Introduction to Steganography

CSRI Masterclass

jack.sturgess@cs.ox.ac.uk

1

Plan

- Theory
  - What is cryptography?
  - What is steganography?
  - Cryptography vs. steganography
  - Examples of steganography
  - How an image-based stegosystem works
- Break
- Practical
  - Create an image-based stegosystem

2

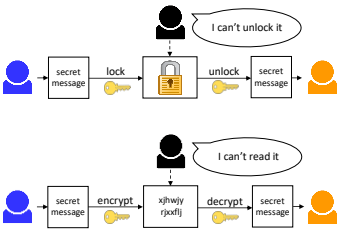
What Is Cryptography?

- Literally, 'hidden-writing', from the Greek words *kryptos* (hidden) and *graphein* (writing)
- The practice of...
  - encrypting messages to hide their contents
  - sending messages without your adversary being able to read the messages

3

What Is Cryptography?

- Analogous to sending a message in a locked container
  - Requires a key to lock and unlock



4

What Is Cryptography?

- Caesar cipher
  - Simple substitution cipher

$$c_i = p_i + k$$

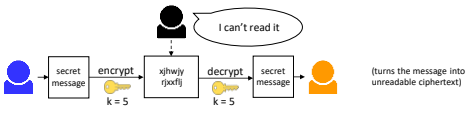
Key = 5

Plaintext: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Ciphertext: f g h i j k l m n o p q r s t u v w x y z a b c d e

Plaintext: S E C R E T M E S S A G E

Ciphertext: x j h w j y r j x x f l j



5

What Is Cryptography?

- Cryptography is widely used
  - Accessing webpages, smart devices
  - Logging into games, online services
  - Sending and receiving messages, emails, voice-over-IP
  - Making non-cash payments (cards, apps, crypto-currencies)

Codewheel

Enigma machine

One-time pad

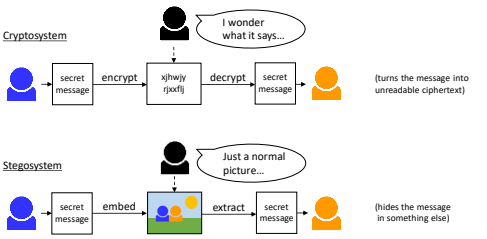
6

What Is Steganography?

- Literally, 'cover-writing', from the Greek words *steganos* (covered, concealed) and *graphein* (writing)
- The practice of...
  - concealing messages under the cover of something else, such as an innocuous message, image, sound, or video
  - sending messages without your adversary knowing that you're sending messages
  - sending messages with double meanings: one overt and one covert

Cryptography vs. Steganography

- Cryptography hides the **content** of a message
- Steganography hides the **existence** of a message



Examples of Steganography

Invisible ink

Body art

Prison Break protagonist had information tattooed on his body to aid his escape

Moon cakes

Chinese folklore says that messages smuggled in moon cakes helped overthrow the Mongols

Inspector Sands, please report to room 7B.

Code words

Translation: there is a fire in room 7B

Examples of Steganography

Null cipher

All of the other words are just cover

Dear George,
Greetings to all at Oxford. Many thanks for **your** letter and for the summer examination **package**. All entry forms and fees forms should be **ready** for final dispatch to the syndicate by **Friday** 20th or at the latest, I am told, by the **21st**. Admin has improved here, though there is **room** for improvement still; just give us all two or **three** more years and we will really show you! **Please** don't let these wretched 16+ proposals **destroy** your basic O and A pattern. Certainly **this** sort of change, if implemented **immediately**, would bring chaos.
Sincerely yours, Edward

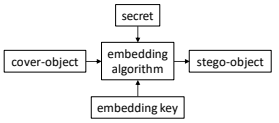
Spot the Difference!

Examples of Steganography

- An example of image steganography
  - This 2.48 MB bitmap contains a 271 KB file
- Images contain lots of **redundancy**
  - Lots of unnecessary bits
  - Lots of capacity to hide other data
- Images are widely used in modern steganography

Stegosystems

- A **cover-object** (or cover-text, cover-image, etc.) is an original, unaltered object
- A **secret** may be **embedded** into a cover-object, typically using a **key**, to obtain a **stego-object** (or stego-text, stego-image, etc.)



- The secret can then be **extracted** from the stego-object using the key, if one was used
- The secret may be split over multiple cover-objects using different keys for added security

Stegosystems

- If detected, a stegosystem is broken
  - Current message is lost
  - Future messages will be lost

Invisible ink

How could we reset a detected invisible ink stegosystem?
- Write messages in a different place (new cover)
- Write messages in a different manner (new algorithm)
- Change the ink's chemical composition such that it requires a new triggering agent (new key)

- Sometimes, cryptography and steganography are combined
  - First, encrypt the message to protect it
  - Second, hide it to avoid suspicion in transit
  - Arguably, if you do one correctly, you shouldn't need the other

Steganography Trade-offs

- Steganography techniques face a trade-off between three factors:
  - **Capacity** – how much data can be hidden in the cover-object
  - **Undetectability** – how difficult it is to prove the existence of the hidden data
  - **Robustness** – how much cover-object manipulation the embedded data can endure
    - For example: cropping, compression, filtering, etc.

