# Digital Watermarking and Steganography

**by Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, Ton Kalker**

## Chapter 12. Steganography

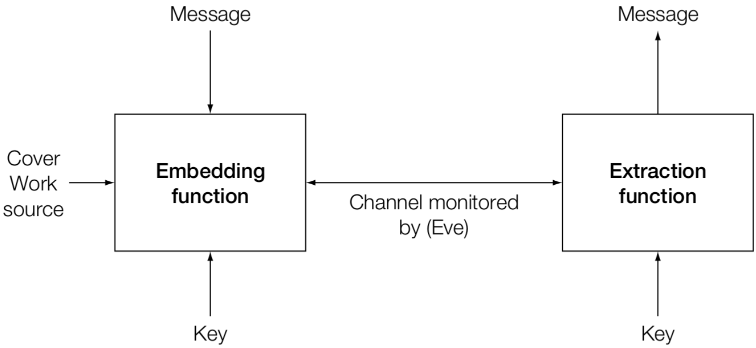
Lecturer: Jin HUANG

2015

# Difference to Watermark

- Imperceptible: watermark.
- Undetectable: steganography.

# The Model


Message
Message
Cover
Work
source
Embedding
function
Extraction
function
Channel monitored
by (Eve)
Key
Key

# The Warden

The warden is part of the channel.

- **Passive**
- Active
- Malicious: trying to impersonate Alice or Bob or otherwise tricking them.

# Embedding

The cover work is

- Preexisting, and will not be modified: cover lookup.
- Generated, and will not be modified: cover synthesis.
- Preexisting and modified: cover modification.

# Look up

- Labeling work by messages.
- Deliver the messages by sequence of transmission.

Example

- $1024$ songs for $10$-bit message.
- $1024$ sequential transmissions lead to $10k$-bit.

# Synthesis

Creates the stego Work without recourse to a cover Work.

British spies in Wold War II

- Source: a big book of conversations.
- By selecting different phrases from the book.

Packed but nature sequence of look up.

# Modification

- Type and magnitude of change.
- Location of change
  - Sequential
  - (Pseudo) random: pseudo-random walk.
  - Adaptive: informed.

# The Secret Key

Shared between Alice and Bob

- Seed the pseudo-random walk.
- Seed the noise signal.

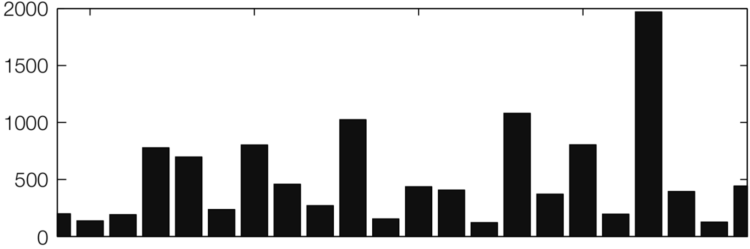
# The First Attempt

Using LSB.

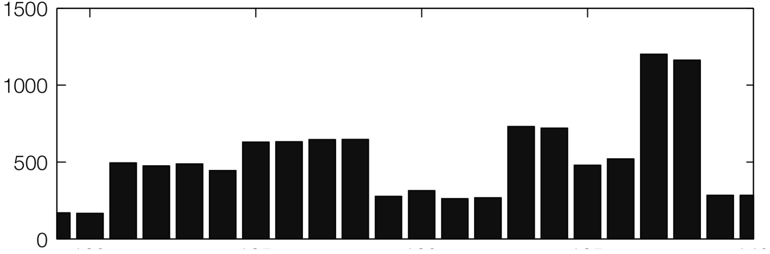
pixel values can be divided into disjoint pairs of values

- $(2i, 2i+1)$
- $2i \to 2i+1 : 1$, $2i+1 \to 2i : 0$.

# A Comparison




normal picture




LSB picture

# Practical Steganographic Methods

- OutGuess
- Masking Embedding as Natural Processing

# For Simple Detection

In a bin consists of a pair of values $(f, \bar{f})$.

In normal work, if $f > \bar{f}$, how much information can be embedded into this bin?

