Hide & Seek

An overview of Information Hiding

mano@cs.auckland.ac.nz

4,000,000
1,000,000
40,000
400,000,000


ONE DOLLAR
$1
NEW ZEALAND

Listen to us, we said to the government.

And the government now listens.

But we complain that it does.

Encryption - Example

Pershing sails from NY June 1.

Crefuvat fnvyf sebz AL Whar 1.

Info Hiding - Example

Apparently neutral’s protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

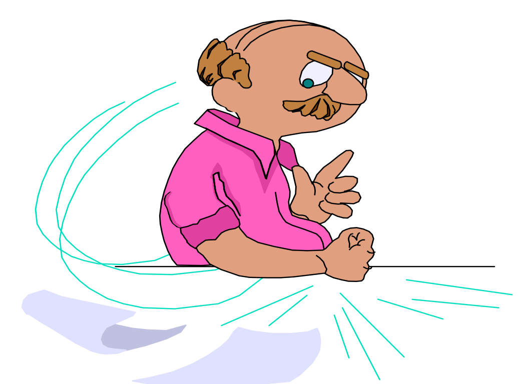
A message sent by a spy in World War II.

The business of hiding

Steganography

Steganography

- The word derives from Greek, and literally means "covered writing"
- While cryptography scrambles messages so that they cannot be understood, steganography hides messages so that they cannot be seen.
- It includes a variety of secret communication methods that conceal the message's very existence.



Steganography - Example

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

A message sent by a spy in World War II.

Steganography - Example

A<u>p</u>parently n<u>e</u>utral's p<u>r</u>otest i<u>s</u> t<u>h</u>oroughly
d<u>i</u>scounted a<u>n</u>d i<u>g</u>nored. I<u>s</u>man h<u>a</u>rd h<u>i</u>t.
B<u>l</u>ockade i<u>s</u>sue a<u>f</u>fects p<u>r</u>etext f<u>o</u>r
e<u>m</u>bargo o<u>n</u> b<u>y</u>-products, e<u>j</u>ecting s<u>u</u>ets
a<u>n</u>d v<u>e</u>getable o<u>i</u>ls.

Pershing sails from NY June 1.

vs. Encryption

Pershing sails from NY June 1.

Crefuvat fnvyf sebz AL Whar 1.

Steganography

- Information can be hidden in a variety of media: images, audio, network packets, etc.
- Changes to cover media after hiding information is not human-noticeable

The business of seeking

Steganalysis

Steganalysis

- Finding if a given media has any hidden information
- A difficult problem in general

Steganalysis

- Most steganalysis schemes attempt to detect if there is hidden information in a given media or not
- A few schemes attempt to detect the size of the hidden information if there is any

Outline

- An example of steganography: LSB steganography
- An example of steganalysis: LSB steganalysis

