

# Revisiting Weighted Stego-Image Steganalysis



**Andrew Ker**

adk@comlab.ox.ac.uk

*Oxford University Computing Laboratory*



**Rainer Böhme**

rainer.boehme@inf.tu-dresden.de

*Technische Universität Dresden,  
Institute for System Architecture*

SPIE/IS&T Electronic Imaging, San Jose, CA

28 January 2008

# Revisiting Weighted Stego-Image Steganalysis

## Outline

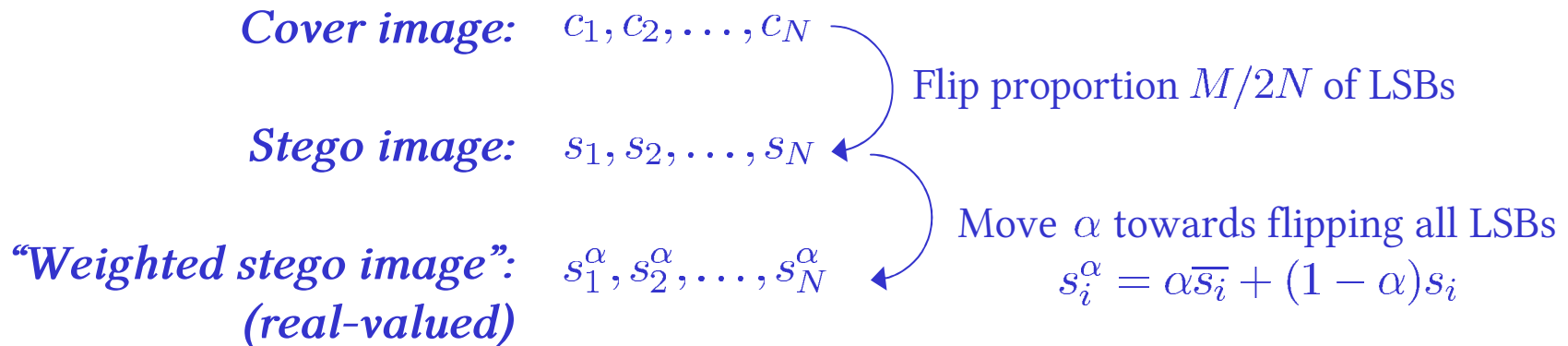
- *The Weighted Stego Image (WS) method*
- *Performance*
- *Re-engineering WS*
- *Performance*
- *WS for sequential embedding*
- *Performance*

# The WS Method

Imagine a single-channel cover image with  $N$  pixels, and a payload of  $M$  bits (possibly zero) inserted by overwriting a selection of LSBs.

*WS steganalysis estimates the (proportionate) payload size  $p = \frac{M}{N}$ .*

# The WS Method



**Theorem [Fridrich & Goljan, 2004]**

The function  $E(\alpha) = \sum_{i=1}^N w_i (s_i^\alpha - c_i)^2$  is minimized at  $\alpha = M/2N$ ,

where the  $w_i$  are a vector of weights.

# The WS Method

## Theorem

The function  $E(\alpha) = \sum_{i=1}^N w_i (s_i^\alpha - c_i)^2$  is minimized at  $\alpha = M/2N$ .

## WS Steganalysis

1. Estimate cover by filtering the stego image.

$$\hat{c}_i = \text{Average of the four stego pixels neighbouring } s_i$$

2. Decide on a weight vector.

$$w_i = \frac{1}{1 + \sigma_i^2} \quad \sigma_i^2 \text{ is the local variance of the four stego pixels neighbouring } s_i$$

