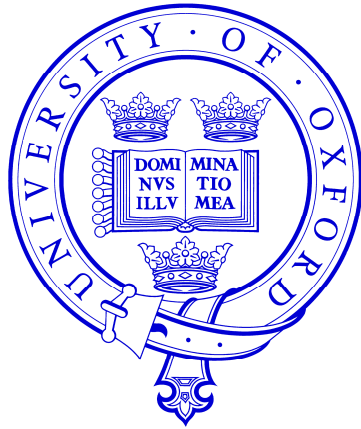


Information Hiding and Covert Communication



Andrew Ker

adk@comlab.ox.ac.uk

*Royal Society University Research Fellow
Oxford University Computing Laboratory*

Foundations of Security Analysis and Design

Bertinoro, Italy, 25-26 August 2008

Information Hiding and Covert Communication

Part 3: More Efficient Steganography

- Abstraction of the embedding problem, simple example of efficiency
- Embedding codes
 - Syndrome codes: the Hamming code family
- Embedding efficiency bound
- More embedding codes
 - Golay code
 - The ZZW construction
- Asymptotics


Abstraction

- *separate the embedding operation from the details of what payload is embedded where.*

Say that

- the cover consists of a number of **locations**,
- each location contains one q -ary symbol,
- an **embedding change** alters the symbol at one location.

We stick to binary symbols,
 $GF(2) = \{0, 1\}$



Embedding operations might be: replacement of LSB of pixel, adjustment of LSB of quantized coefficient, adjustment of remainder (mod q) ...

Aim: maximize the amount of information transmitted, while minimizing the number of embedding changes.

(Implicitly: all embedding changes are equally risky.)

Embedding efficiency

Cover:

0	1	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ...

Payload: 0 0 1 1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 1 ...

Stego:

0	0	1	1	0	1	1	1	1	0	1	1	0	0	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ...

Regardless of payload size, on average each bit of payload requires 0.5 embedding changes.

We say that the embedding efficiency $e = 2$ bits per change.

Example of efficient embedding

Cover:

0	1	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ...

Payload: **0 0 1 1 0 1 1 1 1 0 1 1** ...

Stego:

0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0
---	----------	---	---	---	---	---	----------	---	----------	---	---	---	---	----------	---	---	---	---

 ...

Take cover in groups of 3, payload in groups of 2.

Embed a payload group in each cover group using the encoding:

0 0	↔	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	or	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1
0	0	0								
1	1	1								
0 1	↔	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>0</td><td>1</td></tr></table>	0	0	1	or	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>1</td><td>1</td><td>0</td></tr></table>	1	1	0
0	0	1								
1	1	0								
1 0	↔	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	or	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>1</td><td>0</td><td>1</td></tr></table>	1	0	1
0	1	0								
1	0	1								
1 1	↔	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>1</td><td>0</td><td>0</td></tr></table>	1	0	0	or	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>1</td><td>1</td></tr></table>	0	1	1
1	0	0								
0	1	1								

Example of efficient embedding

Cover:

0	1	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ...

Payload: **0 0 1 1 0 1 1 1 1 0 1 1** ...

Stego:

0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0
---	----------	---	---	---	---	---	----------	---	----------	---	---	---	---	----------	---	---	---	---

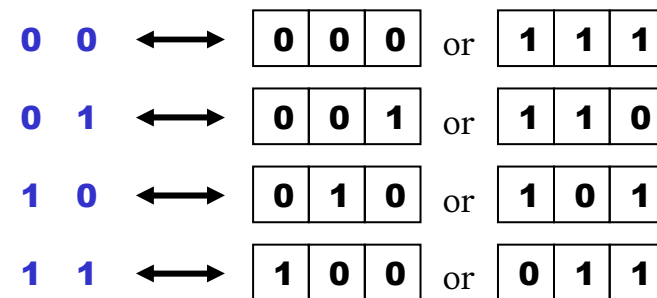
 ...

Payload rate $\alpha = 2/3$.

Embedding changes:

1/3 with probability 3/4

0/3 with probability 1/4



Embedding efficiency $e = 8/3$ bits per change.

Embedding codes

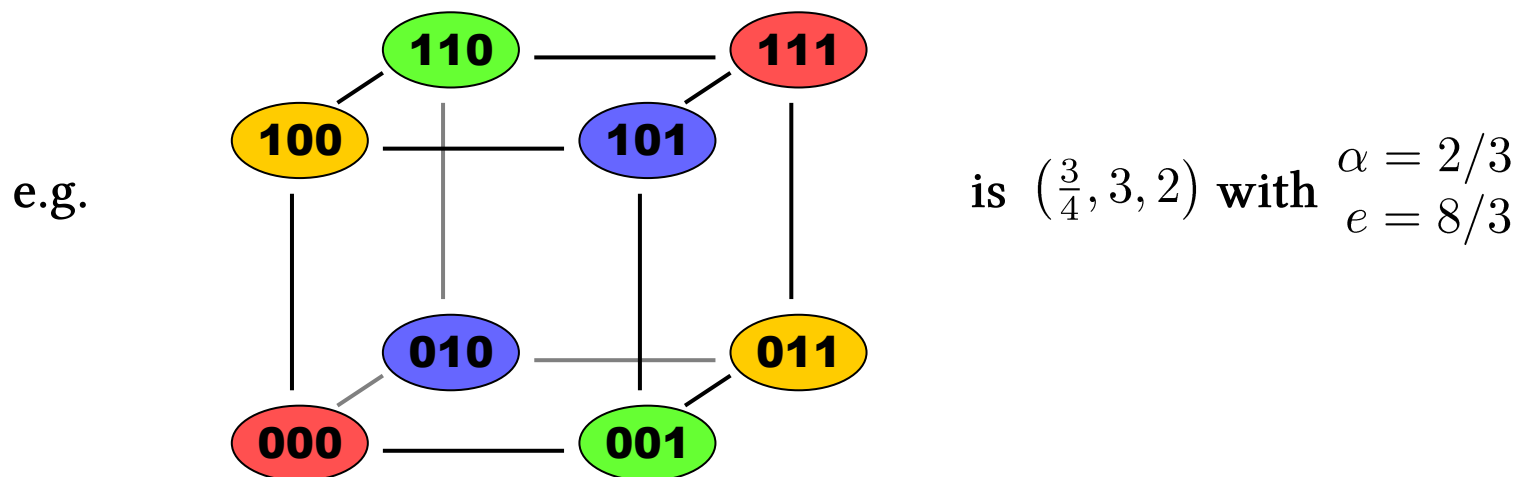
A binary embedding code is:

- a partition of $GF(2)^N$ into 2^M subsets S_1, \dots, S_{2^M} .
- an embedding map $Emb : GF(2)^N \times GF(2)^M \rightarrow GF(2)^N$ with $Emb(x, m) \in S_m$.

The average embedding distance is $\rho_\alpha = \frac{1}{2^{N+M}} \sum_{x \in GF(2)^N} \sum_{m \in GF(2)^M} d(x, Emb(x, m))$.

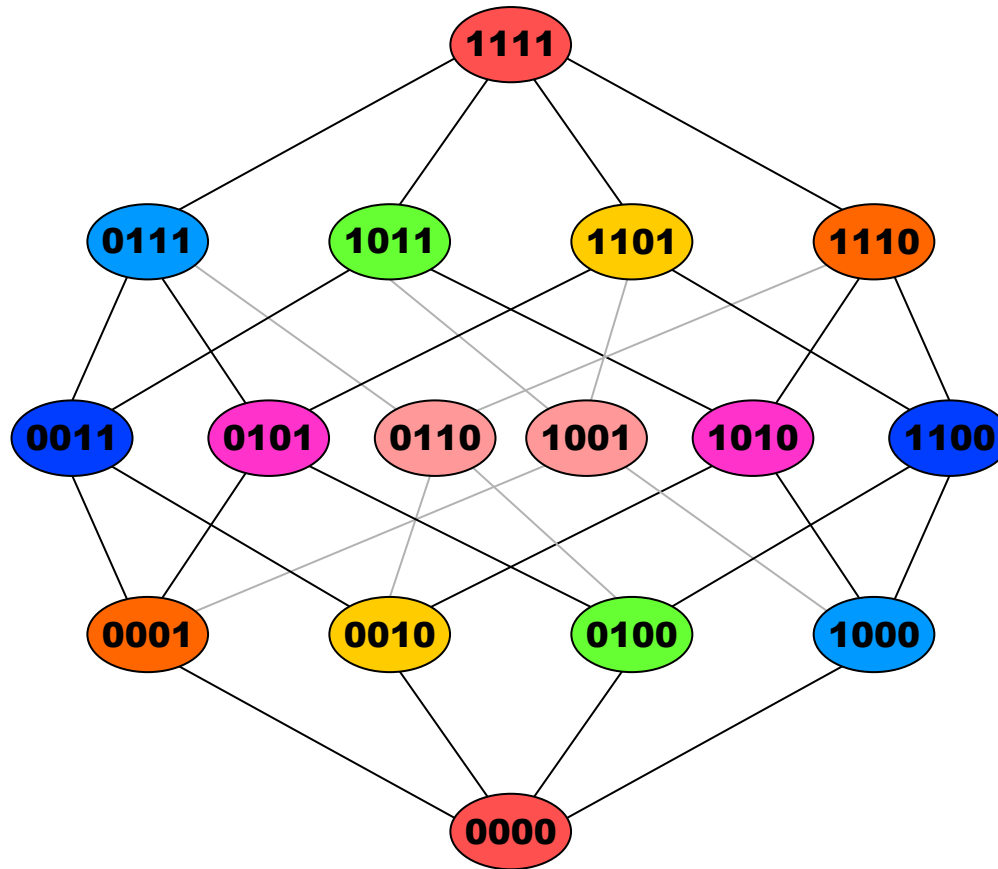
We say that the code is (ρ_α, N, M) .