LSB Steganography

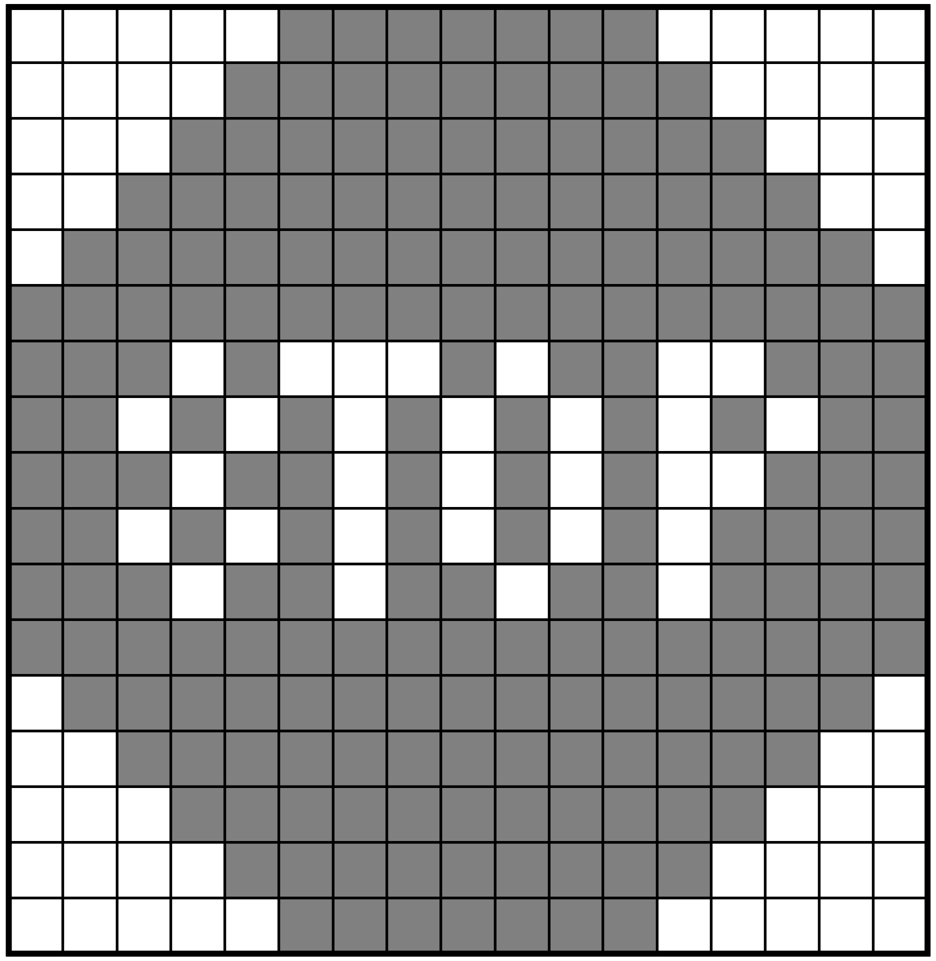
LSB Steganography

- LSB image steganography uses the least significant bits of pixels to represent the hidden message.

Example: A bitmap cover

- Consider an 8-bit grayscale bitmap image
- Each pixel in the bitmap is stored as a byte representing a grayscale value
- Change the last bit of each of the data bytes to reflect the message that needs to be hidden

Example: A bitmap cover



Example: A bitmap cover

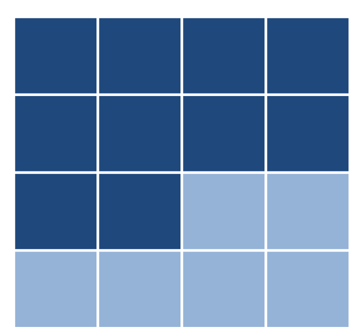
11010010 01001010 10010111 10001100
00010101 01010111 00100110 01000011

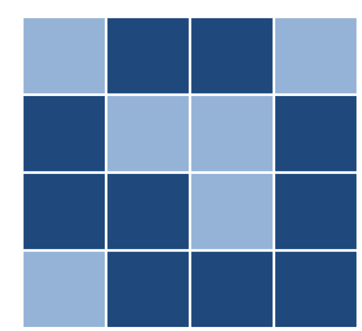
A: 01000001

11010010 0100101**1** 1001011**0** 10001100
0001010**0** 0101011**0** 00100110 01000011

Where to embed

- Note that LSB image steganography uses the least significant bits of pixels to represent the hidden message.
- Two possible ways to pick cover pixels: Sequential, and Random





How to embed

- Two possible ways to alter LSB
    - Replace (LSB replacement)
    - Add/subtract one (LSB matching)

LSB Replacement

- Flip the LSB of the cover pixel as required based on the bit we want to hide.
- Pixel value 10101110 could become 10101111 or stay as it is so as to represent a single bit of the message.

if (c & 0x1 == 0x1) c' = c OR c – 1 *// odd colour value*
if (c & 0x1 == 0x0) c' = c OR c + 1 *// even colour value*

LSB Matching

- Add or subtract 1 to/from the pixel value if the LSB of the cover pixel does not match the bit we want to hide.

- To add or subtract? Choose randomly!

$$c' = c \textbf{ OR } c - 1 \textbf{ OR } c + 1$$

LSB Steganography

- Only about 50% of the chosen cover pixels actually change their values
- The new colour is either the old colour plus one or old colour minus one

These observations are useful for Steganalysis

Media Operations

- LSB steganography is easy to implement, but it is vulnerable to almost all media transformations.
- For example, cropping an image that has a hidden message can result in losing the entire message.

Media Operarations

- Consider a hidden message ABC, which is 01000001 01000010 01000011 in binary.
- Assume that a crop operation on the image file resulted in losing the first two bits.
- In this case, we have lost the character A, but the characters B and C are intact.
- Still, since we do not know about the bit losses, we may incorrectly end up with a wrong grouping of bits: 00000101 00001001 000011 .
- We not only lost A, but also B and C.