Let fraction $\alpha$ is used to embed

$$f' = f - \frac{\alpha}{2}(f - \bar{f})$$
$$\bar{f}' = \bar{f} + \frac{\alpha}{2}(f - \bar{f})$$

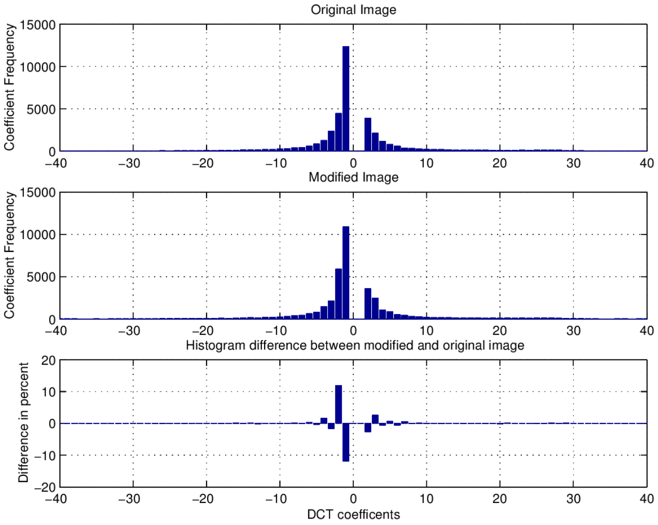
So

$$f' > \bar{f}' \Longrightarrow \alpha \leq \frac{2\bar{f}}{f + \bar{f}}.$$

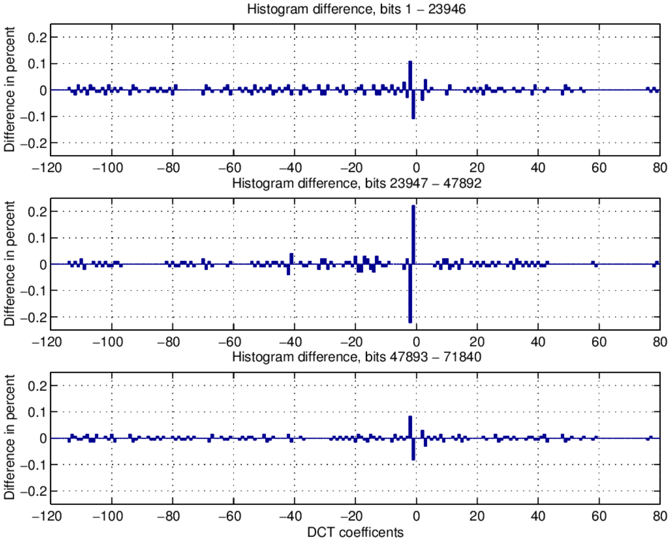
# Capacity

- Embedding capacity.
- Steganographic capacity.

# Small $\alpha$


Original Image
Coefficient Frequency
15000
10000
5000
0
−40
−30
−20
−10
0
10
20
30
40
Modified Image
Coefficient Frequency
15000
10000
5000
0
−40
−30
−20
−10
0
10
20
30
40
Histogram difference between modified and original image
Difference in percent
20
10
0
−10
−20
−40
−30
−20
−10
0
10
20
30
40
DCT coefficents

# More Advanced Method




*Defending Against Statistical Steganalysis.*

# Basic Idea

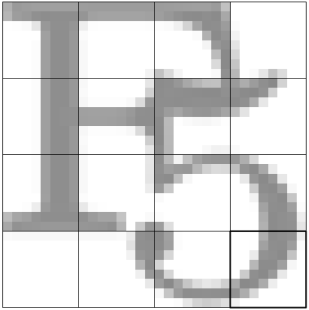
Each bin contains a lots of pixel pairs.

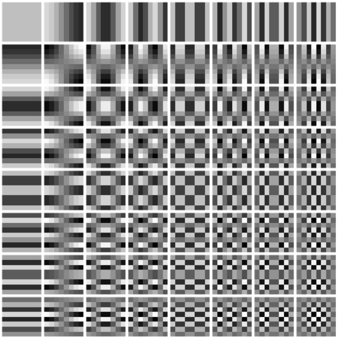
- Some of them for embedding.
- Some of them for correction.

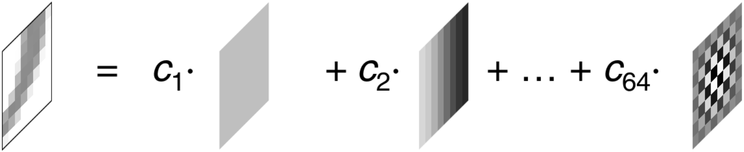
Identical histogram

- One embedding goes with one correction.