It achieves a payload rate of $\alpha = M/N$ with embedding efficiency $e = M/\rho_\alpha$.



Embedding codes

e.g.



is $(\frac{5}{4}, 4, 3)$ with $\alpha = \frac{3}{4}$
 $e = \frac{12}{5}$

Syndrome coding

- A **linear** $[n, k]$ code \mathcal{C} is a k -dimensional subspace of $GF(2)^n$.
- Its **parity check matrix** is a $(n - k) \times n$ matrix H s.t. $Hx = 0$ iff $x \in \mathcal{C}$.
- For each s in $GF(2)^{n-k}$, the **coset leader** for s is a vector $e_L(s) \in GF(2)^n$ with least weight (number of 1s) such that $He_L(s) = s$.

Matrix Embedding Theorem

Every linear $[n, k]$ code generates a $(\rho_a, n, n - k)$ embedding code by the following construction:

$$S_m = \{y \in GF(2)^n \mid Hy = m\}$$

$$Emb(x, m) = x + e_L(m - Hx)$$

Then ρ_a is the average weight of all coset leaders.

Hamming code family

For $p = 1, 2, 3, \dots$ we can form a binary $[2^p - 1, 2^p - p - 1]$ code with parity check matrix

$$\left(\begin{array}{ccc|l} 0 & 0 & 0 & \\ 0 & 0 & 0 & \text{all distinct} \\ \vdots & \vdots & \vdots & \text{nonzero} \\ 0 & 0 & 0 & \text{binary strings} \\ 0 & 1 & 1 & \text{of length } p \\ 1 & 0 & 1 & \end{array} \right)$$

These are called *Hamming codes*.

All equations of the form $Hx = s$ can be solved using an x with a single nonzero entry (and $Hx = 0$ is solved by zero x) so there are

- $2^p - 1$ coset leaders of weight 1, and
- 1 coset leader of weight 0.

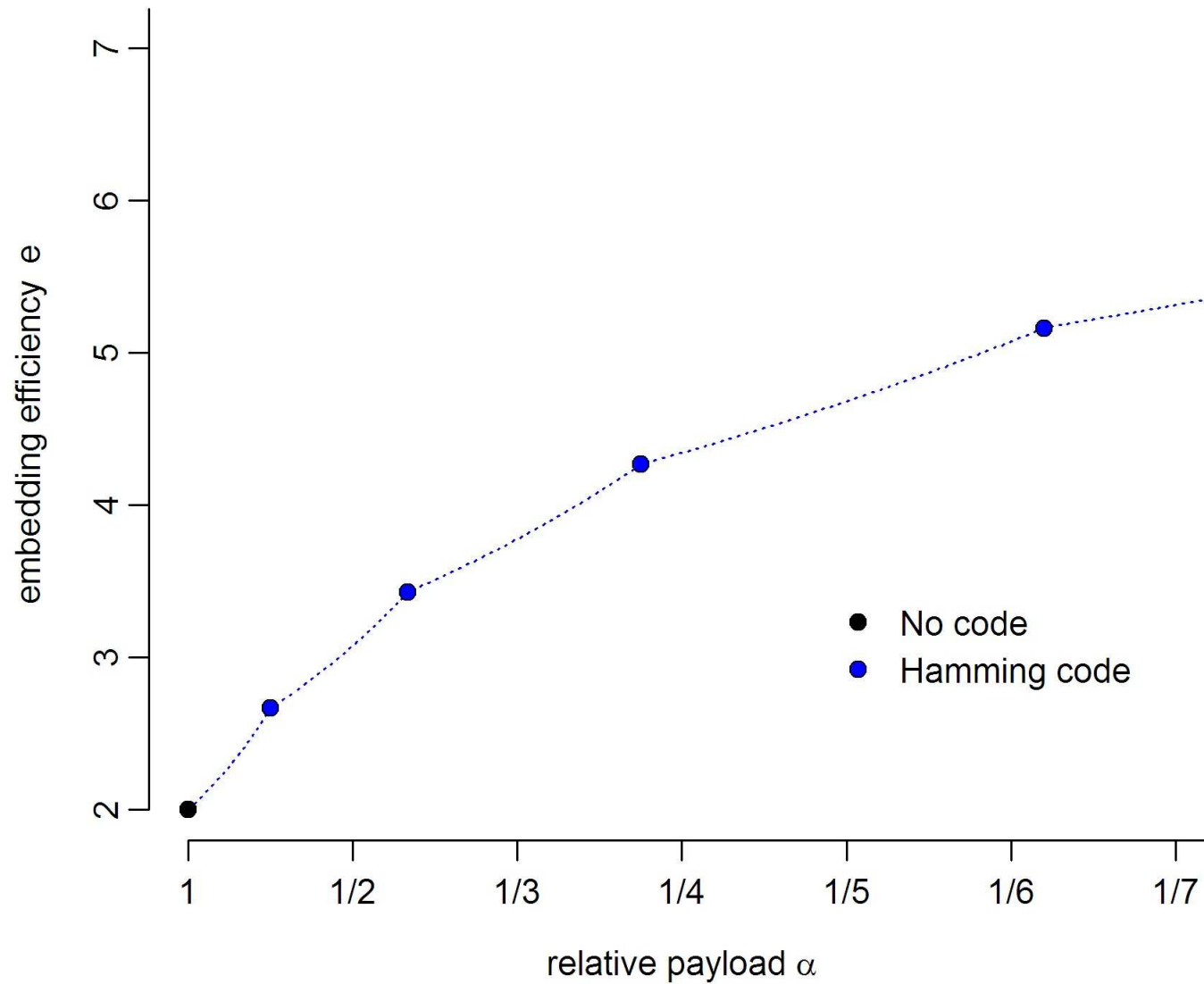
i.e. the average coset leader weight is $1 - 2^{-p}$.

So Hamming codes generate $(1 - 2^{-p}, 2^p - 1, p)$ embedding codes

$$\alpha = \frac{p}{2^p - 1}$$

$$e = \frac{p}{1 - 2^{-p}}$$

Embedding efficiency



Efficiency bound

Recall that the average embedding distance, and embedding efficiency,

$$\rho_a = \frac{1}{2^{N+M}} \sum_{x \in GF(2)^N} \sum_{m \in GF(2)^M} d(x, Emb(x, m)) \quad e = M/\rho_a$$

Related quantities are the worst-case embedding distance and worst-case embedding efficiency

$$\rho = \max_m \max_x d(x, Emb(x, m)) \quad e_w = M/\rho$$

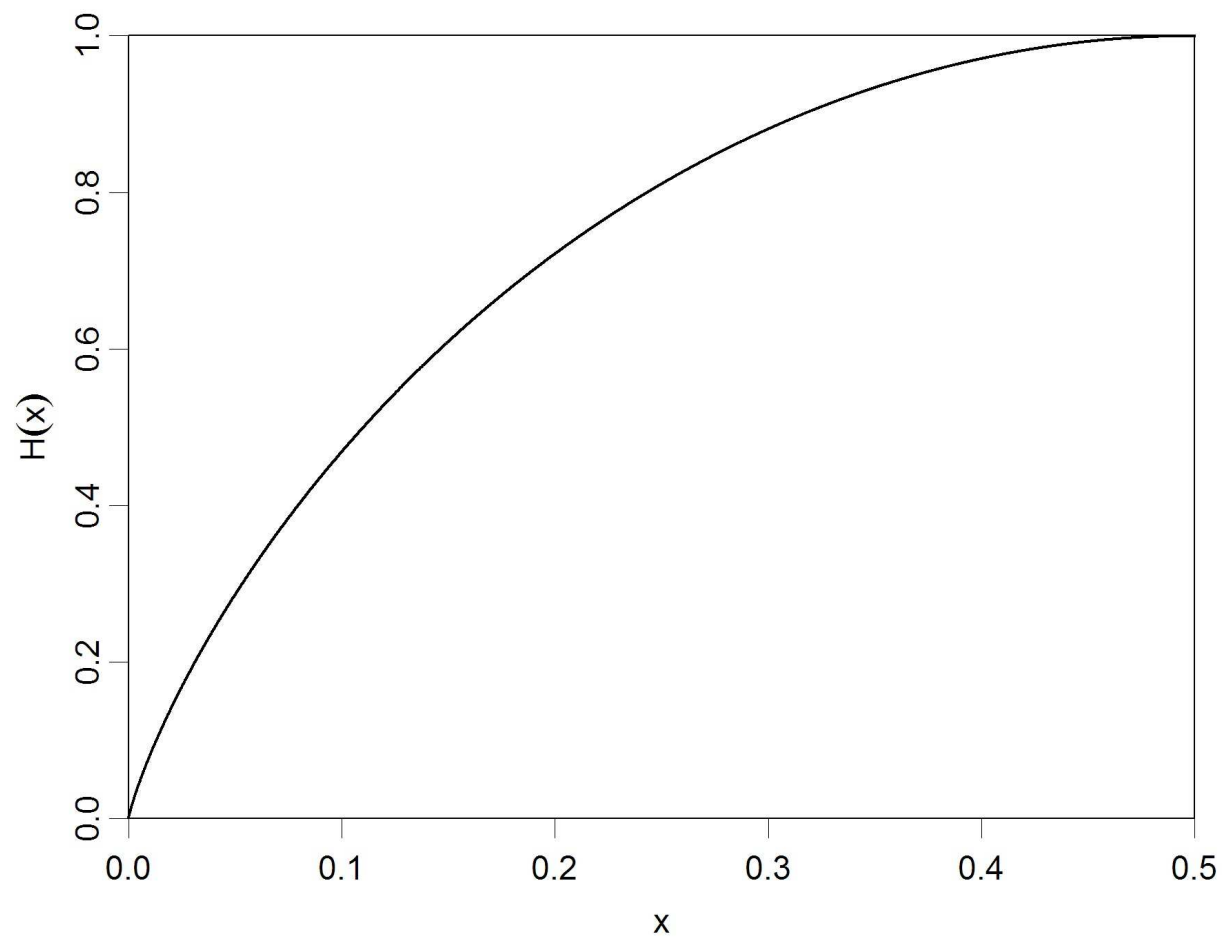
Pick any $x \in GF(2)^n$, if x can reach all 2^M subsets with $\leq \rho$ changes,