Media Operations

- A solution to this is to introduce synchronisation characters in the message stream.
  - Losing bits within two synchronisation markers will mean losing only that part of the message.
- But, there is a considerable overhead in using synchronisation characters.
- Another solution is to use self-synchronising code sets to encode the message.
  - This, one will notice, has little overhead.

Self-Synchronising Codes

- If some bits are lost in a self-synchronising code encoded stream, the decoder will regain synchronisation automatically.
- Self-synchronising codes can be used for compression.

Example: T-Codes

- The construction of T-Codes is done via a recursive copy-and-prepend process called T-augmentation.
- Let us build a T-code set to understand the process.

Example: T-Codes

- A simple T-code set consists of the alphabets. With a binary alphabet, this is $S = \{ 0, 1 \}$.
- We then remove one of the elements of the set and use it as a prefix to extend the initial set so that we get more codes.
- Let us use the first element 0 as the prefix.
- The new code set therefore is $S_{(0)} = \{ 1, \underline{0}0, \underline{0}1 \}$.

Example: T-Codes

- The code set $S_{(0)}$ is $\{ 1, 00, 01 \}$.

- For the next level, if we use 1 as the prefix, we get the set $S_{(0,\ 1)} = \{ 00, 01, \underline{1}1, \underline{1}00, \underline{1}01 \}$

- If we use 01 as the prefix, we would get the set $S_{(0,\ 01)} = \{ 1, 00, \underline{01}1, \underline{01}00, \underline{01}01 \}$.

Example: T-Codes

- Consider the message helloworld! that contains 8 different characters { h, e, l, o, w, r, d, ! } with frequencies { 1, 1, 3, 2, 1, 1, 1, 1 } respectively.

- Encoding this message requires constructing a T-code set with T-augmentation level 3 (i.e., $\log_2 8$).

- Using short codes as prefixes at each T-augmentation level, we get the T-code set

- $S_{(0,\ 1,\ 00)}$ = { 01, 11, 100, 101, 0000, 0001, 00100, 00101 }.

Example: T-Codes

Character	Code
h	100
e	101
l	01
o	11
w	0000
r	0001
d	00100
!	00101

helloworld! = 100.101.01.01.11.0000.11.0001.01.00100.00101

Example: T-Codes

- Typical errors one may encounter while decoding a bit stream are bit losses, inversions, and additions.
- Let us examine how the bit stream representing helloworld! will be decoded in each of these cases.

Example: Bit loss

- Assume that the two underlined bits in 100.101.01.01.11.0000.11.0001.01.00100.00101 are missing.

- The bit stream will then be decoded as 100.101.01.01.100.01.100.01.01.00100.00101, or hellhlhlld!, where underlining shows the errors.

Example: Bit inversion

- Assume that the two underlined bits have been inverted in the bit stream 100.101.01.01.11. 0001.10.0001.01.00100.00101.
- The bit stream will then be decoded as 100.101.01.01.11.0001.100.00101. 00100.00101, or hellorh!d!, where the underlining shows the errors.

Example: Bit addition

- Assume that the two underlined bits had been added to the bit stream 100.1<u>11</u>01.01.01.11.0000.11.0001.01.00100.00101.
- The bit stream will then be decoded as 100.<u>11</u>.101.01.01.11.0000.11.0001.01.00100.00101, or h<u>l</u>elloworld!, where the underlining shows the errors.

Media Operations: Recap

- A solution to this is to introduce synchronisation characters in the message stream.
  - Losing bits within two synchronisation markers will mean losing only that part of the message.
- But, there is a considerable overhead in using synchronisation characters.
- Another solution is to use self-synchronising code sets to encode the message.
  - This, one will notice, has little overhead.

Message Replication

- In order for the message to survive operations such as cropping, we can use start and end delimiters for every message, and where possible the message is embedded multiple times.
- Parts of the extracted messages may be corrupt, but a best match will give us the full message in most cases.
- Even if the full message cannot be obtained, the partial message usually gives some meaningful indication of the original message.

Rotations & Flips

- If we use an LSB steganography, rotations and flipping of the media can corrupt the message.
- The message can be made resistant to these operations with some slight modification.
- For instance, rotating or flipping of an image results in changing the origin and scan directions; when extracting the message, we therefore check all the possible combinations of the origin and scan directions.