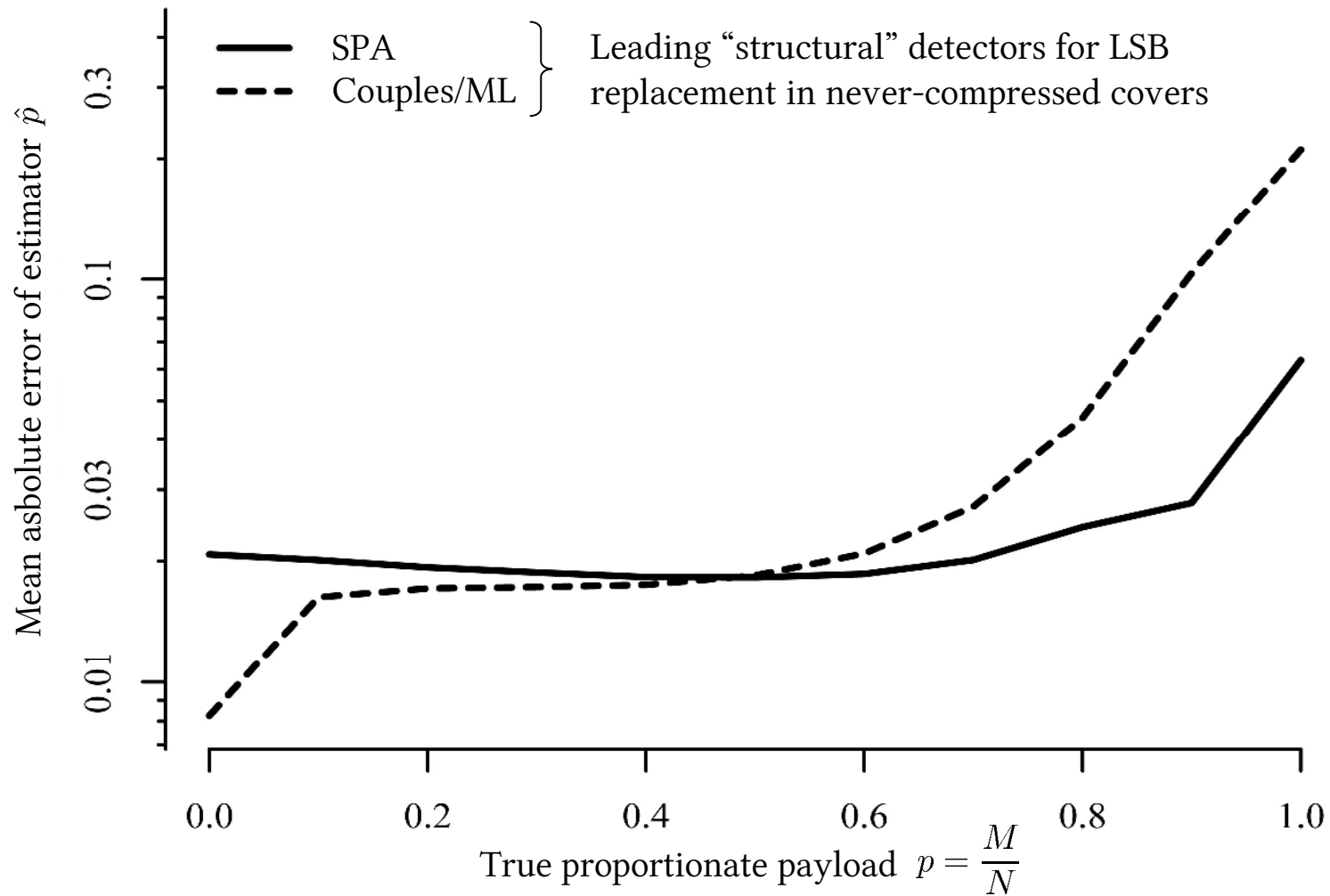
3. Compute “flat-pixel correction”.

$$r = - \text{estimate of bias introduced by flat areas in cover image}$$

Estimate proportionate payload size

$$\hat{p} = r + 2 \operatorname{argmin}_{\alpha} \sum_{i=1}^N w_i (s_i^\alpha - \hat{c}_i)^2 = r + \frac{2}{N} \sum_{i=1}^N w_i (s_i - \hat{c}_i)(s_i - \bar{s}_i).$$