$$2^M \leq \sum_{i=0}^{\rho} \binom{N}{i} \leq 2^{NH(\rho/N)}$$

Binary entropy function

$$H : [0, 1/2] \rightarrow [0, 1]$$

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



Efficiency bound

Recall that the average embedding distance, and embedding efficiency,

$$\rho_a = \frac{1}{2^{N+M}} \sum_{x \in GF(2)^N} \sum_{m \in GF(2)^M} d(x, Emb(x, m)) \quad e = M/\rho_a$$

Related quantities are the worst-case embedding distance and worst-case embedding efficiency

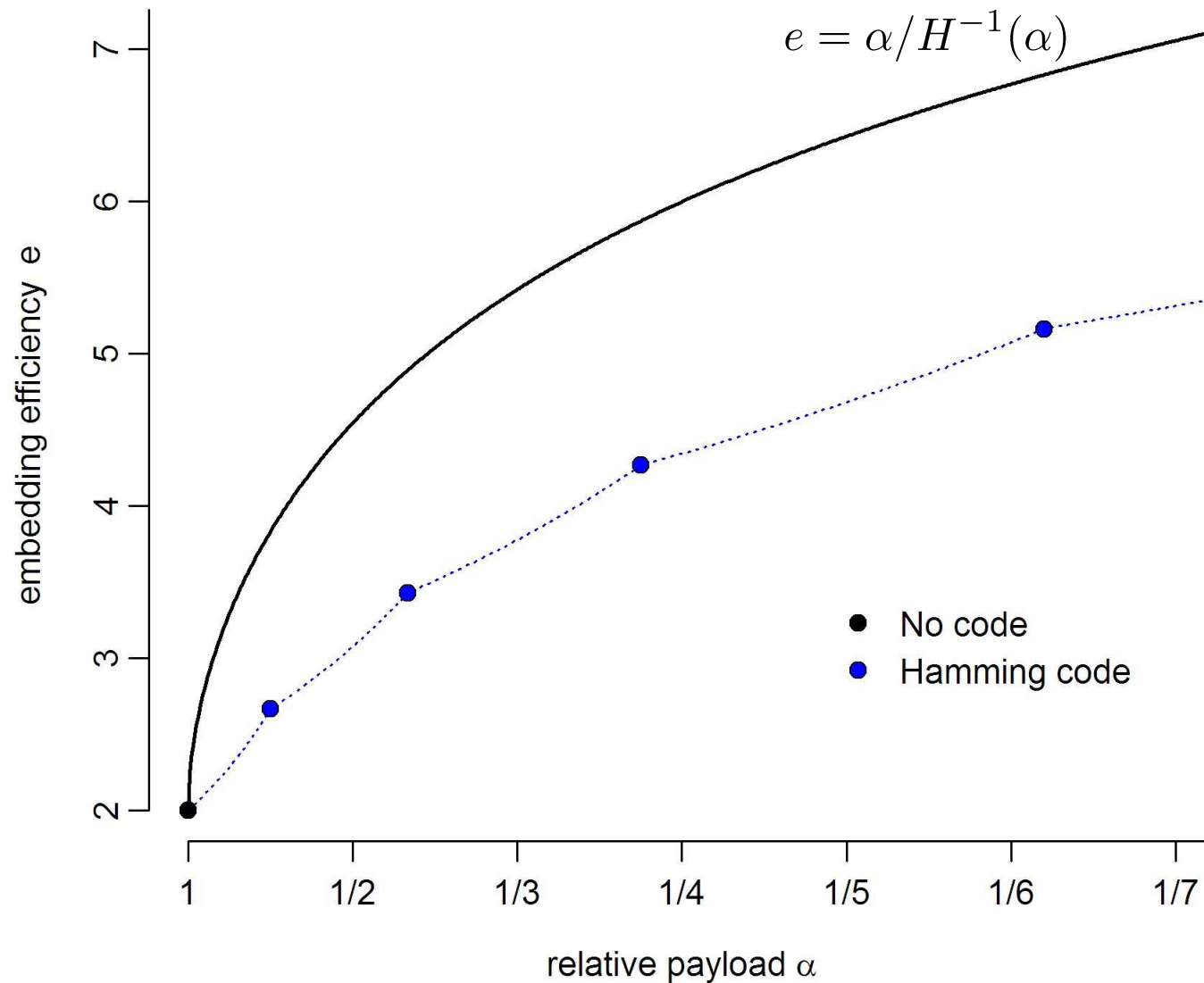
$$\rho = \max_m \max_x d(x, Emb(x, m)) \quad e_w = M/\rho$$

Pick any $x \in GF(2)^n$, if x can reach all 2^M subsets with $\leq \rho$ changes,

$$2^M \leq \sum_{i=0}^{\rho} \binom{N}{i} \leq 2^{NH(\rho/N)} \quad \longrightarrow \quad \begin{aligned} H^{-1}(M/N) &\leq \rho/N \\ H^{-1}(\alpha) &\leq \alpha/e_w \\ e_w &\leq \alpha/H^{-1}(\alpha) \end{aligned}$$

It can be shown that the same bound holds “usually” (or “asymptotically”) for the average embedding efficiency e , too.

Embedding efficiency



Perfect codes

A code is **perfect** if spheres around the codewords partition the entire space.

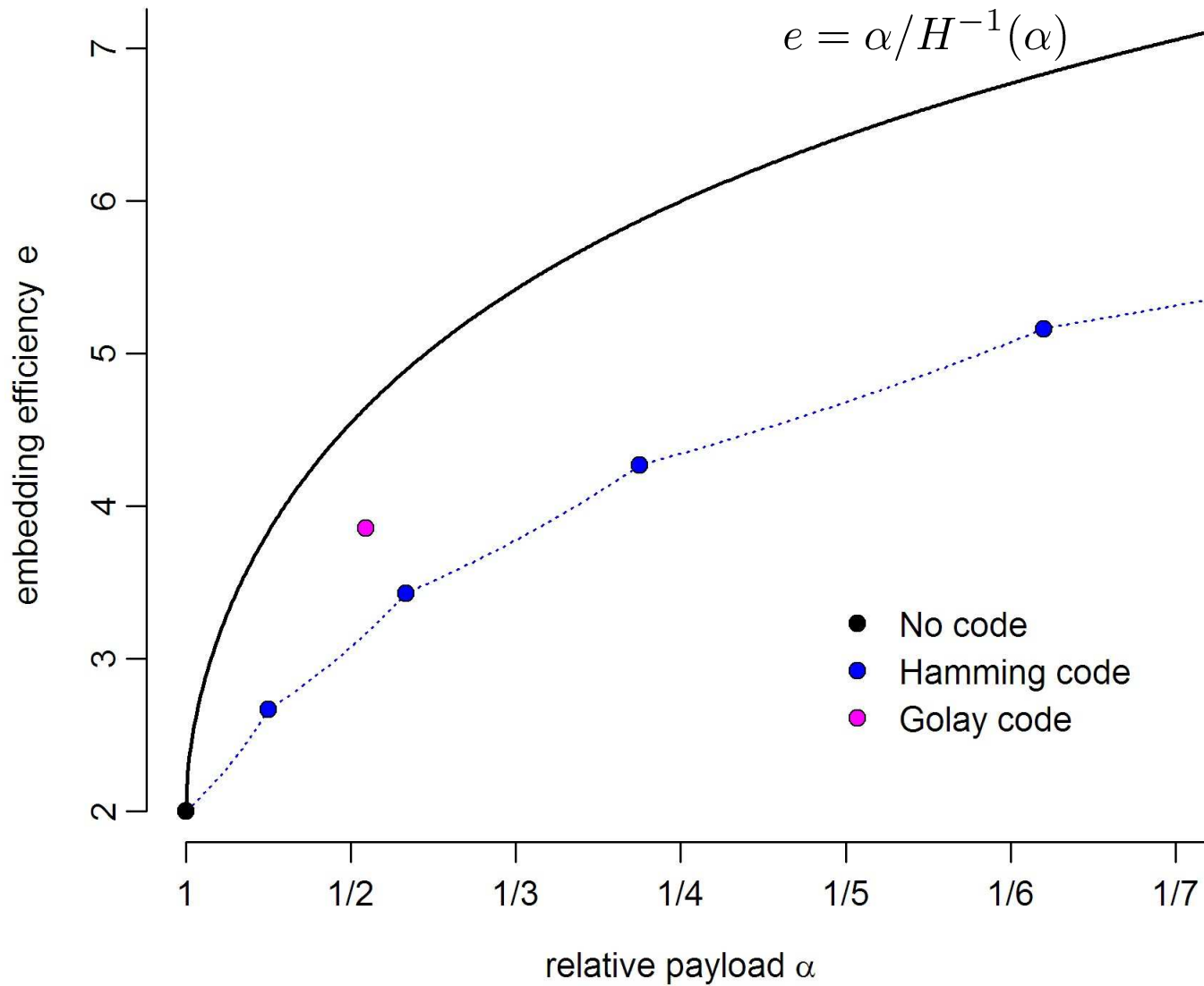
When converted to an embedding code, this corresponds to no “wasted space”, so we might look for perfect codes to make good embedding codes.