LSB Steganalysis

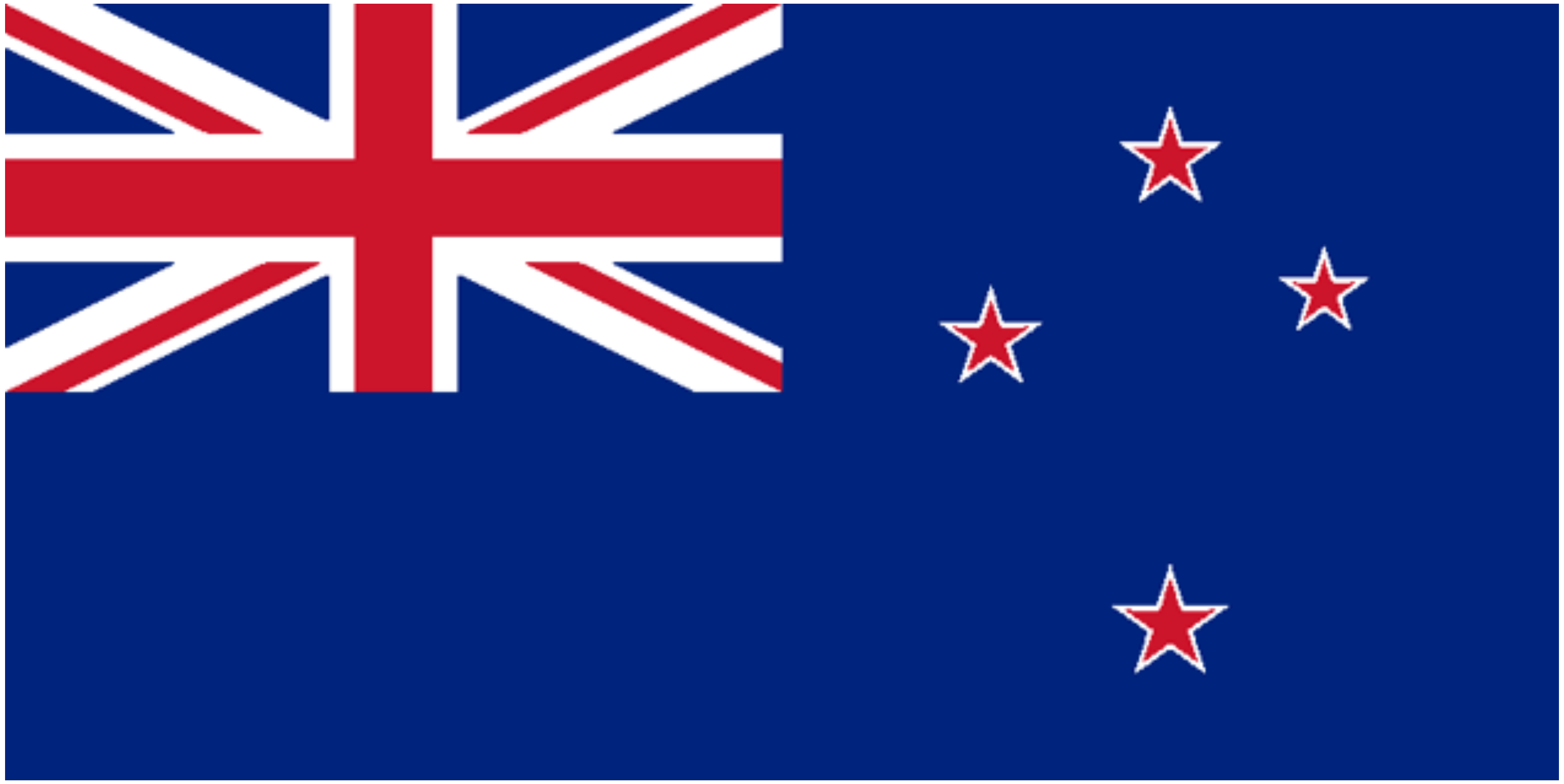
Steganalysis

- Most steganalysis schemes attempt to detect if there is hidden information in a given media or not
- A few schemes attempt to detect the size of the hidden information if there is any
- Here we will look at a simple scheme that falls under the second category: a scheme that detects the size of the hidden information