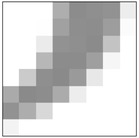
# DCT Coefficients





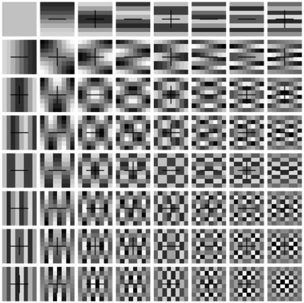


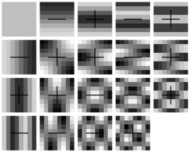
# DCT Compression

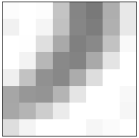


64 brightness values

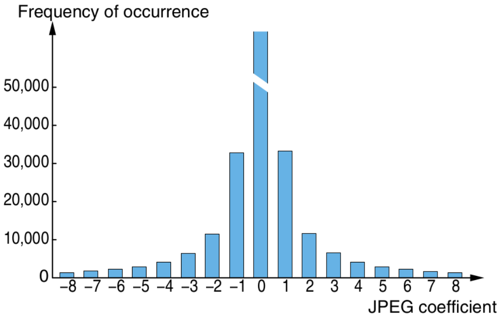
➽ 19 nonzero JPEG coefficients







# DCT Characteristic Properties




$$P(X=1) > P(X=2) > P(X=3) > P(X=4)$$

$$P(X=1)-P(X=2) > P(X=2)-P(X=3) > P(X=3)-P(X=4)$$

# Model-Based Steganography

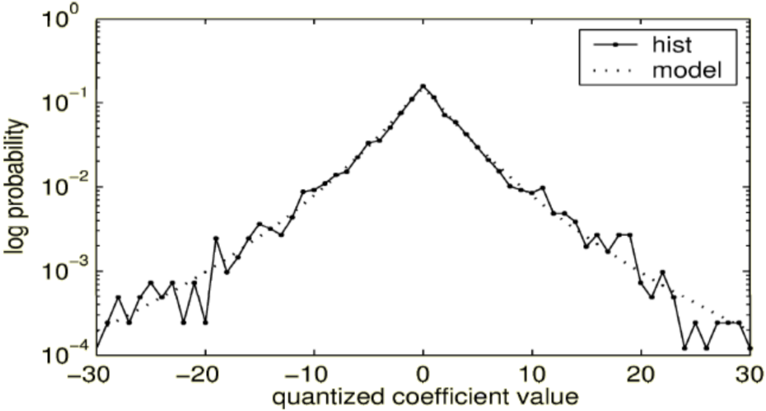
Generalized Cauchy model with probability density function (pdf)

- Generalized Cauchy distribution (GCD):

$$P(x) = \frac{p-1}{2s} \left| \frac{|x|}{s} + 1 \right|^{-p}.$$

- $p > 1, s > 0$ are the two parameters.

# Illustration of GCD



# Two-Class Pattern Classification

Two components in a cover work $(c_{inv}, c_{emb})$:

$$\begin{aligned} p_0 &= P(c_{emb} = 0 | c_{inv} = MSB_7(2i)) \\ &= \frac{T_c[2i]}{T_c[2i] + T_c[2i+1]} \\ &= 1 - P(c_{emb} = 1 | c_{inv} = MSB_7(2i)). \end{aligned}$$

The probability of $2i$ in the bin $(2i, 2i+1)$.

# Arithmetic Decompress and Compress

Map a uniformly distributed bitstream to a new bitstream with specific distribution.

**Presentation: Arithmetic Coding**

- `http://en.wikipedia.org/wiki/Arithmetic_coding`
- `http://www.cs.cmu.edu/~aarti/Class/10704/Intro_Arith_coding.pdf`

# Reverse Compression

- In embedding:

  uniformly distributed bitstream

  $\stackrel{\text{Decompress}}{\Longrightarrow}$

  GCD distributed bitstream

- In detection:

  GCD distributed bitstream

  $\stackrel{\text{Compress}}{\Longrightarrow}$

  uniformly distributed bitstream

# Embedding Efficiency

The average number of embedded bits per unit distortion.