Sadly, the only nontrivial perfect binary codes are the Hamming codes and the **Golay code**, with parity check matrix

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

The Golay code gives a $(\frac{2921}{1024}, 23, 11)$ embedding code, so $\alpha = 11/23$
 $e \approx 3.856$

Embedding efficiency



The ZZW construction

Fix positive integer p . Split cover into groups of size $q = 2^p$.

For each group, suppose that the cover bits are x_1, \dots, x_q . Compute their XOR $x_1 \oplus \dots \oplus x_q$.

- If this matches next payload bit, change nothing.
- If the XOR does not match the payload bit, change one of the locations x_1, \dots, x_q . Communicate p additional bits by the choice of which to change.

The ZZW construction

Fix positive integer p . Split cover into groups of size $q = 2^p$.

For each group, suppose that the cover bits are x_1, \dots, x_q . Compute their XOR $x_1 \oplus \dots \oplus x_q$.

- If this matches next payload bit, change nothing.
(0 embedding changes, 1 bit of information transmitted)
- If the XOR does not match the payload bit, change one of the locations x_1, \dots, x_q . Communicate p additional bits by the choice of which to change.
(1 embedding change, $p + 1$ bits of information transmitted)

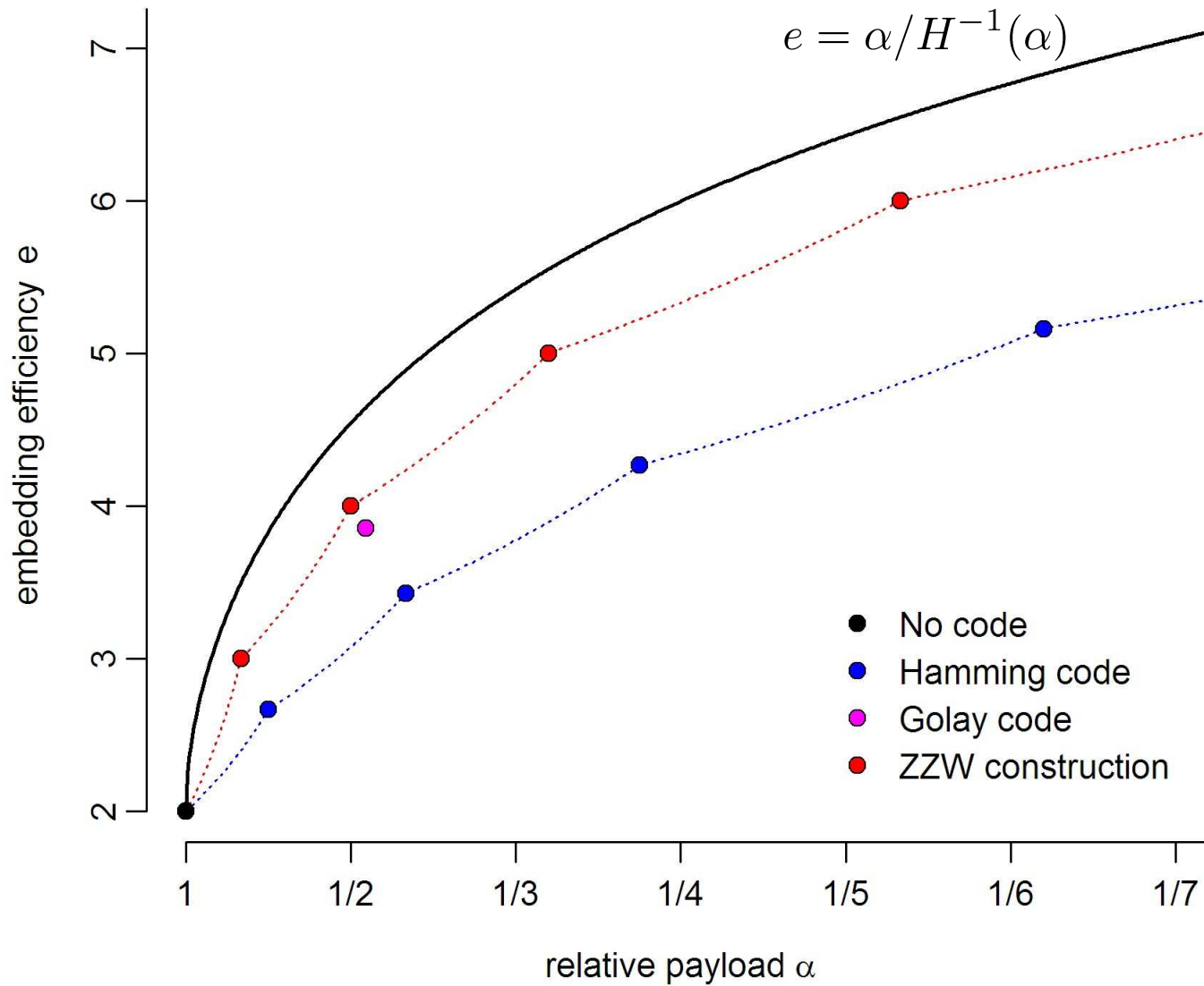
Clever bit: how does the recipient know

which groups had a change? - *Wet Paper codes*

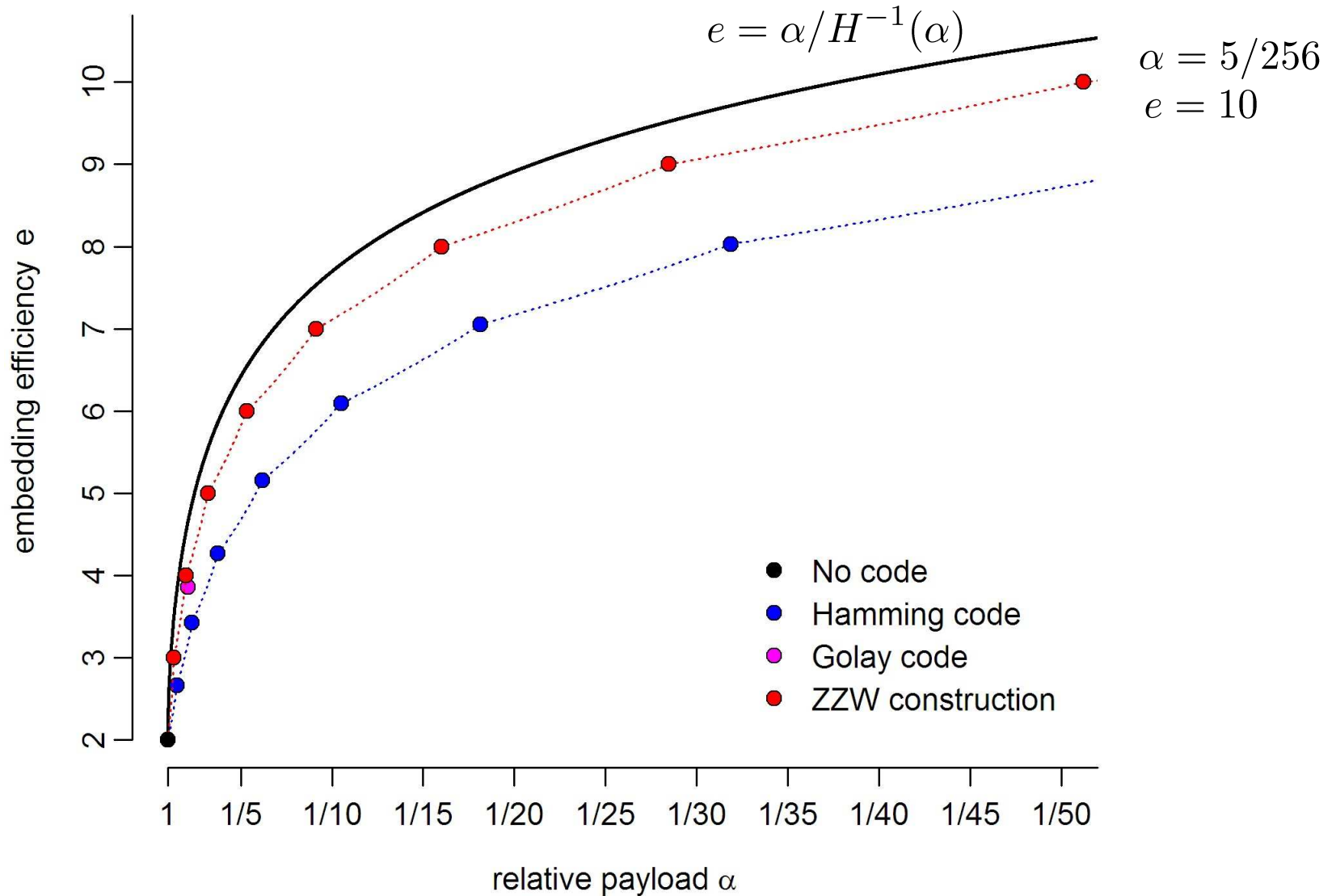
which of x_1, \dots, x_q was changed? - *Hamming codes*

On average, we have a $(\frac{1}{2}, 2^p, 1 + \frac{p}{2})$ embedding code.