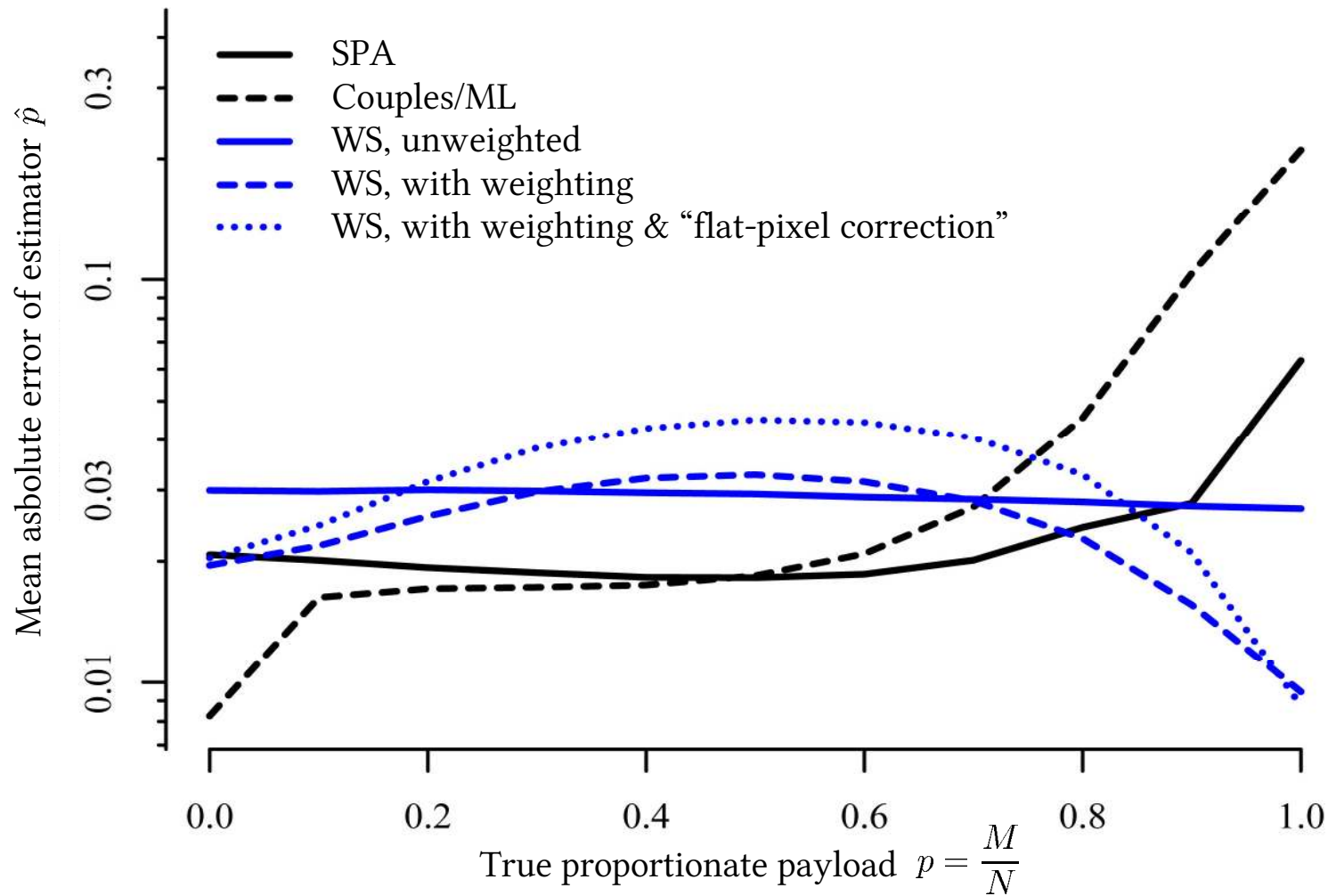
# Performance



Cover source:

3000 grayscale scanned images resampled to 0.3Mpixels

# Performance



Cover source:

3000 grayscale scanned images resampled to 0.3Mpixels

# Adaptive Cover Predictors

- ▶ Estimate cover by filtering the stego image.

$$\hat{c}_i = \text{Average of the four stego pixels neighbouring } s_i$$



# Adaptive Cover Predictors

- ▶ Estimate cover by filtering the stego image.

$$\hat{\mathbf{c}} = \mathbf{s} * \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix}$$

But what about other filters?

$$\hat{\mathbf{c}} = \mathbf{s} * \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{pmatrix}$$

# Adaptive Cover Predictors

- ▶ Estimate cover by filtering the stego image.

$$\hat{c} = s * \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix}$$

But what about other filters?

$$\hat{c} = s * \begin{pmatrix} -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix}$$

$$\hat{c} = s * \begin{pmatrix} e & d & c & d & e \\ d & b & a & b & d \\ c & a & 0 & a & c \\ d & b & a & b & d \\ e & d & c & d & e \end{pmatrix}$$

# Adaptive Cover Predictors

- ▶ Estimate cover by filtering the stego image.