- LSB: $2 = 1/0.5$.
  - $1$ bit: for a uniform distribution binary sequence.
  - Change: $50\%$ of chance to change.
  - Efficiency:
$$\frac{1}{0.5}.$$

# Embedding Efficiency

The average number of embedded bits per unit distortion.

- LSB: $2 = 1/0.5$.
- Model Based:
  - Information:

    $$H(p_0) = -p_0 \log_2 p_0 - (1 - p_0) \log(1 - p_0).$$

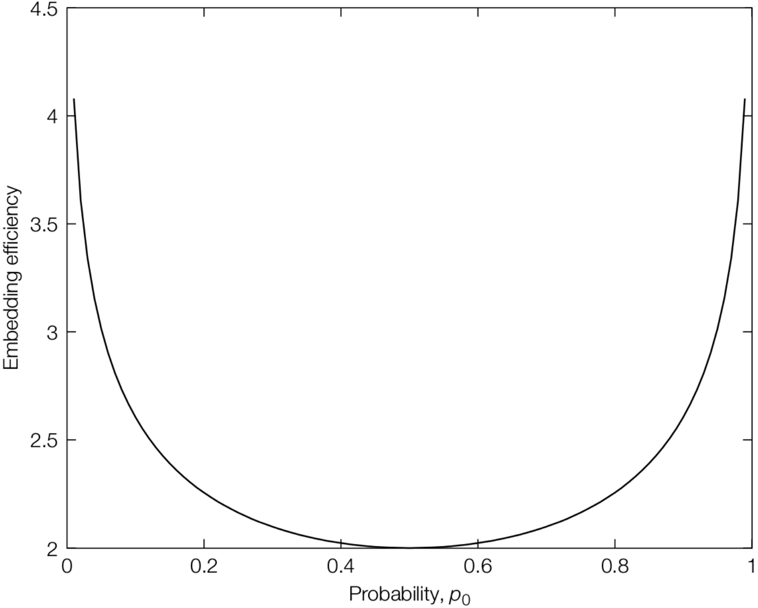
  - Change:

    $$p_0(1 - p_0) + (1 - p_0)p_0 = 2p_0(1 - p_0).$$

  - Efficiency:

    $$\frac{-p_0 \log_2 p_0 - (1 - p_0) \log(1 - p_0)}{2p_0(1 - p_0)}.$$

# Illustration

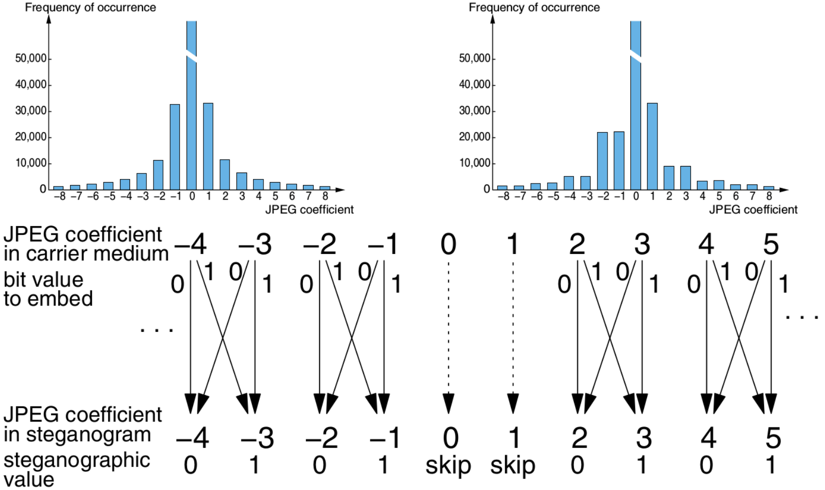


# The Cost of Correction

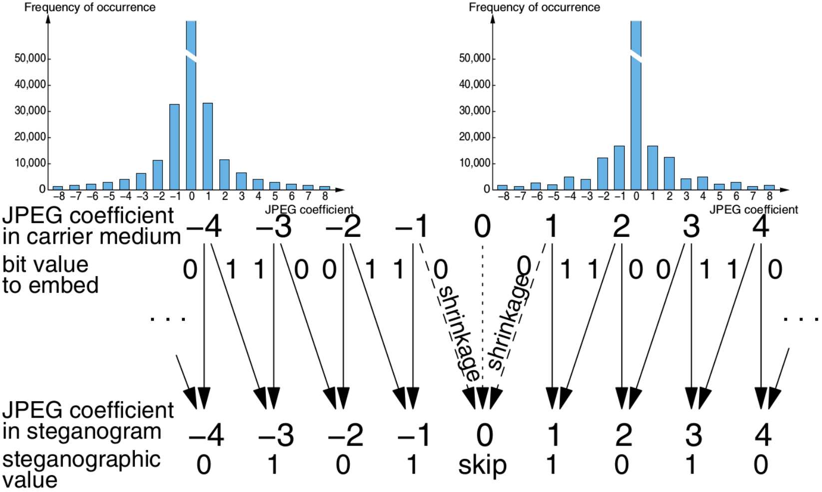
Losing capacity.

- F3, F4, F5, ...

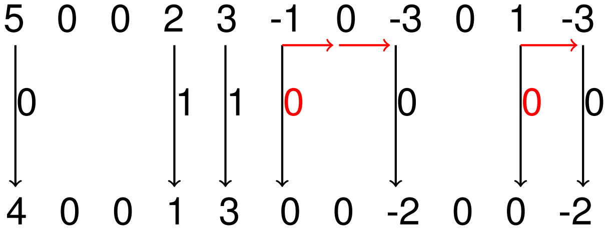
# Jsteg


Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
JPEG coefficient in carrier medium
−4 −3 −2 −1 0 1 2 3 4 5
bit value to embed
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
. . .
. . .
JPEG coefficient in steganogram
−4 −3 −2 −1 0 1 2 3 4 5
steganographic value
0 1 0 1 skip skip 0 1 0 1

# F3


Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
JPEG coefficient in carrier medium
−4 −3 −2 −1 0 1 2 3 4
bit value to embed
0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
shrinkage
shrinkage
. . .
. . .
JPEG coefficient in steganogram
−4 −3 −2 −1 0 1 2 3 4
steganographic value
0 1 0 1 skip 1 0 1 0

# F3 Algorithm




*Embedding* $01100$.

# What Is the Problem in F3?

In normal work

- Decreasing

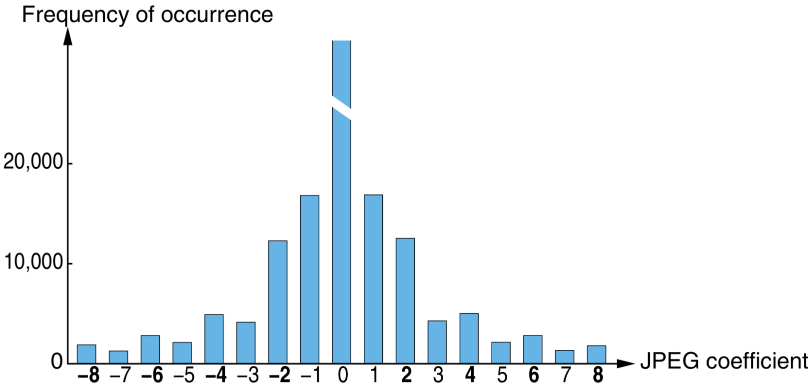
$$P(2i - 1) > P(2i).$$

In Steganographic work

- More on even.

$$P(2i - 1) < P(2i).$$

# Defects of F3

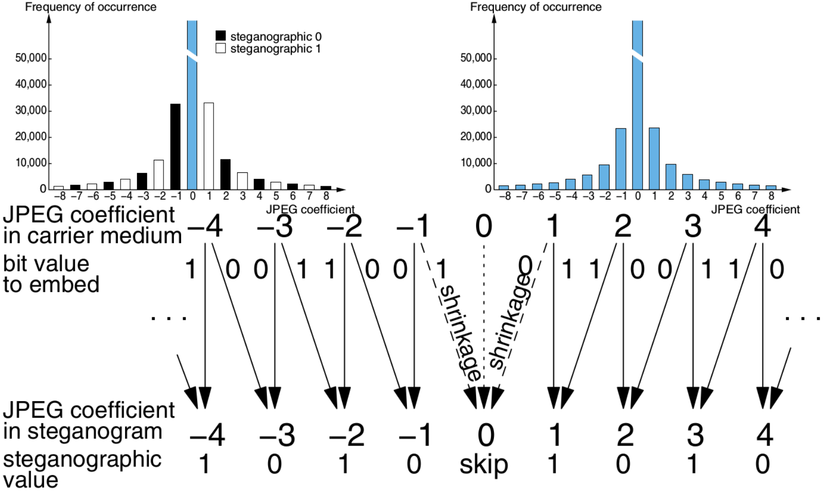


# Reason

Repeated embedding after shrinkage.