Embedding efficiency



Embedding efficiency



Embedding 5120 bits in 262144 locations by
simple replacement: 2560 changes.



Embedding 5120 bits in 262144 locations by
ZZW code: 512 changes.



Asymptotic relationship

Since $e \sim \alpha/H^{-1}(\alpha)$, we can deduce an asymptotic relationship between cover size n , total number of embedding changes c , and total embedded payload p :

$$e \sim \alpha/H^{-1}(\alpha) \quad (\text{as } \alpha \rightarrow 0)$$

$$p/c \sim (p/n)/H^{-1}(p/n) \quad (\text{as } n \rightarrow \infty)$$

$$p/n \sim H(c/n)$$

$$p/n \sim (c/n) \log(c/n)$$

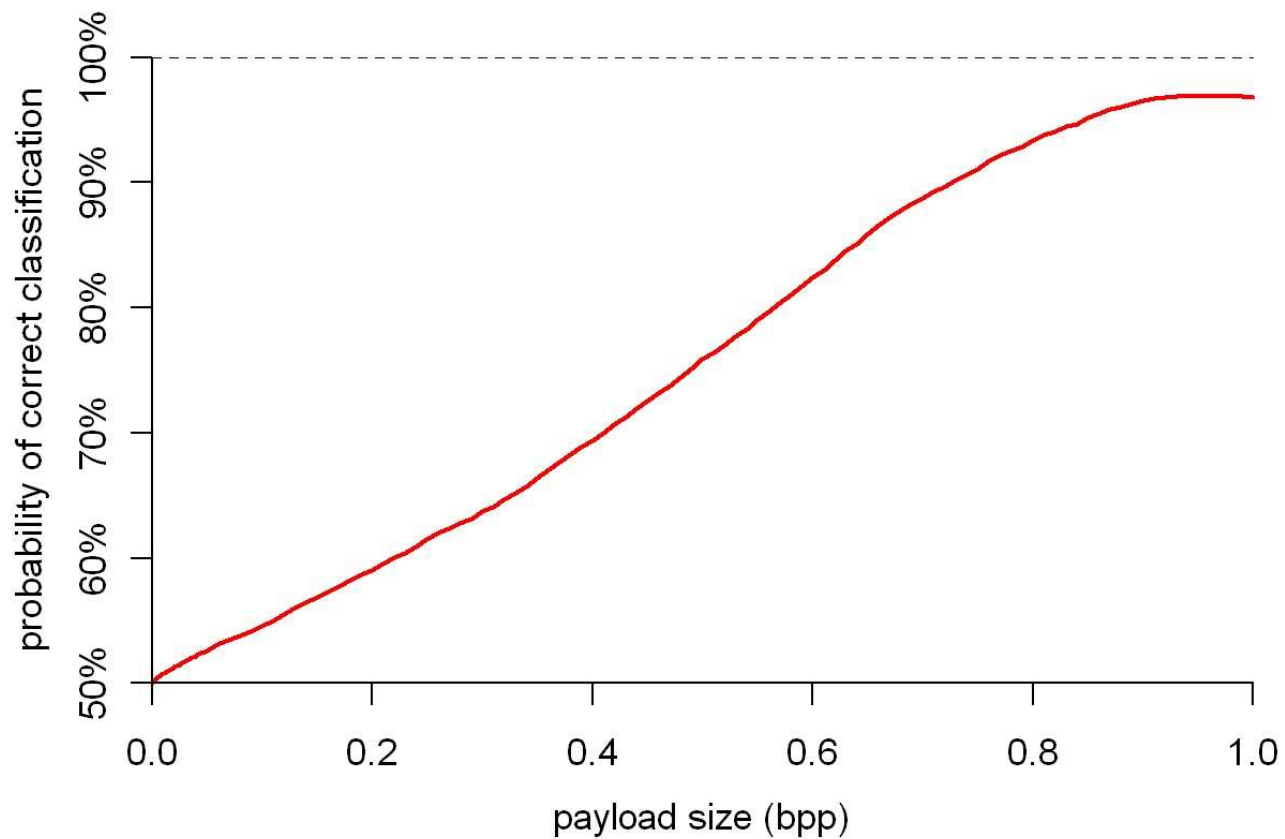
$$p \sim c \log(n/c)$$

Information Hiding and Covert Communication

Part 4: Steganographic Capacity

- Concept of capacity
- Capacity of perfect steganography
 - *The amount of information you can hide is proportional to the cover size*
- Capacity of batch steganography
 - *... proportional to the square root of the number of covers*
- Capacity of Markov covers
 - *... proportional to $\sqrt{n} \log n$*
- Reconciliation, experimental results

Detector performance



... the more information you hide, the greater the risk of discovery.

Capacity

- *the more information you hide, the greater the risk of discovery.*

Fix an embedding method & cover source.

- Given a limit on “risk”, what is the largest payload which can safely be embedded?
- How does this safe maximum payload depend on the size of the cover?

Perfect steganography

- *steganography with no risk.*

For there to be no risk, the cover objects and the stego objects must have exactly the same statistical properties.

Theorem

Randomly modulated codes constructed from codes with “order-1 security” produce perfect steganography.

The amount of information which can be hidden using these randomly modulated codes depends on the cover source, but is guaranteed to be $O(n)$, where n is the number of symbols transmitted.

Moral

Given perfect knowledge of the cover source, the amount of information you can hide is proportional to the cover size.

but... constructing the perfect embedding code requires complete and perfect knowledge of the cover source.

Nobody has perfect knowledge of their cover source!

“Model-based steganography” by Sallee, 2003

... broken by Fridrich in 2004

“Stochastic QIM” by Moulin & Briassouli, 2004

(described as “robust” and “undetectable”)

... broken by Wang & Moulin in 2006

Capacity

- *the more information you hide, the greater the risk of discovery.*

Fix an embedding method & cover source.

- Given a limit on “risk”, what is the largest payload which can safely be embedded?
- How does this safe maximum payload depend on the size of the cover?

Ross Anderson in the 1st Information Hiding Workshop (1996):

“...the more covertext we give the warden [detector], the better he will be able to estimate its statistics, and so the smaller the rate at which Alice will be able to tweak bits safely. The rate might even tend to zero...”

Capacity

- the more information you hide, the greater the risk of discovery.

Fix an embedding method.

- Given a limit on “risk”, what is the largest payload which can safely be embedded?
- How does this safe maximum payload depend on the size of the cover?

... implicitly assumes that the Warden seeks a binary decision. “Safe” can be defined in terms of false positive rate α and false negative rate β :

embedding a certain payload is “safe” if any steganalysis detector must have more than a certain false positive and false negative rate.

Information theoretic bounds

- *bounds detection performance using Kullback-Leibler divergence.*

If X has density function f , and Y has density function g , then the KL divergence from X to Y is

$$D_{\text{KL}}(X, Y) = - \int f(x) \log \frac{g(x)}{f(x)} dx$$

Information Processing Theorem: $D_{\text{KL}}(h(X), h(Y)) \leq D_{\text{KL}}(X, Y)$

Therefore, if trying to separate an instance of X from one of Y , the false positive (Y) rate α and false negative (X) rate β must satisfy

$$\alpha \log \frac{\alpha}{1 - \beta} + (1 - \alpha) \log \frac{1 - \alpha}{\beta} \leq D_{\text{KL}}(X, Y)$$

Information theoretic bounds

- *bounds detection performance using Kullback-Leibler divergence.*

If X has density function f , and Y has density function g , then the KL divergence from X to Y is

$$D_{\text{KL}}(X, Y) = - \int f(x) \log \frac{g(x)}{f(x)} dx$$

Information Processing Theorem: $D_{\text{KL}}(h(X), h(Y)) \leq D_{\text{KL}}(X, Y)$

Therefore, if trying to separate an instance of X from one of Y , the false positive (Y) rate α and false negative (X) rate β must satisfy