Select a filter pattern

$$F = \begin{pmatrix} e & d & c & d & e \\ d & b & a & b & d \\ c & a & 0 & a & c \\ d & b & a & b & d \\ e & d & c & d & e \end{pmatrix}$$

and find the values of  $a\dots e$  to best predict the stego object by itself, i.e. find

$$\underset{F}{\operatorname{argmin}} \|s - F * s\|.$$

⇒ improves cover pixel & payload size estimation accuracy.

# Moderated Weights

- ▶ Decide on a weight vector.

$$w_i = \frac{1}{1 + \sigma_i^2}$$

$\sigma_i^2$  is the local variance of the four stego pixels neighbouring  $s_i$

Our experiments suggested that the weights are too extreme and should be moderated.

$$w_i = \frac{1}{5 + \sigma_i^2}$$

$\sigma_i^2$  is the weighted variance of the neighbouring stego pixels affecting  $s_i$  in the prediction filter

⇒ improves payload size estimation accuracy.

# Bias Correction

- ▶ Correct bias.

The “flat-pixel correction” in [Fridrich & Goljan, EI 2004], doesn’t work very well. A better estimate can be given if we model the cover image by

$$\begin{array}{l} c_1, c_2, \dots, c_N \\ s_1, s_2, \dots, s_N \end{array} \left. \vphantom{\begin{array}{l} c_1, c_2, \dots, c_N \\ s_1, s_2, \dots, s_N \end{array}} \right\} \text{Flip proportion } M/2N \text{ of LSBs}$$

Then

$$\begin{aligned} \mathbb{E}[\hat{p}] &= \frac{2}{N} \mathbb{E} \left[ \sum w_i (s_i - \hat{c}_i) (s_i - \bar{s}_i) \right] \\ &= \dots \\ &= p + p \sum w_i (s_i - \bar{s}_i) (F * (\bar{\mathbf{s}} - \mathbf{s}))_i \end{aligned}$$

⇒ improves payload size estimation accuracy.

# Re-engineered WS

## Theorem

The function  $E(\alpha) = \sum_{i=1}^N w_i (s_i^\alpha - c_i)^2$  is minimized at  $\alpha = M/2N$ .

## WS Steganalysis

1. Estimate cover by filtering the stego image.

Find  $F$  to minimize  $\|\mathbf{s} - F * \mathbf{s}\|$ ,  
then  $\hat{\mathbf{c}} = F * \mathbf{s}$

2. Decide on a weight vector.

$w_i = \frac{1}{5 + \sigma_i^2}$   $\sigma_i^2$  is the local variance of the neighbouring stego pixels affecting  $s_i$  in the prediction filter