LSB Steganography

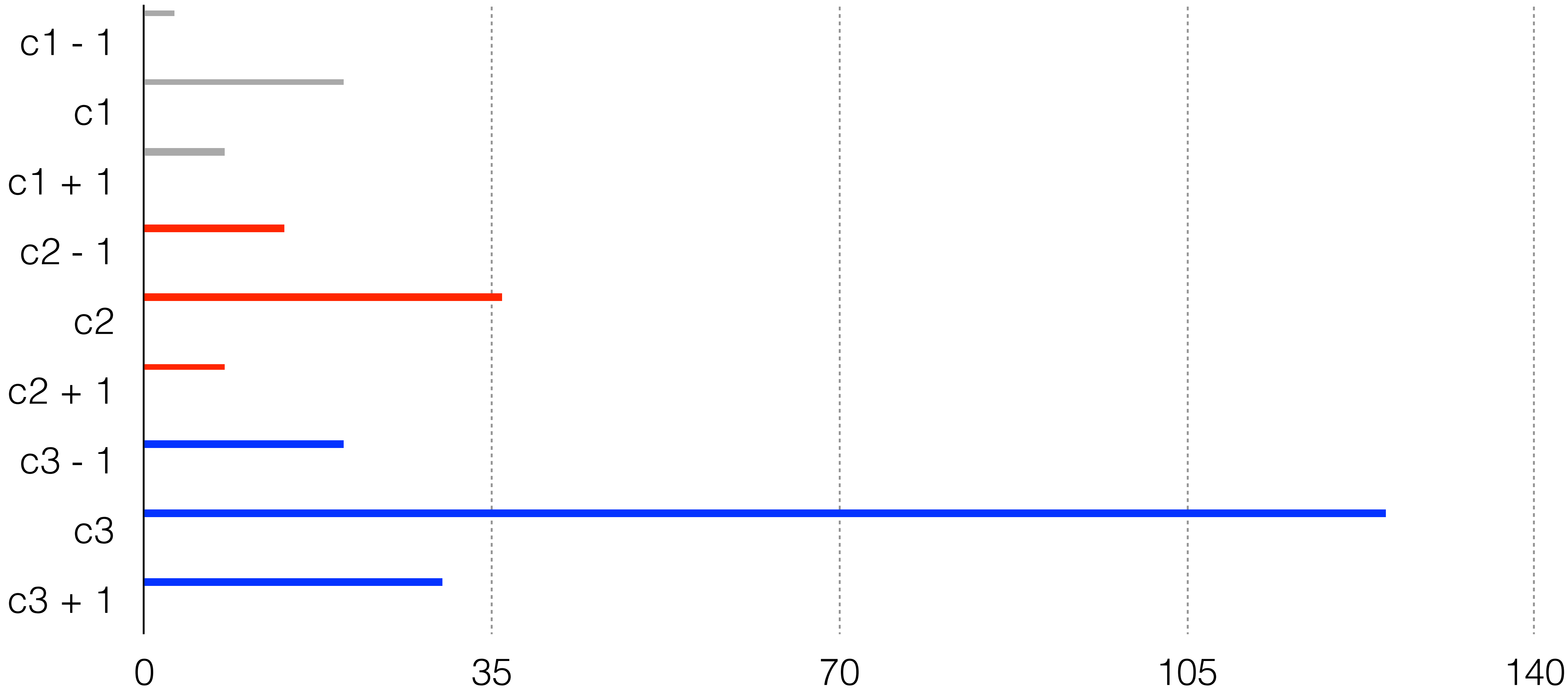
- Only about 50% of the chosen cover pixels actually change their values
- The new colour is either the old colour plus one or old colour minus one
- We use these two observations to estimate the size of hidden message