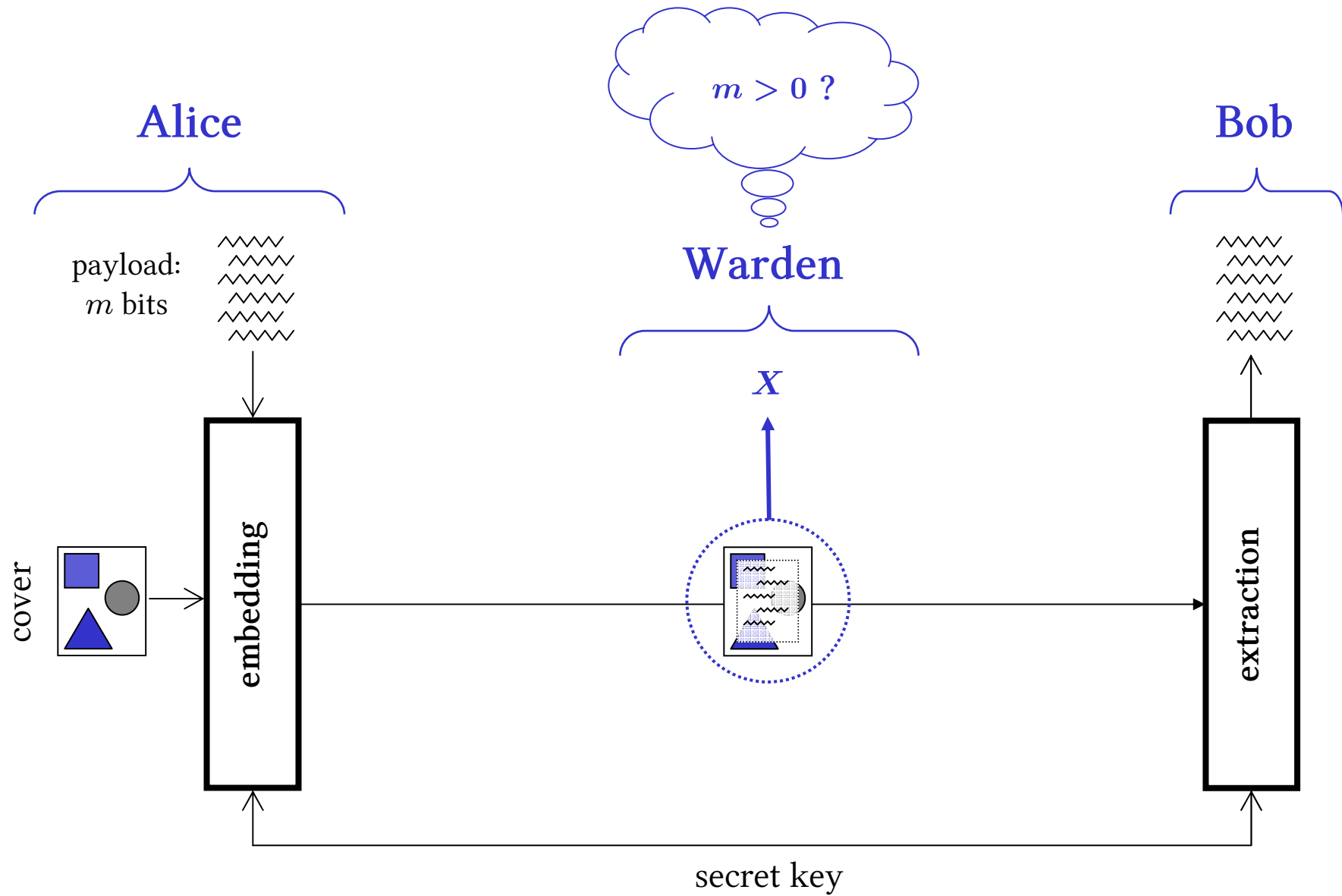
$$\alpha \log \frac{\alpha}{1 - \beta} + (1 - \alpha) \log \frac{1 - \alpha}{\beta} \leq D_{\text{KL}}(X, Y)$$

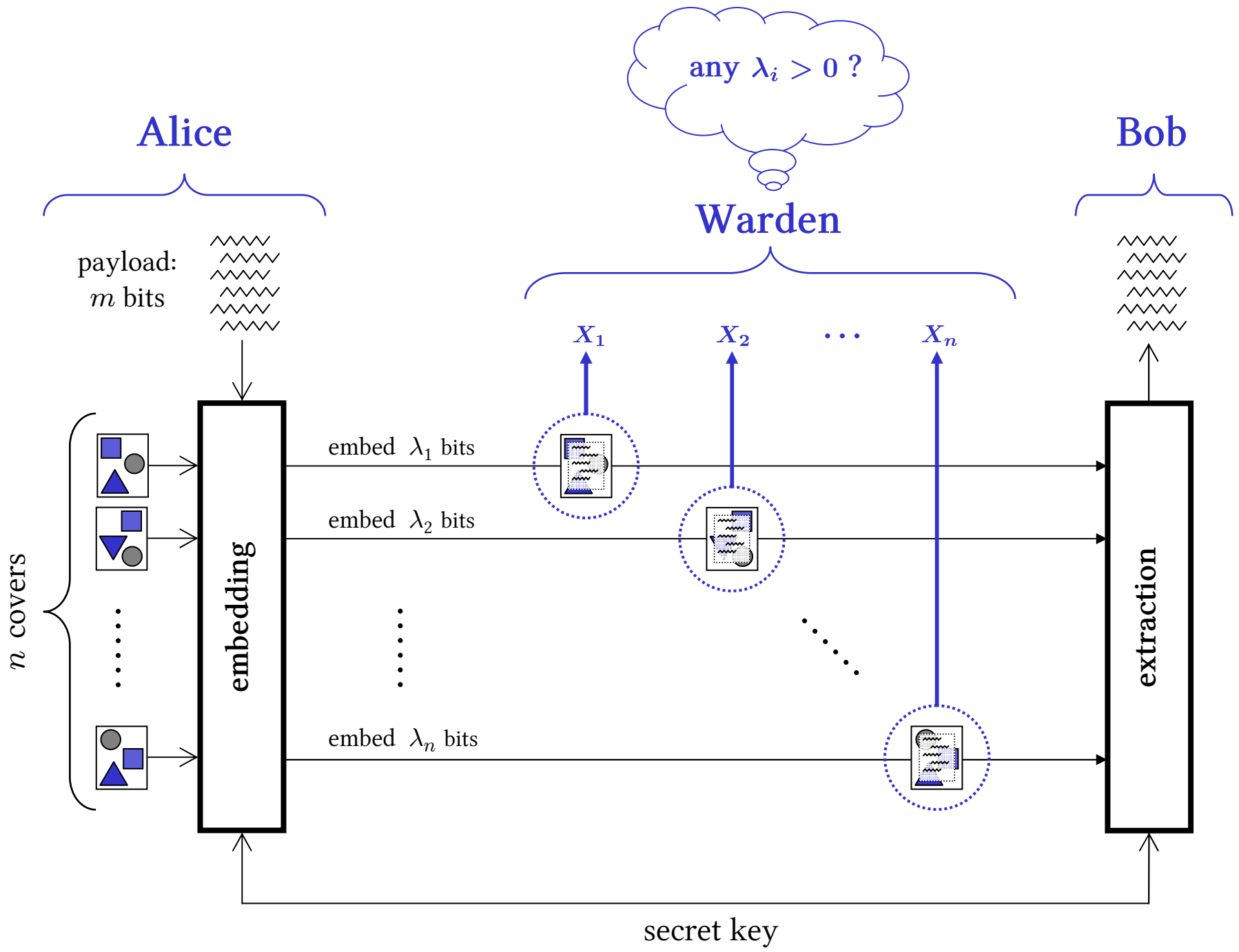
To apply this, we need to know $D_{\text{KL}}(\text{cover objects}, \text{stego objects})$.

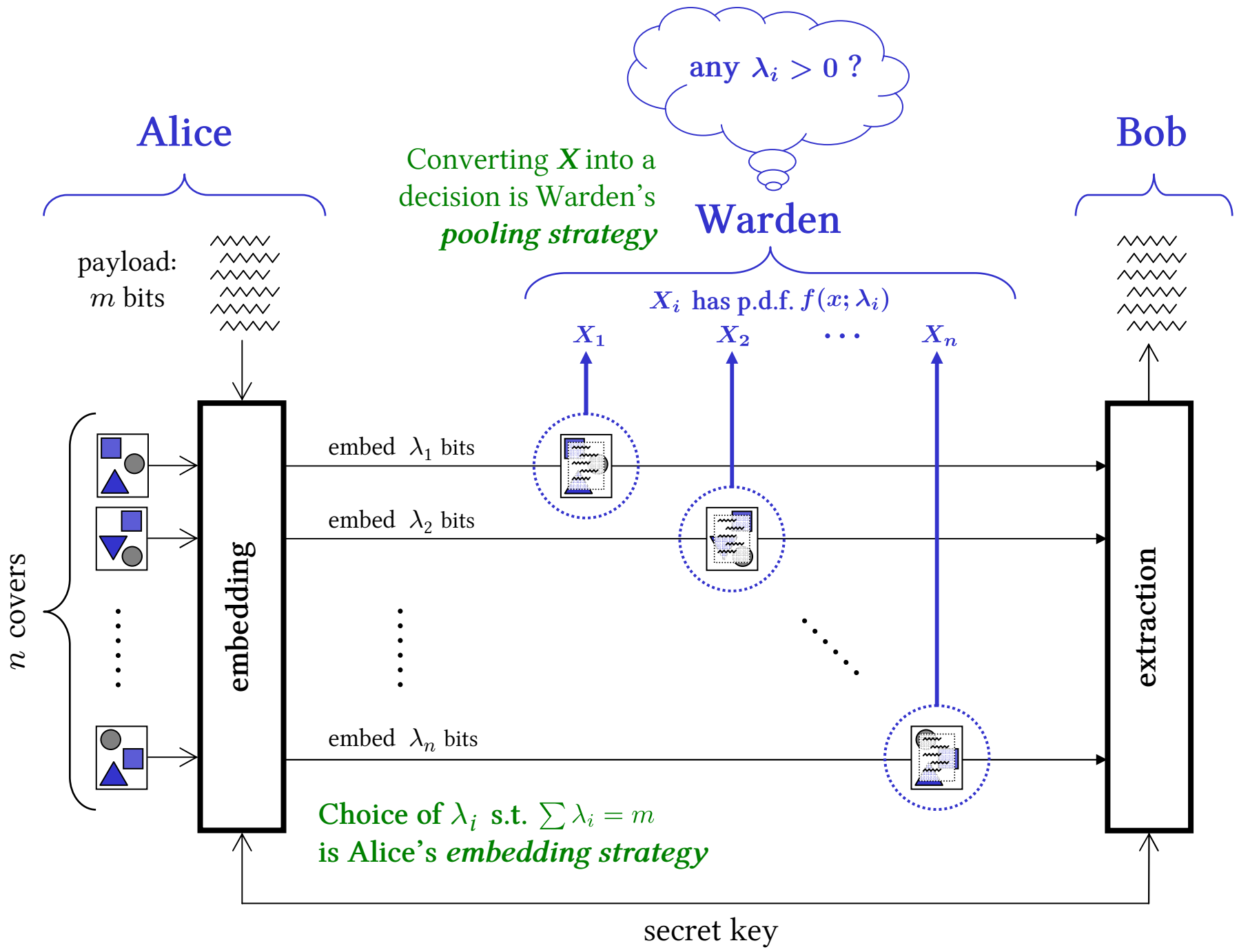
We have already seen that statistical models for covers are inaccurate.