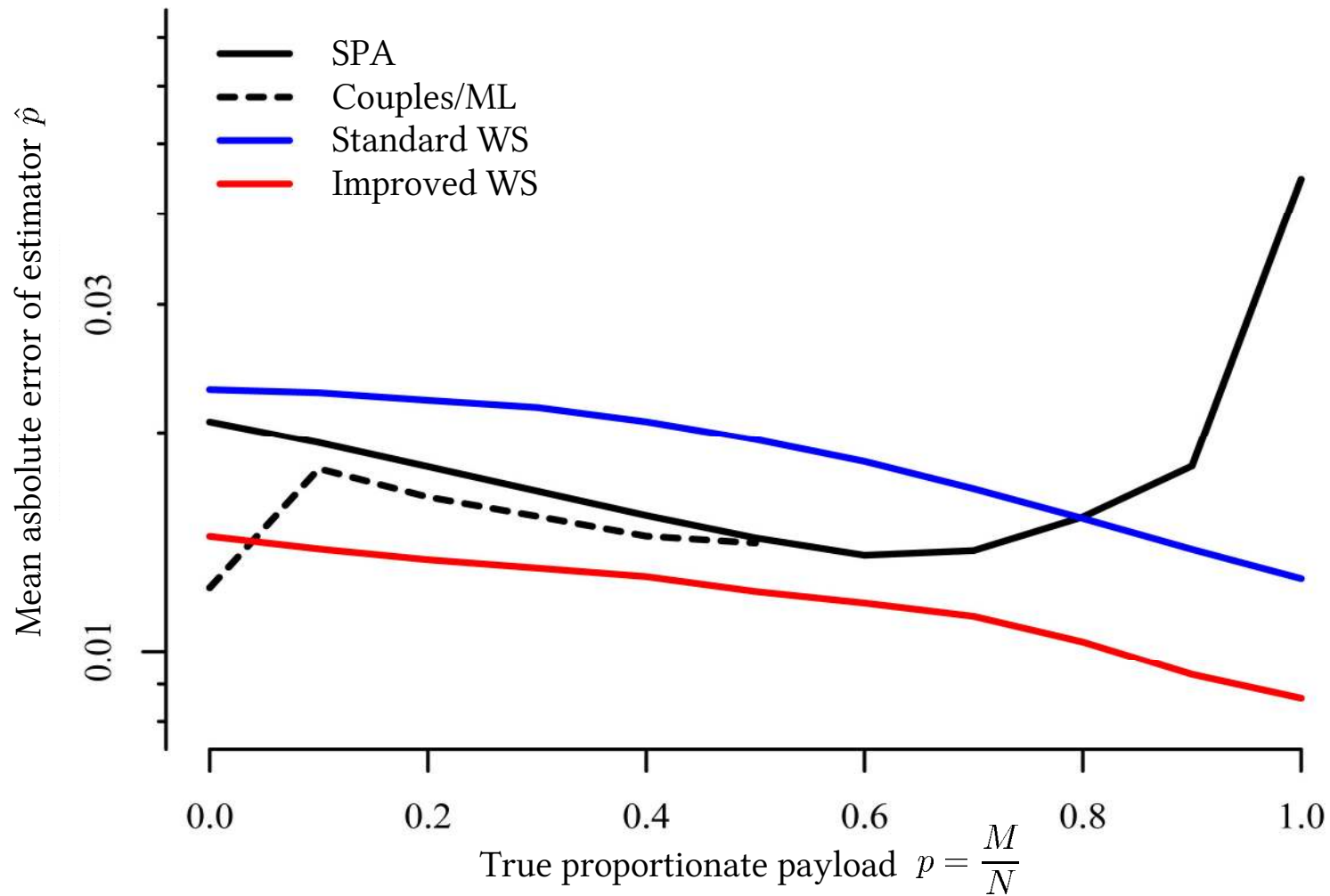
3. Compute bias correction.

$$r = -p \sum w_i (s_i - \bar{s}_i) (F * (\bar{\mathbf{s}} - \mathbf{s}))_i$$

Estimate proportionate payload size

$$\hat{p} = r + 2 \operatorname{argmin}_{\alpha} \sum_{i=1}^N w_i (s_i^\alpha - \hat{c}_i)^2 = r + \frac{2}{N} \sum_{i=1}^N w_i (s_i - \hat{c}_i) (s_i - \bar{s}_i).$$

