
A handy fact about binary numbers is that a sequence of bits can be shifted right or left by dividing or multiplying by a power of 2.

- Dividing by $2^n$ shifts right $n$ bit positions
- Multiplying by $2^n$ shifts left $n$ bit positions

\[
\begin{align*}
11010110101110 & \div 2^2 = 00110101101011 \\
11010110101110 & \times 2^2 = 01011010111000
\end{align*}
\]
Step 1: Reduce Bits of Guest

- We don’t need all of the bits in RGB to get a decent picture
Check Removing Bits

Plmage baselm, fewerBits;
int i = 0;
int wid=450;
int hi=300;
color c;
int fact=1;
```java
void setup() {
    size(wid, hi);
    baselm = loadImage("egypt.jpg");
}

void draw() {
    image(baselm, 0, 0);
    loadPixels();
    for (int i = 0; i<wid*hi; i++){
        c = color( fact*(int(red(pixels[i])/fact)),
                   fact*(int(green(pixels[i])/fact)),
                   fact*(int(blue(pixels[i])/fact)));
        pixels[i] = c;
    }
    updatePixels();
}

void mousePressed() {
    fact = 2 * fact;
}
```
void setup() {
    size(wid, hi);
    baselm = loadImage("egypt.jpg");
}

void draw() {
    image(baselm, 0, 0);
    loadPixels();
    for (int i = 0; i < wid*hi; i++){
        c = new color(red(pixels[i])/fact),
                   green(pixels[i])/fact),
                   blue(pixels[i])/fact));
        pixels[i] = c;
    }
    updatePixels();
}

void mousePressed() {
    fact = 2 * fact;
}
Step 2: Replace Bits In Host

- Put guest bits into right 2 bits of host
null

Processing Code For Guest $\rightarrow$ Host

```cpp
Plimage crowd, fog;
int i = 0;
int srcw=512;
int srch=346;
int wid=450;
int hi=300;
color c, cprime;

void setup( ) {
  size(srcw, srch);
  crowd = loadImage("egypt.jpg");
  fog = loadImage("fog.jpg");
  image(fog,0,0);
  for (int i=0; i<srcw; i++){
    for(int j=0; j<srch; j++) {
      c = get(i,j);
      if (i<wid && j<hi) {
        cprime=crowd.get(i,j);
        cprime=color(4*(int(red(c))/4) + (int(red(cprime))/64),
                   4*(int(green(c))/4) + (int(green(cprime))/64),
                   4*(int(blue(c))/4) + (int(blue(cprime))/64));
        set(i,j, cprime);
      } else {
        set(i,j,c);
      }
    }
  }
}

void draw( ) {
  if (mousePressed) {
    saveFrame("stegFog.png");
  }
}
```

Code To Save Result on Click

Encoding Code
Compare fog.jpg with stegFog.png

Really?
Just Do It!

fog.jpg

stegFog.png
How Does It Work

- After the pictures are loaded

```c
  cprime = color(4*(int(red(c))/4) + (int(red(cprime))/64),
                 4*(int(green(c))/4) + (int(green(cprime))/64),
                 4*(int(blue(c))/4) + (int(blue(cprime))/64));
```

- Clear right 2 bits of host
- Extract left 2 bits of guest
- New combined color
Recover The Image

```c
Pimage fog;
int flip = 0;
int srcw=512;
int srch=346;
int wid=450;
int hi=300;
color c, cprime;

void setup( ) {
    size(srcw, srch);
    fog = loadImage("stegFog.png");
    image(fog,0,0);
}

void draw( ) {
    if (mousePressed) {
        for (int i=0; i<srcw; i++){
            for(int j=0; j<srch; j++) {
                c = get(i,j);
                if (i<wid && j<hi) {
                    cprime=color(64*(int(red(c))%4),
                              64*(int(green(c))%4),
                              64*(int(blue(c))%4));
                    set(i,j, cprime);
                } else {
                    set(i,j,c);
                }
        }
    }
}
```
Just Do It!
Read in the file, and then on mouse click, pull out the bits and make a picture

```
cprime=color(64*(int(red(c))%4),
       64*(int(green(c))%4),
       64*(int(blue(c))%4));
```

Remove right 2 bits

Make them left 2 bits for each color

New color
How Much Is Coded Like Original?

- Run A Test ... www.tineye.com

5 Results

Searched over 1,8825 billion images in 0.013 seconds.

for file: fog.jpg

These results expire in 72 hours. Why?

Share a success story!

TinEye is free to use for non-commercial purposes.

Download the official TinEye extension for Firefox with right-click functionality!

Sort Order

- Best Match
- Most Changed
- Biggest Image

www.milliyet.com.tr
2.jpg
http://www.milliyet.com.tr/content/galeri/yeni/...

forum.shiftdelete.net
2.jpg
http://forum.shiftdelete.net/sdn-magazin/gunun->...
Check The “Steganized” File

5 Results
Searched over 1,8825 billion images in 2.609 seconds.
for file: stegFog.png
These results expire in 72 hours. Why?
Share a success story!
TinEye is free to use for non-commercial purposes.
Download the official TinEye extension for Firefox with right-click functionality!

Sort Order
Best Match
Most Changed
Biggest Image

www.milliyet.com.tr
2.jpg
http://www.milliyet.com.tr/content/galeri/yenil/

forum.shiftdelete.net
2.jpg
http://forum.shiftdelete.net/sdn-magazin/gunun-...
Summary

- Put guest bits into right 2 bits of host

Watch It:
Push the bits out the left side, slowly revealing the guest
Summary

- Put guest bits into right 2 bits of host

Watch It:
Push the bits out the left side, slowly revealing the guest