Batch steganography

- *instead of performing steganography and steganalysis in isolated objects, consider a large number of covers.*







Batch steganographic capacity

A tractable and interesting capacity question:

How does maximum secure capacity depend on the number of covers n ?

It turns out that the order of growth is independent of all the other capacity factors.

Theorem

- Assuming
- Alice's covers are of bounded size,
 - X_i independent,
 - $\exists \mathcal{Y}$ with positive measure such that $\frac{\partial f}{\partial \lambda}(y; 0) > 0$ on \mathcal{Y} ,
 - $\int \frac{\partial^2 \log f}{\partial \lambda^2}(x; 0) f(x; 0) dx < \infty$.

No matter what the acceptable false positive rate α and false negative rate β ,

1. There is a pooling strategy for the Warden such that, no matter what Alice's embedding strategy, if $m/\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is unacceptable.
2. There is an embedding strategy for Alice such that, no matter what the Warden's pooling strategy, if $m/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is acceptable.

Proof sketch

1. There is a pooling strategy for the Warden such that... if $m/\sqrt{n} \rightarrow \infty$ then... Alice [will be eventually caught].

The pooling strategy is:

*payload detected if $P = |\{X_i \mid X_i \in \mathcal{Y}\}|$ is greater than a critical threshold P^**

By the Berry-Esséen Theorem (& independence):

$$P \sim N(\mu, \sigma^2)$$

$$\begin{aligned}\mu &= \sum_i \int_{\mathcal{Y}} f(x; \lambda_i) dx \\ &= n \int_{\mathcal{Y}} f(x; 0) dx + \sum_i \lambda_i \int_{\mathcal{Y}} \frac{\partial f}{\partial \lambda}(x; \hat{\lambda}_i) dx \\ &= cn + \Theta(m)\end{aligned}$$

$$\begin{aligned}\sigma^2 &= \sum_i \int_{\mathcal{Y}} f(x; \lambda_i) dx \int_{\bar{\mathcal{Y}}} f(x; \lambda_i) dx \\ &= \Theta(n)\end{aligned}$$

Detector performance tends to perfect, as $n \rightarrow \infty$

Proof sketch

2. There is an embedding strategy for Alice such that... if $m/\sqrt{n} \rightarrow 0$ then... Alice [will not be caught].

The embedding strategy is:

spread the payload m equally amongst the n covers: $\lambda_i = m/n$ for all i

By the information processing theorem, *any* pooling strategy has false positive rate α and false negative rate β satisfying

$$\begin{aligned} \alpha \log \frac{\alpha}{1-\beta} + (1 - \alpha) \log \frac{1-\alpha}{\beta} &\leq D_{\text{KL}}(\mathbf{X} \mid \text{no embedding}, \mathbf{X} \mid \text{embedding}) \\ &= \sum_{i=1}^n D_{\text{KL}}(X_i \mid \lambda = 0, X_i \mid \lambda = \lambda_i) \\ &= n D_{\text{KL}}(X_i \mid \lambda = 0, X_i \mid \lambda = \frac{m}{n}) \end{aligned}$$

Proof sketch

$$\begin{aligned} \alpha \log \frac{\alpha}{1-\beta} + (1-\alpha) \log \frac{1-\alpha}{\beta} &\leq n D_{\text{KL}}(X_i | \lambda = 0, X_i | \lambda = \frac{m}{n}) \\ &= -n \int f(x; 0) \log \frac{f(x; \frac{m}{n})}{f(x; 0)} dx \\ &= -n \int f(x; 0) [\log f(x; \frac{m}{n}) - \log f(x; 0)] dx \\ &= -n \int f(x; 0) \left[\frac{m}{n} \frac{\partial f}{\partial \lambda}(x; \frac{m}{n}) / f(x; \frac{m}{n}) + \frac{(\frac{m}{n})^2}{2} \frac{\partial^2 \log f}{\partial \lambda^2}(x; \hat{\lambda}) \right] dx \\ &\quad \text{where each } \hat{\lambda} \in (0, \frac{m}{n}) \\ &\rightarrow -m \frac{\partial}{\partial \lambda} \int f(x; \frac{m}{n}) dx + \frac{m^2}{2n} \int \frac{\partial^2 \log f}{\partial \lambda^2}(x; 0) f(x; 0) dx \\ &\rightarrow 0 + 0 \end{aligned}$$

Detector performance tends to random, as $n \rightarrow \infty$

Theorem

Assuming

- Alice's covers are of bounded size,
- X_i independent,
- $\exists \mathcal{Y}$ with positive measure such that $\frac{\partial f}{\partial \lambda}(y; 0) > 0$ on \mathcal{Y} ,
- $\int \frac{\partial^2 \log f}{\partial \lambda^2}(x; 0) f(x; 0) dx < \infty$. — *Finite Fisher Information*

Common sense

*Embedding
makes something
more (or less)
likely*

No matter what the acceptable false positive rate α and false negative rate β ,

1. There is a pooling strategy for the Warden such that, no matter what Alice's embedding strategy, if $m/\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is unacceptable.
2. There is an embedding strategy for Alice such that, no matter what the Warden's pooling strategy, if $m/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is acceptable.

Theorem

Assuming

- Alice's covers are of bounded size,
- X_i independent,
- $\exists \mathcal{Y}$ with positive measure such that $\frac{\partial f}{\partial \lambda}(y; 0) > 0$ on \mathcal{Y} ,
- $\int \frac{\partial^2 \log f}{\partial \lambda^2}(x; 0) f(x; 0) dx < \infty$. — *Finite Fisher Information*

Common sense

*Embedding
makes something
more (or less)
likely*

No matter what the acceptable false positive rate α and false negative rate β ,

1. There is a pooling strategy for the Warden such that, no matter what Alice's embedding strategy, if $m/\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is unacceptable.
2. There is an embedding strategy for Alice such that, no matter what the Warden's pooling strategy, if $m/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is acceptable.
3. What if $m/\sqrt{n} \rightarrow c$?

Moral

Given some conditions on the covers, the amount of information you can hide is proportional to the square root of the number of covers.

but... what about individual covers?

Steganography in Markov chains

- *turn to single cover objects, with a very general model.*

- Model the cover source as a Markov chain, a memoryless finite state machine moving at random from state to state.

The sequence of states is denoted (X_1, X_2, \dots, X_n) .

The states could represent the pixels of a cover image, or the quantized DCT coefficients, or groups of pixels, etc.

- We model the embedding process as random substitution of states described by a matrix (c_{xy}) : if the stego object is (Y_1, Y_2, \dots, Y_n) and the *embedding change rate* is γ ,

$$P(Y_i = y | X_i = x) = (1 - \gamma)\delta_{xy} + \gamma c_{xy}.$$

Steganography in Markov chains

- *turn to single cover objects, with a very general model.*

- Model the cover source as a Markov chain, a memoryless finite state machine moving at random from state to state.

The sequence of states is denoted (X_1, X_2, \dots, X_n) .

The states could represent the pixels of a cover image, or the quantized DCT coefficients, or groups of pixels, etc.

- We model the embedding process as random substitution of states described by a matrix (c_{xy}) : if the stego object is (Y_1, Y_2, \dots, Y_n) and the *embedding change rate* is γ ,

$$P(Y_i = y | X_i = x) = (1 - \gamma)\delta_{xy} + \gamma c_{xy}.$$

(We must have $\sum_x c_{xy} = 1$ for all x . If also $\sum_y c_{xy} = 1$ for all y , the embedding is called **doubly stochastic**.)

Theorem

- Assuming
- Markov chain is irreducible,
 - Embedding is independent of cover,
 - Embedding is doubly stochastic,
 - There is some sequence (y_1, \dots, y_m) such that $\frac{\partial}{\partial \gamma} P((Y_1, \dots, Y_m) = (y_1, \dots, y_m)) > 0$.