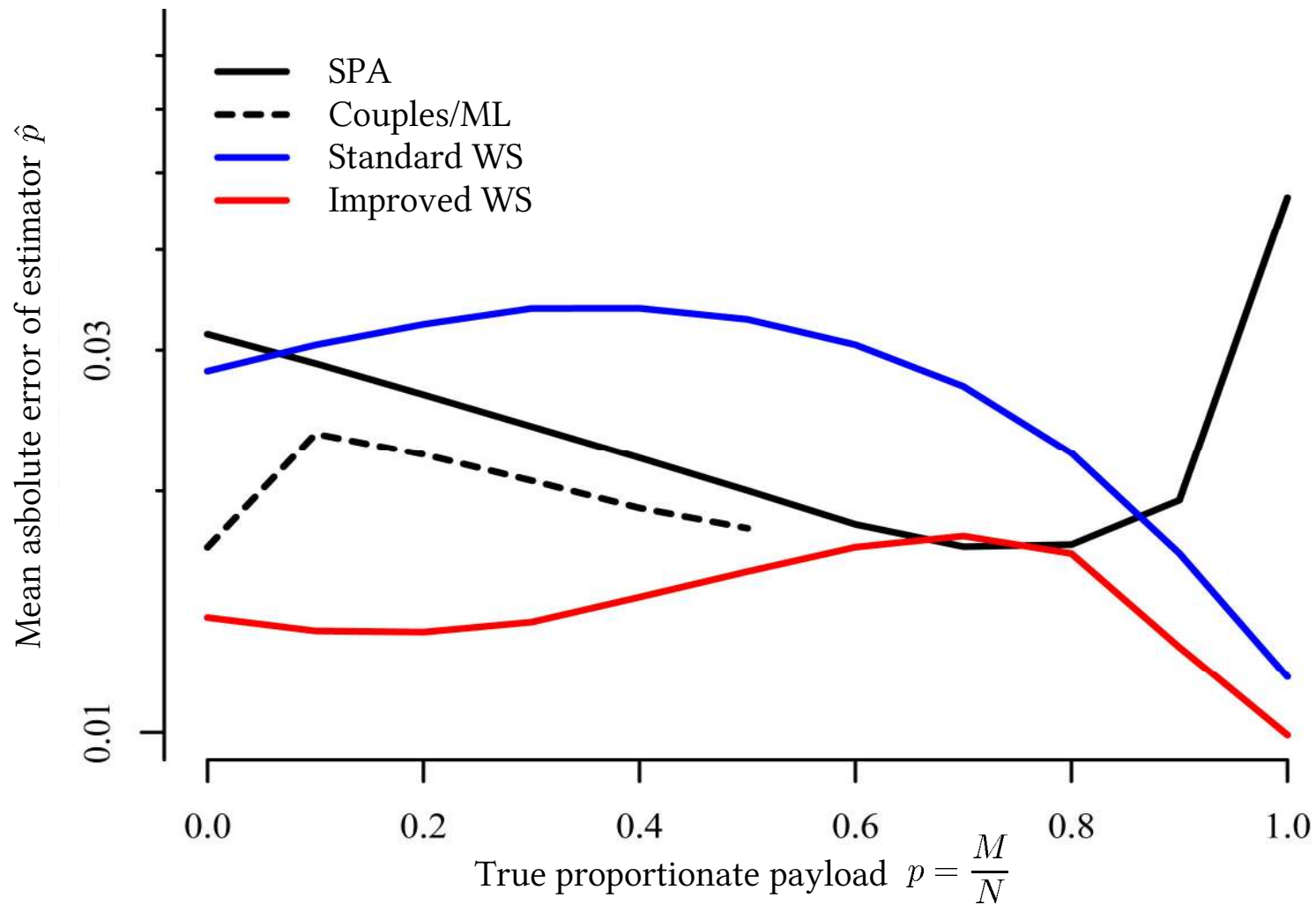
# Performance



Cover source:

1600 grayscale RAW digital camera images cropped to 0.3Mpixels

# Performance

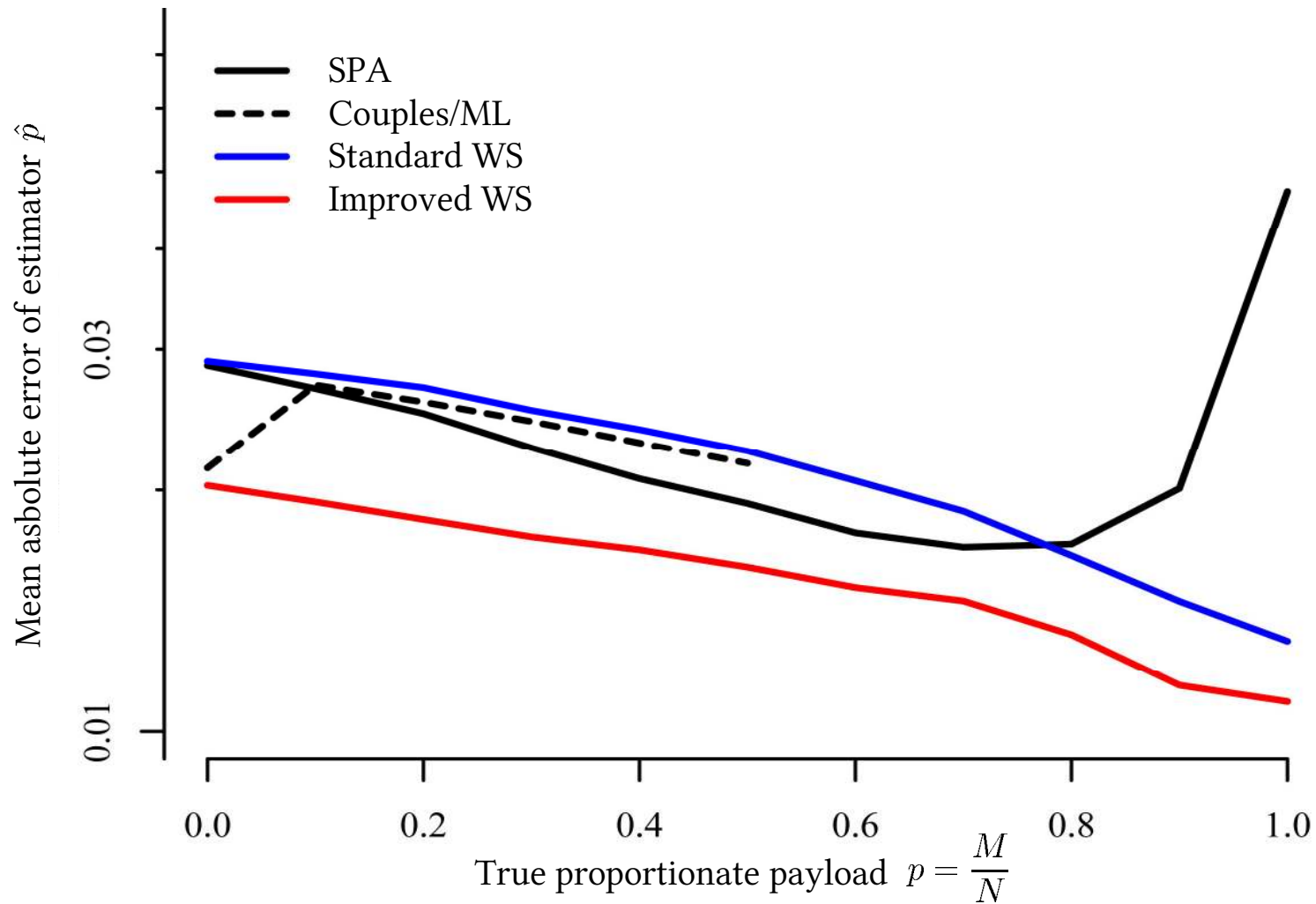


Cover source:

1600 grayscale RAW digital camera images resampled to 0.3Mpixels



# Performance



Cover source:

3000 grayscale scanned images resampled to 0.3Mpixels

# WS For Sequential Payload

*Cover image:*  $c_1, c_2, \dots, c_N$

*Stego image:*  $s_1, s_2, \dots, s_N$

*Weighted stego image:*  $s_1^j, s_2^j, \dots, s_N^j$

Flip first  $M$  LSBs with probability  $1/2$

Go halfway to flipping first  $j$  LSBs

$$s_i^j = \begin{cases} \frac{1}{2}\overline{s_i} + \frac{1}{2}s_i, & i \leq j \\ s_i, & i > j \end{cases}$$

## Theorem

The function  $F(j) = \sum_{i=1}^N (s_i^j - c_i)^2$  is minimized at  $j = M$ .

