Image Steganography

Dr Ian Cornelius

Hello

Hello (1)

Learning Outcomes

1. Understand the concept of image steganography
2. Demonstrate knowledge on how to use image steganography in a body of work

Steganography

Steganography (1)

- Steganography is concerned with the study and practice of concealing information in objects
- It is done in a manner that it deceives the viewer
  - the viewer is under the impression there is no hidden content in the object
- Essentially, the information is hidden in plain sight
  - only the intended recipient will be able to view it

Steganography (2)

Is this Cryptography?

- No.
- Cryptography is concerned with modifying a string to make it difficult to get the original string
  - you are unable to get the original string back from the modified version
- Both the original and modified string look completely different to one another
  - i.e. `abcd` -> `1@z*`

Image Steganography

Image Steganography (1)

- A technique concerned with hiding data inside an image
- Done in a manner that prevents an unintended user from detecting the hidden message
- The following elements are required:
  - a **cover image**: an image that will hold the message
  - the **message**: the message to be sent, it can be plain or encrypted text or even an image
  - a **key**: the key is used to hide the message, it is *optional*

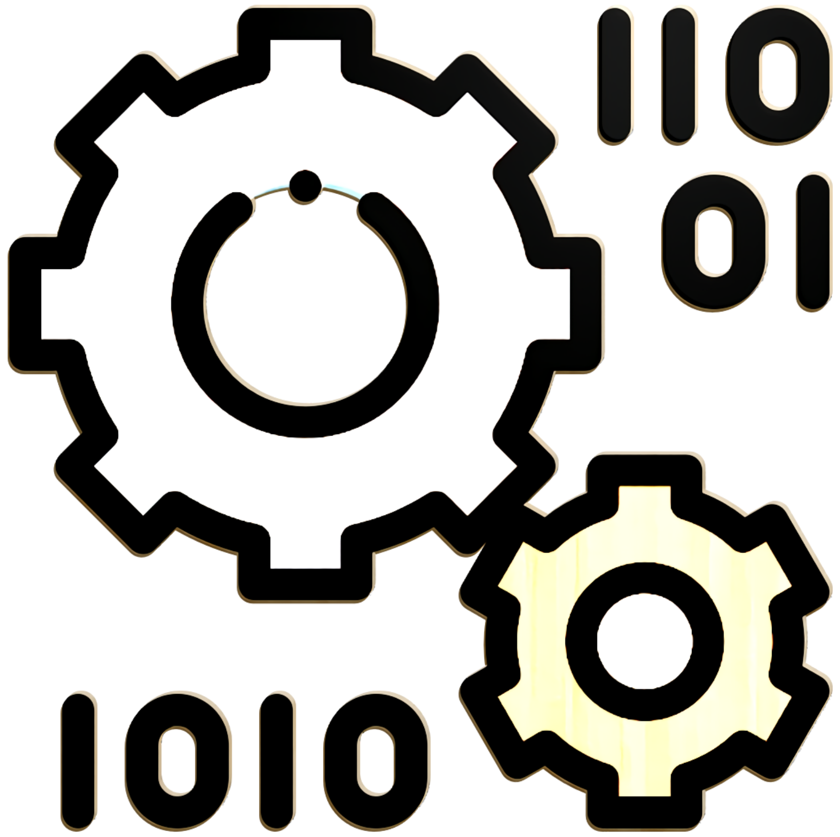

Image Steganography (2)

Applications of Image Steganography

- Image steganography is useful for a multitude of things:
  - securing private files
  - transmitting messages or data with revealing the existence of a message
  - hiding passwords or encryption keys
  - transporting sensitive documents between users

Image Steganography (3)

Types of Steganography Techniques

- Several types and forms of steganography:
  - **physical**: does not require the use of digital mediums or files; this type includes:
    - passing messages written with invisible ink which is read by the recipient by applying certain chemicals
    - use of ciphering techniques to hide the information with textual information, i.e. caesar cipher
  - **microdots**: shrinking messages to tiny dimensions, becoming almost invisible
  - **digital**: involves the use of digital mediums such as image, audio and video files

Image Steganography in Practice

Image Steganography in Practice (1)

- Requires knowing about pixels and colour models
  - re-visit the previous video if you need to revise
- Say we have a pixel with values (0, 0, 255)
  - `red = 0`
  - `green = 0`
  - `blue = 255`
  - therefore, our pixel is blue
- For an 8-bit system, a pixel can accommodate eight digits
  - represented in a binary format
  - the largest number in eight bits is: `11111111`
    - this is equal to `255`
  - the smallest number in eight bits is: `00000000`
    - this is equal to `0`
- Our RGB values in binary are:
  - `binaryRGB = (00000000, 00000000, 11111111)`

Image Steganography in Practice (2)

- Consider the following table, representing 3 pixels
- Each pixel is a particular RGB value associated to it in eight-bit form

Pixel	RGB	R	G	B	Image Color
1	(45, 28, 220)	00101101	00011100	11011100	
2	(166, 196, 12)	10100110	11000100	00001100	
3	(31, 86, 94)	00011111	01010110	01011110	

Image Steganography in Practice (3)

- How can we hide the number `169` into it?
- First, we need to convert the decimal number to binary: `10101001`
- Each digit of the binary number is then used to replace the least significant bit (LSB) from our pixels
  - shown in the bold and red below

Pixel	RGB in Decimal	R	G	B	Image Color
1	(45, 28, **221**)	0010110**1**	0001110**0**	1101110**1**	
2	(166, **197**, 12)	1010011**0**	1100010**1**	0000110**0**	
3	(**30**, **87**, 94)	0001111**0**	0101011**1**	01011110	

- Some of our RGB values in decimal have also changed, due to changing the least significant bit
  - shown in bold and blue

Image Steganography in Practice (4)

Before

Pixel	RGB in Decimal	R	G	B	Image Color
1	(45, 28, 220)	00101101	00011100	11011100	
2	(166, 196, 12)	10100110	11000100	00001100	
3	(31, 86, 94)	00011111	01010110	01011110	

After

Pixel	RGB in Decimal	R	G	B	Image Color
1	(45, 28, **221**)	0010110**1**	0001110**0**	1101110**1**	
2	(166, **197**, 12)	1010011**0**	1100010**1**	0000110**0**	
3	(**30**, **87**, 94)	0001111**0**	0101011**1**	01011110	

Image Steganography in Practice (5)

- The process we have just gone through is known as the Least Significant Bit (LSB)
  - a common method that is often used for image steganography
- Takes into account the pixel information of an image
- Works best when the image file is larger than that of the message

LSB Algorithm Steps

- **Step 1**. Select a cover image and choose a message to hide
- **Step 2**. Find the pixels of the cover image
  - **2a**. Extract the RGB values of the first pixel
  - **2b**. Convert each value to its binary equivalent
- **Step 3**. Extract the first character of the message
  - **3a**. Convert the character to its binary value
- **Step 4**. Hide each digit of the *characters* binary value into the last bit of the RGB binary value
  - **4a**. Move onto the next pixel if required
- Repeat **Step 3 to 4a** as necessary until all characters of the message are completed.

Goodbye

Goodbye (1)

Questions and Support

- Questions? Post them on the **Community Page** on Aula
- Additional Support? Visit the Module Support Page
- Contact Details:
    - Dr Ian Cornelius, ab6459@coventry.ac.uk