Example



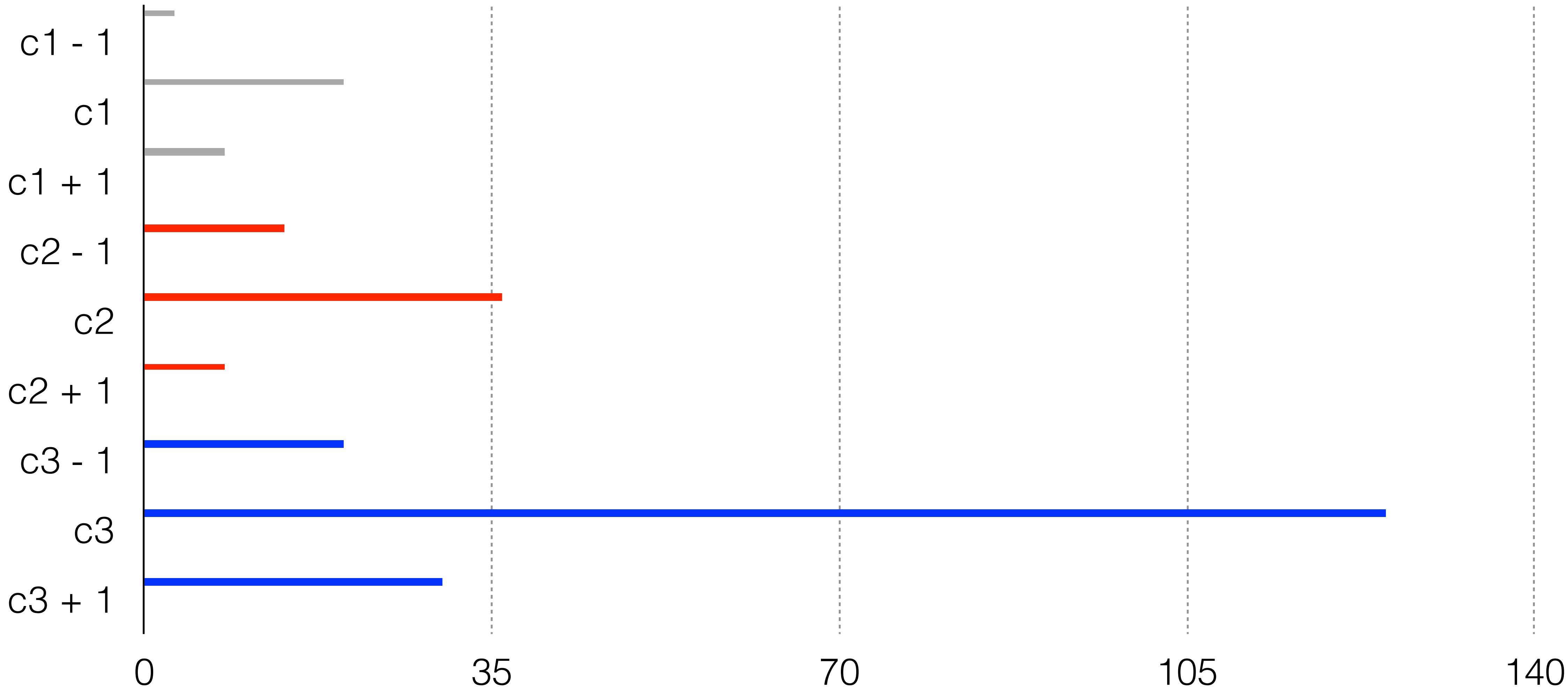
3 Colours: c_1, c_2, and c_3. Possible new colours after hiding a message are c_1, c_1+1, c_1-1, c_2, c_2+1, c_2-1, c_3, c_3+1, and c_3-1.

Example



3 Colours: c_1, c_2, and c_3. Possible new colours after hiding a message are c_1, c_1+1, c_1-1, c_2, c_2+1, c_2-1, c_3, c_3+1, and c_3-1.

Example



$$\frac{[\ \#(c_1+1) + \#(c_1-1) + \#(c_2+1) + \#(c_2-1) + \#(c_3+1) + \#(c_3-1)\ ]}{\text{Total pixel count}}$$

Example

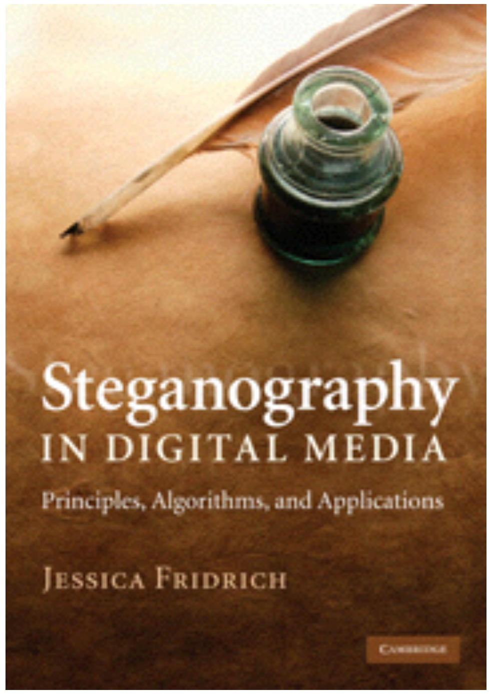
- Message length estimation using close-colours
- Works only for synthetic images with a small number of colours (e.g. logos and flags)
    - Unlikely that information will be hidden in images with low colour count
    - Interesting nevertheless since the scheme detects the message length accurately.