No matter what the acceptable false positive rate α and false negative rate β ,

1. There exists a detector such that, if the embedding rate γ satisfies $\gamma\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is unacceptable.
2. For any detector, if the number of embedding rate γ satisfies $\gamma\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is acceptable.

Theorem

Assuming

- Markov chain is irreducible,
- Embedding is independent of cover,
- Embedding is doubly stochastic,
- There is some sequence (y_1, \dots, y_m) such that $\frac{\partial}{\partial \gamma} P((Y_1, \dots, Y_m) = (y_1, \dots, y_m)) > 0$.

Ensures convergence to equilibrium

Common sense

Embedding makes something more (or less) likely

No matter what the acceptable false positive rate α and false negative rate β ,

embedding changes $c/\sqrt{n} \rightarrow \infty$

1. There exists a detector such that, if the embedding rate γ satisfies $\gamma\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is unacceptable.

$c/\sqrt{n} \rightarrow 0$

2. For any detector, if the number of embedding rate γ satisfies $\gamma\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ then for sufficiently large n Alice's risk is acceptable.

Moral

Given a Markov cover source and random embedding satisfying some weak conditions, the number of embedding changes you can make is proportional to the square root of the cover size.

Recall, with asymptotically efficient embedding codes, the payload size p and the number of embedding changes c can satisfy $p \sim c \log(n/c)$.

Then $c = O(\sqrt{n})$ implies $p = O(\sqrt{n} \log n)$.

Reconciliation

Moulin's theorem says:

Steganographic capacity is linear,
if Alice knows her cover source perfectly.

Batch steganographic theorem says:

Steganographic capacity is a square root law,
if (roughly) Alice's embedding is imperfect (and non-adaptive).

Markov chain capacity theorem says:

Steganographic capacity measured in embedding changes is a square root law,
if (roughly) Alice's embedding is imperfect (and non-adaptive).

Experimental results

We would like to test some contemporary steganography and steganalysis methods, to see if a square root law is observed.

Idea:

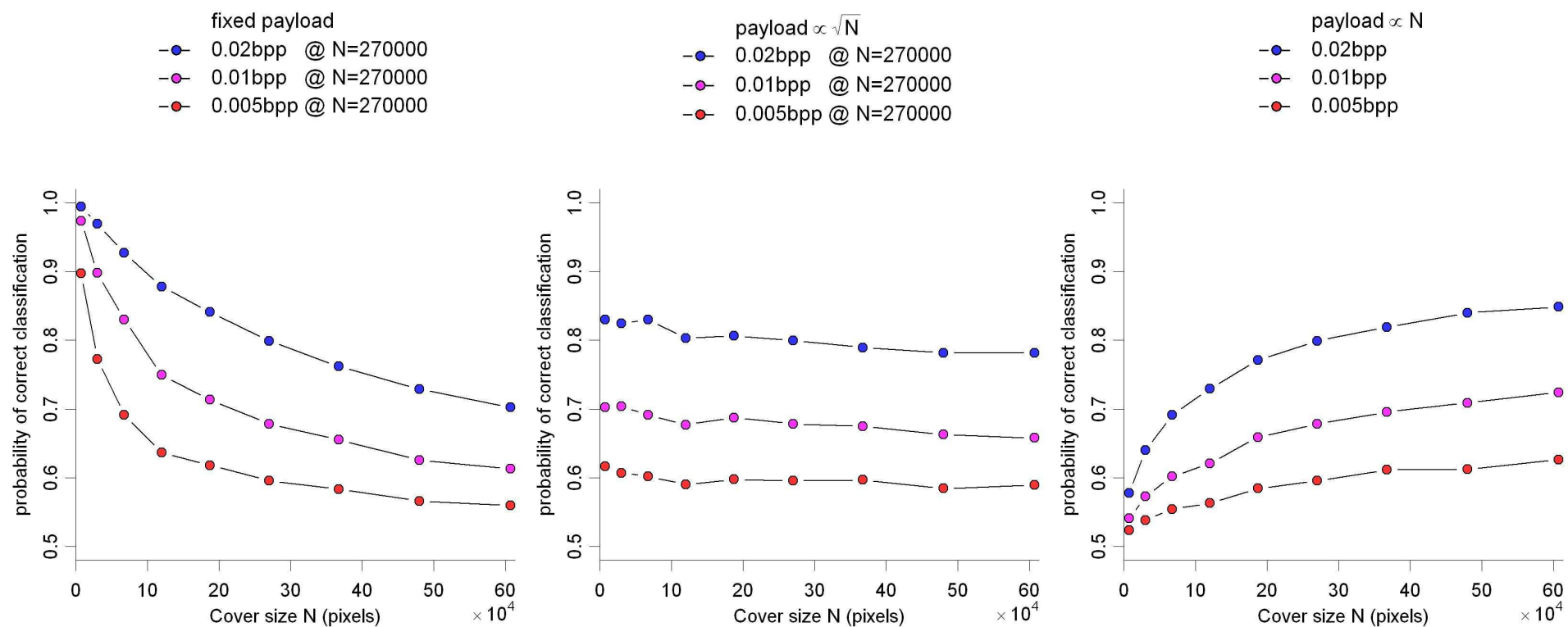
- Create image sets of different sizes.
- Embed lots of different-size payloads in each set, perform steganalysis.
- Look for the relationship between cover and payload size and detectability.

It is difficult to make image sets of different sizes which do not also have different characteristics: noise, density, etc.

The best we can do is **crop down** large images to smaller ones, choosing the crop region to preserve local variance or a similar noise statistic.

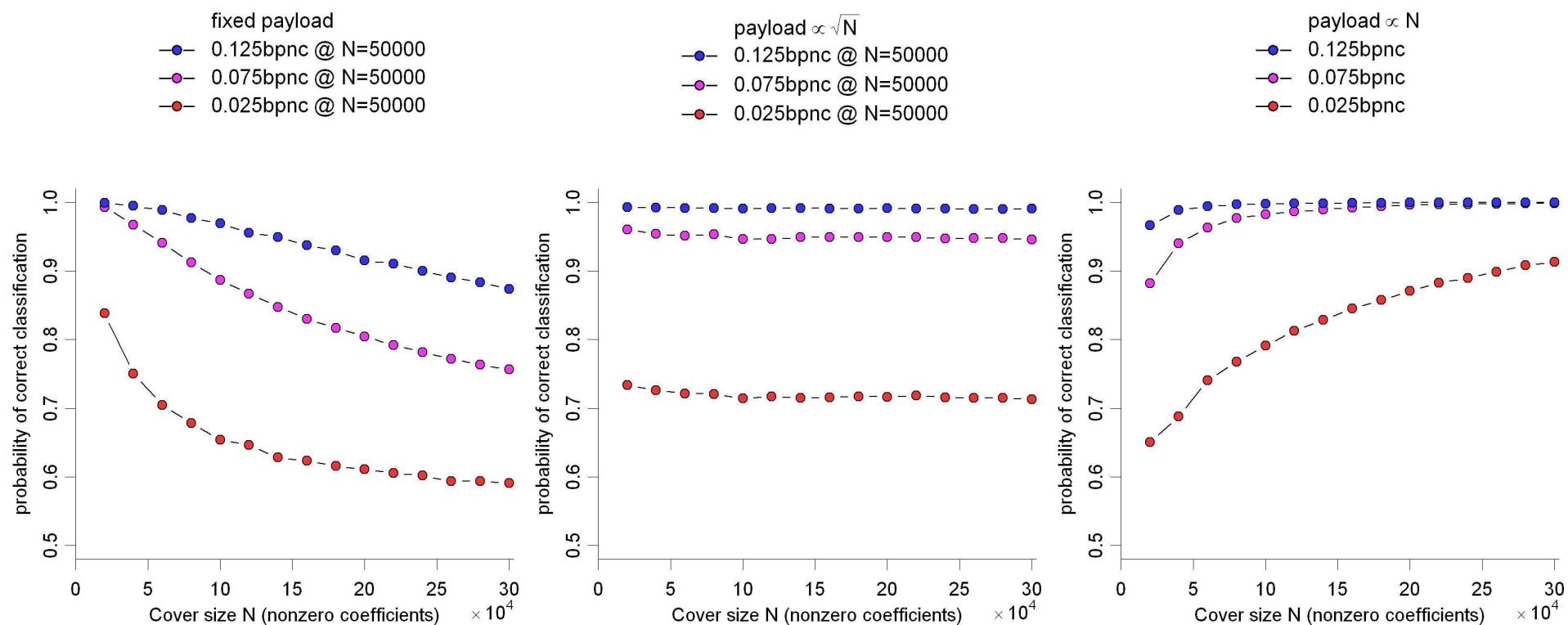
Experimental results

LSB replacement steganography & Couples steganalysis



Experimental results

F5 steganography & “Merged features” SVM steganalysis



Conclusions

- The complete square root law for capacity will be a suite of theorems, proving the result under different circumstances.

We already know some conditions under which it holds, and it appears to hold in practice too.

- Although many authors describe steganography and steganalysis payloads in terms of “bits per pixel” or “bits per nonzero DCT coefficient”, it is more accurate to think in terms of “bits per square root pixel,” etc.
- A big outstanding question: what is the multiplicative constant in the square root law?

It will depend on the cover source and the embedding method:

- *could use this to compare the security of different embedders,*
- *could also compare the security of different cover sources.*