- Happens for embedding $0$ only.
- Equivalent to add more $0$ into the message code.

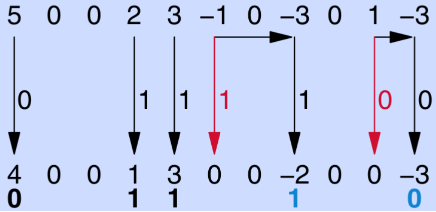
# F4



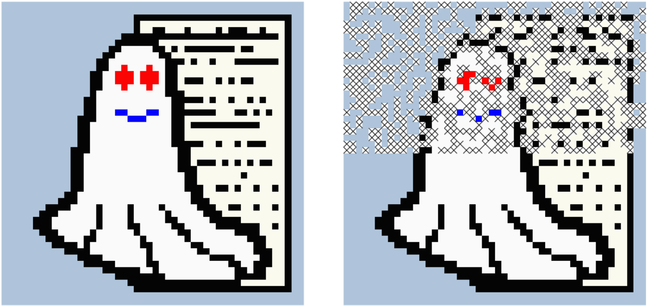
Frequency of occurrence
steganographic 0
steganographic 1
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
JPEG coefficient in carrier medium
−4 −3 −2 −1 0 1 2 3 4
bit value to embed
1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0
shrinkage
shrinkage
. . .
. . .
JPEG coefficient in steganogram
−4 −3 −2 −1 0 1 2 3 4
steganographic value
1 0 1 0 skip 1 0 1 0

# F4 Algorithm

- Steganographic interpretation
  - Positive coefficients: LSB
  - Negative coefficients: inverted LSB
- Skip 0, adjust coefficients to message bit
  - Decrement positive coefficients
  - Increment negative coefficients
  - Repeat if shrinkage occurs

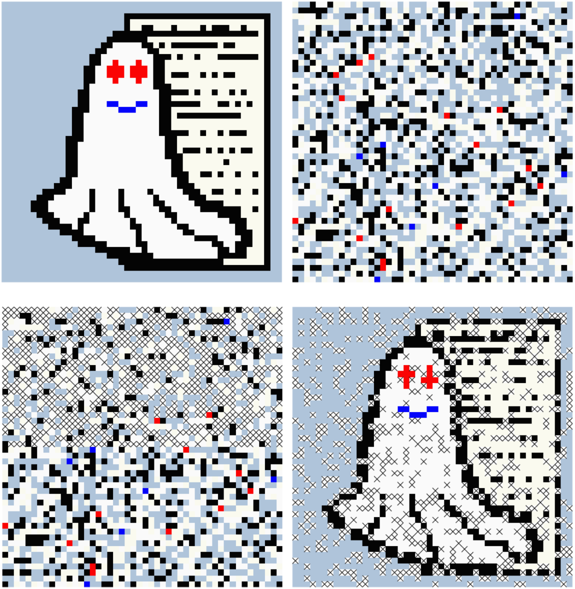


# F4 Defects



*Compare similar blocks or reverse fitting GCD.*

# Random Walk



# More Payload?

Example: Embedding 1736 bits

- F4: 1157 changes.
- F5: 459 changes by matrix encoding.
  - Embedding efficiency: 3.8 bits per change.

# Matrix Encoding

Embedding $b_1, b_2$ to $x_1, x_2, x_3$ with at most $1$ change.

$$b_1 = LSB(x_1) \text{ XOR } LSB(x_2)$$
$$b_2 = LSB(x_2) \text{ XOR } LSB(x_3)$$

- Four equal probability cases.
- Change $x_i$ accordingly.