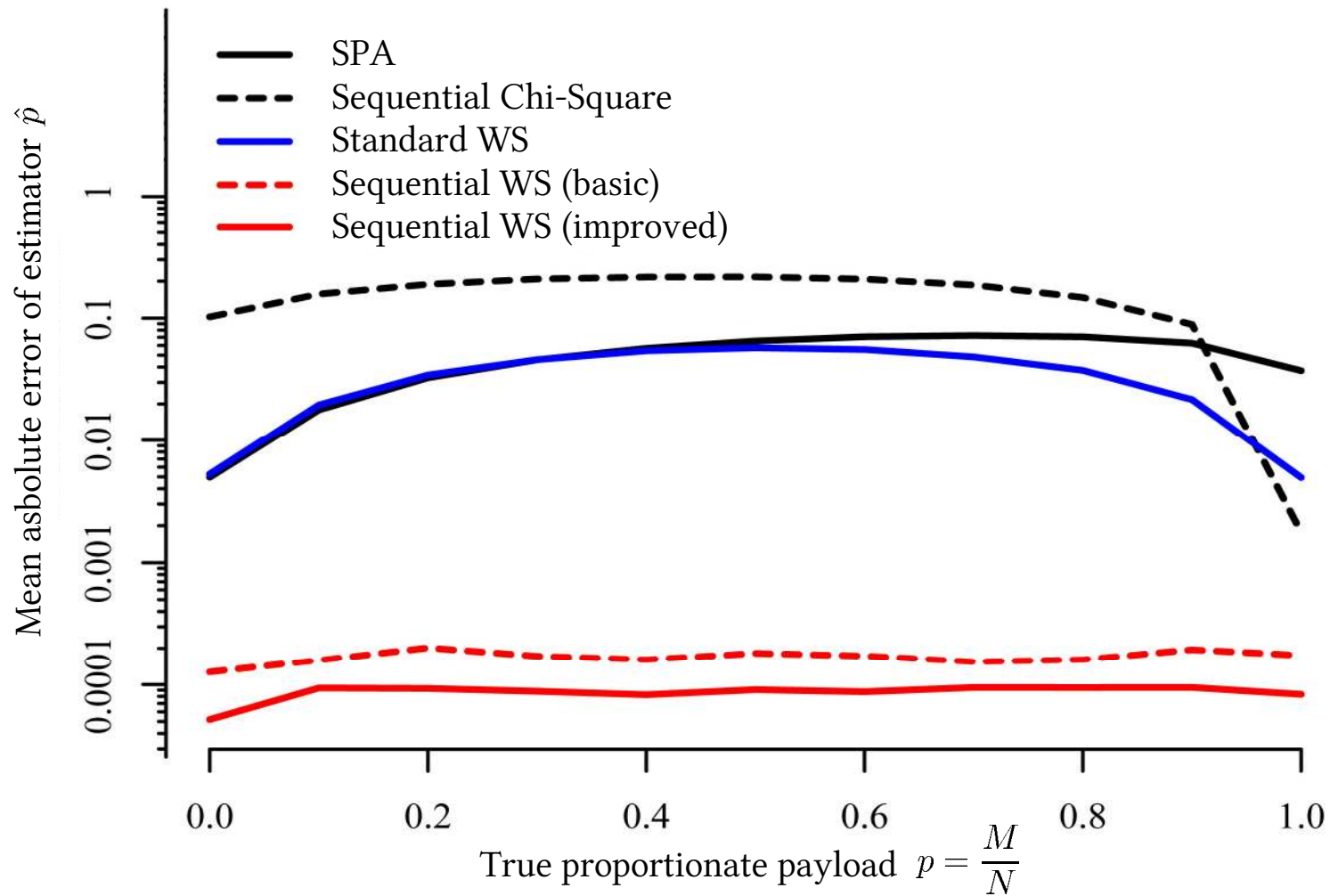
## Sequential WS Steganalysis

1. Estimate cover by filtering stego image:  $\hat{c} = s * (\text{filter})$
2. Estimate size of payload:

$$\hat{M} = \operatorname{argmin}_j \left( \sum_{i=1}^j \left( \left( \frac{1}{2}s_i + \frac{1}{2}\overline{s_i} \right) - \hat{c}_i \right)^2 + \sum_{i=j+1}^N (s_i - \hat{c}_i)^2 \right).$$

Weighting can also be used.

# Performance



Cover source:

1000 digital camera images, cropped to 0.3Mpixels

# Conclusions

- WS, a steganalysis method for LSB replacement, received little attention.

*Its performance was a little worse than “structural” detectors.*

- We upgraded its three components: cover prediction, weighting, and bias correction.

*For never-compressed covers, its performance is (almost always) superior to any other detector, and its computational complexity is low.*

- There are simple modifications for specialized detection of sequentially-located payload.

*The performance here is orders of magnitude better than its competitors.*

- WS has been unjustly neglected and, because of its modular design, there may be many other applications.

End

[adk@comlab.ox.ac.uk](mailto:adk@comlab.ox.ac.uk)