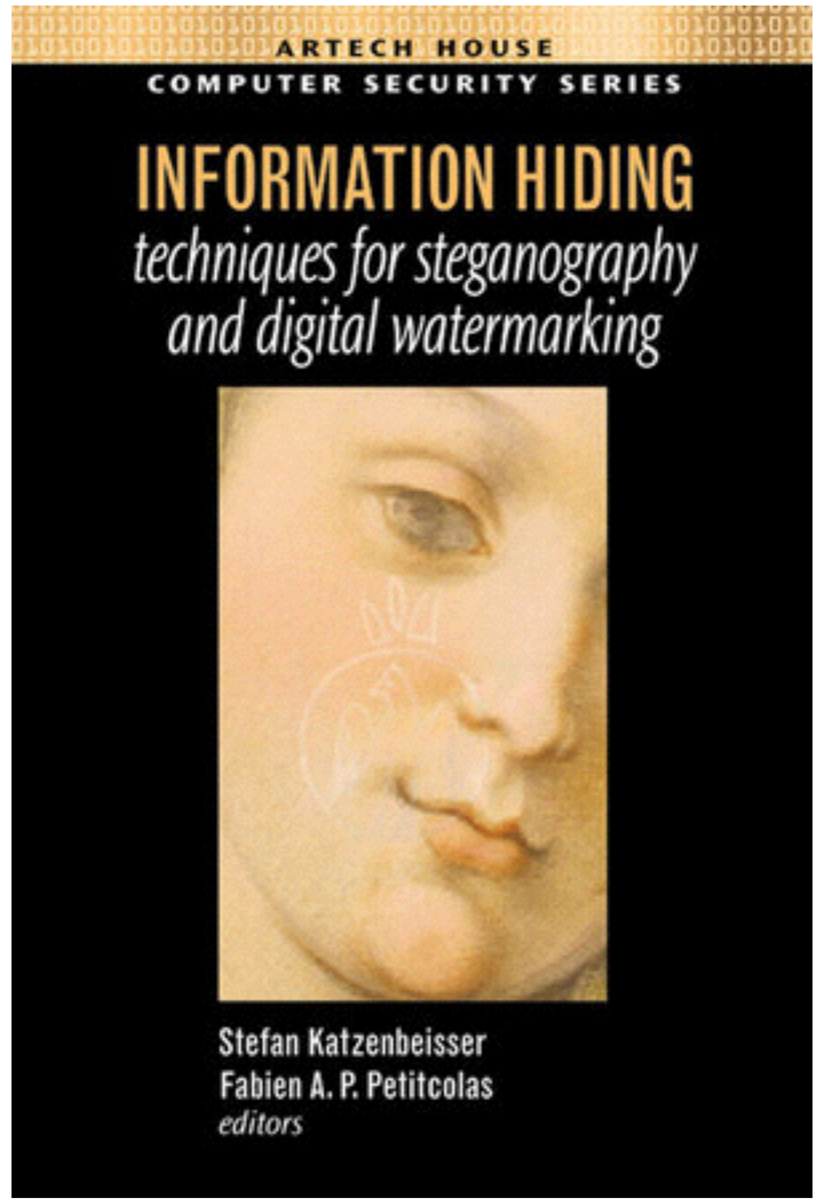
Summary

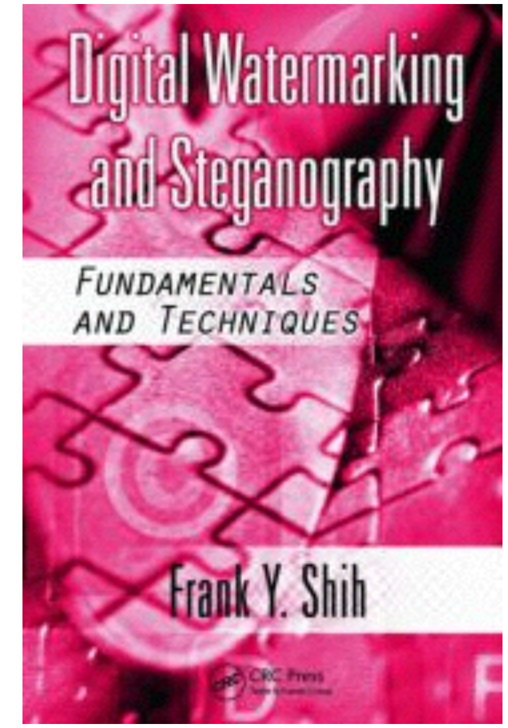
- Steganography is a promising approach to have private communication that complements cryptography.