# Example

$$b_1 = LSB(x_1) \text{ XOR } LSB(x_2)$$
$$b_2 = LSB(x_2) \text{ XOR } LSB(x_3)$$

| 0,0 | 1,0 | 0,1 | 1,1 |
|---|---|---|---|
| / | $\bar{x}_1$ | $\bar{x}_3$ | $\bar{x}_2$ |

Efficiency:

$$2/(3/4) = 8/3 > 2.$$

# Question: Matrix Embedding

| | (62, 96, 47) | (73, 45, 86) |
|---|---|---|
| (0, 1) | | |
| (1, 1) | | |

*Fill the encoded value.*

| (16, 69, 35) | (94, 23, 88) |
|---|---|
| | |

*Decode the message.*

# A Hamming Code

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

# Presentation: Matrix Embedding

- The idea of parity matrix.
- Efficiency.

# Upper Bound on Embedding Efficiency

For a message set $\mathcal{M}$, in a $n$-pixel image, what is the minimal number of change $R$ (in the sense of expectation).

- The bound of $\frac{\log_2 |\mathcal{M}|}{R}$:
  - Larger means better efficiency.
  - The upper bound indicates the optimal situation.

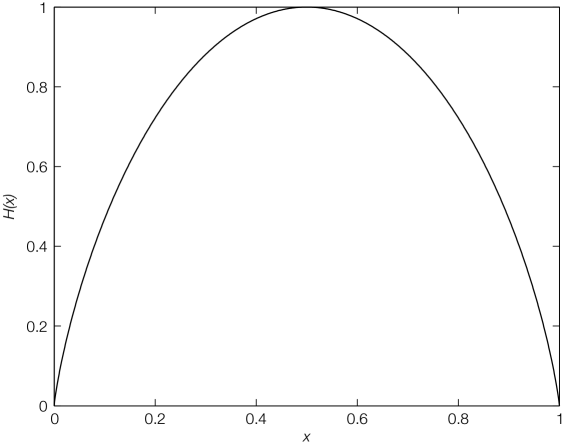
# Just Some Math

$$
\begin{aligned}
\log_2 |\mathcal{M}| &\le \log_2 \sum_{i=0}^{R} \binom{n}{i} 2^i \\
&\le nH(R/n) \quad \text{information theory}
\end{aligned}
$$

# $H(x)$

Binary entropy function

$$H(x) = -x \log_2 x - (1-x)\log_2(1-x).$$

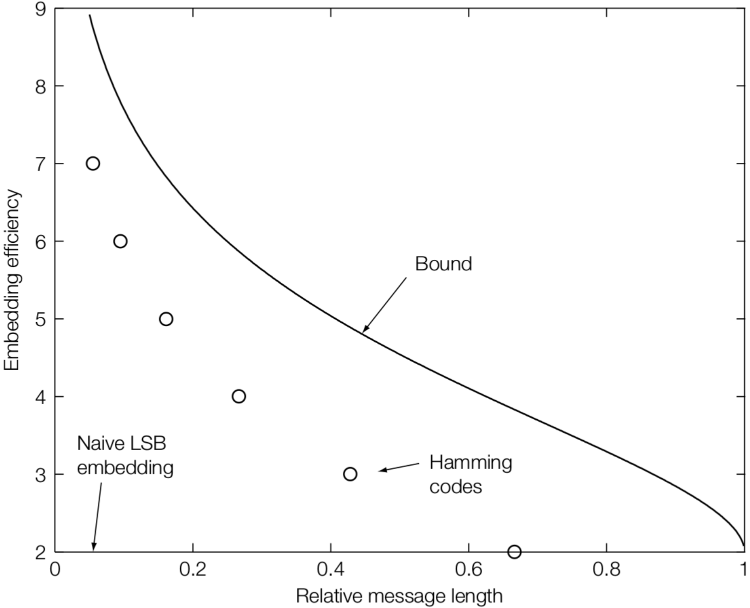


# Continue the Math

$$\begin{aligned}
\alpha = \frac{\log_2 |\mathcal{M}|}{n} &\le H(R/n) \\
\frac{n}{R} &\le 1/H^{-1}(\alpha), \quad H^{-1} \in [0, 0.5] \\
\frac{\log_2 |\mathcal{M}|}{R} \frac{n}{\log_2 |\mathcal{M}|} &\le 1/H^{-1}(\alpha) \\
e = \frac{\log_2 |\mathcal{M}|}{R} &\le \frac{\alpha}{H^{-1}(\alpha)}.
\end{aligned}$$

- $\alpha$: relative message length.
- $e$ embedding efficiency.