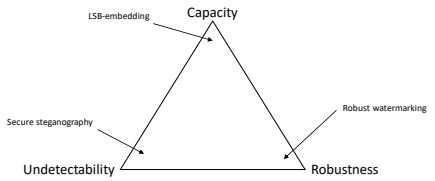


Image-based Stegosystem

- Worked example: an image-based, secret-key stegosystem
  - Sender embeds the secret into a cover-image using a secret key
  - Stego-image is sent to the receiver
  - Receiver extracts the secret using the same key

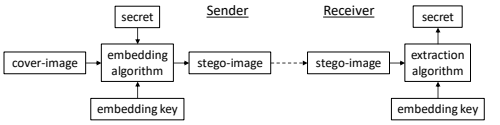
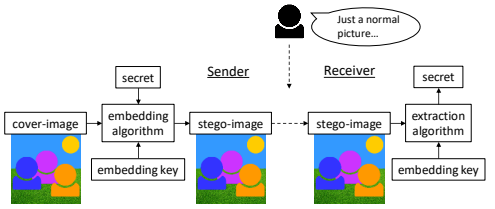


Image-based Stegosystem

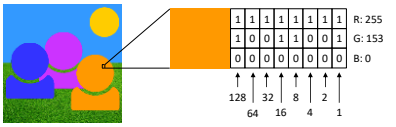
- In order not to arouse suspicion, minimise visible degradation
  - The stego-image should be **practically indistinguishable** from the cover-image



- This is achieved by identifying and exploiting **redundancy** in the cover-image

Image-based Stegosystem

- In a 24-bit RGB image, every pixel is represented by 24 bits
  - 8 bits to show how much **red** it has
  - 8 bits to show how much **green** it has
  - 8 bits to show how much **blue** it has

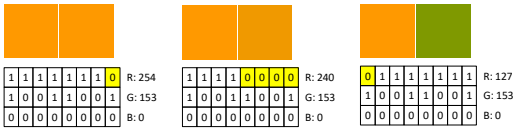


- For example, the amount of green in this pixel is

  1*128 + 0*64 + 0*32 + 1*16 + 1*8 + 0*4 + 0*2 + 1*1 = 153

Image-based Stegosystem

- Each bit contributes a different amount to the total
  - The last bit is the **least significant bit (LSB)**
  - We can modify the last few bits without noticeable difference

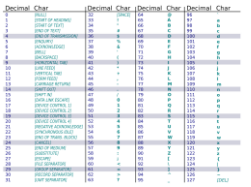


- The last few bits are essentially redundant, as their impact is not visible to the human eye

- We can exploit this redundancy by hiding information in these bits
  - We can modify them and it won't visibly degrade the image
  - Therefore, we can hide information in them

Image-based Stegosystem

- Convert the secret message into bits
  - For text, we can use ASCII encoding
  - Each character is assigned a number between 0 and 127
  - This can be represented by 7 bits
  - For example, T is assigned 84 in ASCII, and 84 is 1010100 in binary
  - So, to embed a T, we embed these 7 bits



- Convert the pixel colour value into bits
  - Choose a colour plane, such as red
  - The pixel's value for that colour will be between 0 and 255
  - This can be represented by 8 bits
  - For example, a purple pixel might have a value of (190, 0, 207)
    - The amount of red is 190
    - 190 is 10111110 in binary
  - So, to embed into this pixel, we modify one of these bits
    - Choose the least significant bit (the 0 at the end)

Image-based Stegosystem

- Embed the secret bits into the pixel bits
  - Overwrite each pixel LSB with a secret bit

- For example, let's embed a T into these pixels:

  - T is 84 in ASCII, which is 1010100 in binary
  - The amount of red in each pixel is shown below in binary
  - Overwrite each pixel LSB with a secret bit

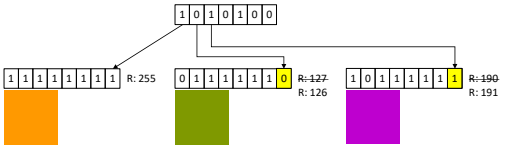


Image-based Stegosystem

- What we have so far:
  - ✓ Secret (converted to bits)
  - ✓ Cover-image (converted to bits)
  - × Embedding key

- Embedding key
  - Embedding data into pixels in their **natural order** is neither undetectable nor robust
    - The adversary can find and extract the secret
    - The adversary can destroy the secret (*e.g.*, by cropping the stego-image)
  - Use the key to **re-order the pixels** before embedding into them
    - Generate a deterministic pseudorandom sequence using the key as a seed
    - Re-order the pixels according to this sequence
  - This spreads the embedded data across the cover-image
    - The intended recipient knows the key and can generate the same sequence to extract the secret; the adversary cannot

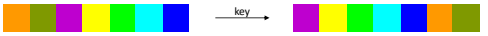
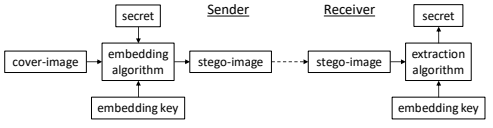


Image-based Stegosystem

- Embedding algorithm:
  1. Use the embedding key to seed a pseudorandom sequence for re-ordering the pixels
  2. Convert the secret message into bits
  3. Cycle through the cover-image pixels in that order, embedding one secret bit into each LSB

- Extraction algorithm:
  1. Use the embedding key to seed the same pseudorandom sequence
  2. Cycle through the stego-image pixels in that order, extracting each LSB
  3. Convert the output bits back into the secret message



Practical

- Task: create an image-based stegosystem
  - Go to the lab
  - Work in groups of 2 or 3
  - Follow the instructions on the worksheet
  - Ask for help when needed