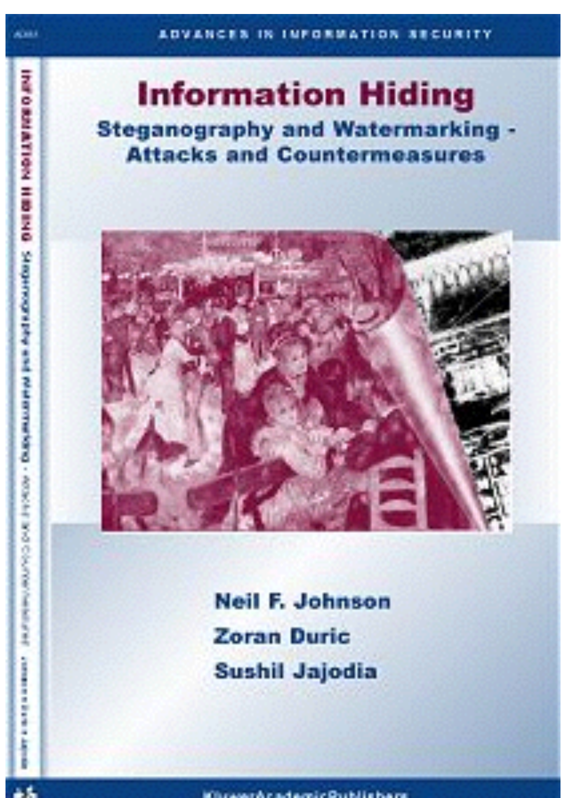
- Steganalysis is required for law-enforcing agencies.

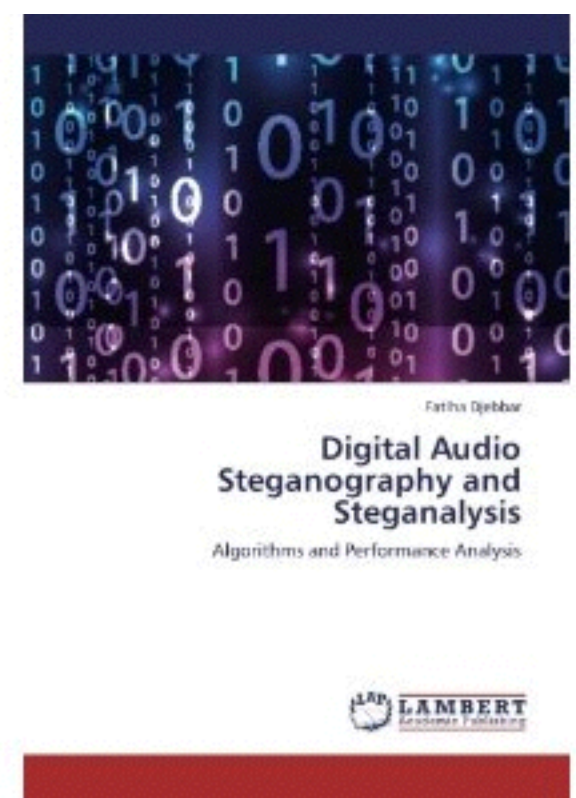
Standing on others shoulders


Steganography
IN DIGITAL MEDIA
Principles, Algorithms, and Applications
JESSICA FRIDRICH


ARTECH HOUSE
COMPUTER SECURITY SERIES
INFORMATION HIDING
techniques for steganography and digital watermarking
Stefan Katzenbeisser
Fabien A. P. Petitcolas
editors


Digital Watermarking and Steganography
FUNDAMENTALS AND TECHNIQUES
Frank Y. Shih
CRC Press


ADVANCES IN INFORMATION SECURITY
Information Hiding
Steganography and Watermarking - Attacks and Countermeasures
Neil F. Johnson
Zoran Duric
Sushil Jajodia
KluwerAcademicPublishers


Fatiha Djebbar
Digital Audio Steganography and Steganalysis
Algorithms and Performance Analysis
LAMBERT

? ¿