# Illustration



# Selection Rule

Choose the parts/locations to change.

- Known for both side: shared.
- Only known for sender: nonshared.

# Nonshared Selection Rule

Motivation:

- In JPEG compress:
  - DCT: float value.
  - Round into integer.
- To minimize the change:
  - Choose values have larges rounding error to change, e.g. $5.47$:
    - to embed $0$: $5.47 \to 5, -0.47$.
    - to embed $1$: $5.47 \to 6, +0.53$.
- More like normal compress procedure, but
  - How recipient detect the message?

# Other Cases

- Adaptive steganography
  - If the neighborhood has certain property ...
  - But embedding may change the property.
- Eg. using the pixels with largest neighbor variance.

# Writing on Wet Paper

- Cover image (paper) $x$: has wet region.
- Only allowed to slightly modify the dry part.
- The received image (paper) $y$ drys.
- Where the message is written?

# An Equivalent Well-Known Problem

In information theory: writing in memory with defective cells.

- Writer known the location of stuck cells.
- Reader do not know that.
- How to correctly read that?
- How to write as many bits as possible?

A special case of the Gel'fand-Pinsker channel.

# The Idea, Matrix Embedding Again!

Message $\mathbf{m} \in \{0,1\}^m$ in $y \in \mathbb{Z}^n$ via a parity matrix $\mathbf{D} \in \{0,1\}^{m \times n}$:

$$\mathbf{D}_{m \times n} \mathbf{y}_{n \times 1} = \mathbf{m}_{m \times 1}.$$

XOR is addition modulo $2$.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 16 \\ 69 \\ 35 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

# Wet Paper

In $\mathbf{x}$, we only change part of it.

- Dry part: $\mathbf{x}[j], j \in \mathcal{J} \subset \{1, \cdots, n\}$.
  - Can be changed.
- Wet part: $\mathbf{x}[j], j \notin \mathcal{J}$.
  - Cannot be changed, i.e. fixed.

Thus the change $\mathbf{v} = \mathbf{y} - \mathbf{x}$ has the property:

$$\mathbf{v}[j] = 0, j \notin \mathcal{J}.$$

# A Constrained Equation

Under the constraints:

$$\mathbf{v}[j] = 0, j \notin \mathcal{J}.$$

Solving the following equation.

$$\begin{aligned} \mathbf{D}\mathbf{y} &= \mathbf{m} \\ \mathbf{D}(\mathbf{x} + \mathbf{v}) &= \mathbf{m} \\ \mathbf{D}\mathbf{v} &= \mathbf{m} - \mathbf{D}\mathbf{x}. \end{aligned}$$

# Removing the Known Values

Using a permutation matrix $\mathbf{P}$ to sort fixed $\mathbf{v}[j]$ to the end.

$$\mathbf{P}\mathbf{v} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{|\mathcal{J}|} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix}$$

# Continue

$$
\begin{aligned}
\mathbf{D}\mathbf{v} &= \mathbf{m} - \mathbf{D}\mathbf{x} = \mathbf{z} \\
(\mathbf{D}\mathbf{P}^{-\mathbf{1}})(\mathbf{P}\mathbf{v}) &= \mathbf{z} \\
\begin{pmatrix} \mathbf{H} & \mathbf{K} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix} &= \mathbf{z} \\
\mathbf{H}_{m\times|\mathcal{J}|}\mathbf{u} &= \mathbf{z}.
\end{aligned}
$$

Choosing the solution with the minimal number of changes.

# Question: Wet Paper

$$\mathbf{D} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} 16 \\ 69 \\ 35 \\ 47 \end{pmatrix}, \mathbf{m} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

- $\mathcal{J} = \{1, 2, 3\}$: $\mathbf{y} = ?$.
- $\mathcal{J} = \{2, 4\}$: $\mathbf{y} = ?$.

# Acceleration

- Gaussian elimination: $O(|\mathcal{J}|^3$.
- Matrix LT Process: much lower.

# Perturbed Quantization

One of the most secure steganographic schemes known today.

$$
\begin{aligned}
J = \{j | j \in \{1, \cdots, n\},& \\
&\mathbf{u}[j] \in [L + 0.5 - \epsilon, L + 0.5 + \epsilon], L \in \mathbb{Z}\}.
\end{aligned}
$$