Batch steganography in the real world

Andrew Ker

adk@cs.ox.ac.uk

Department of Computer Science, Oxford University

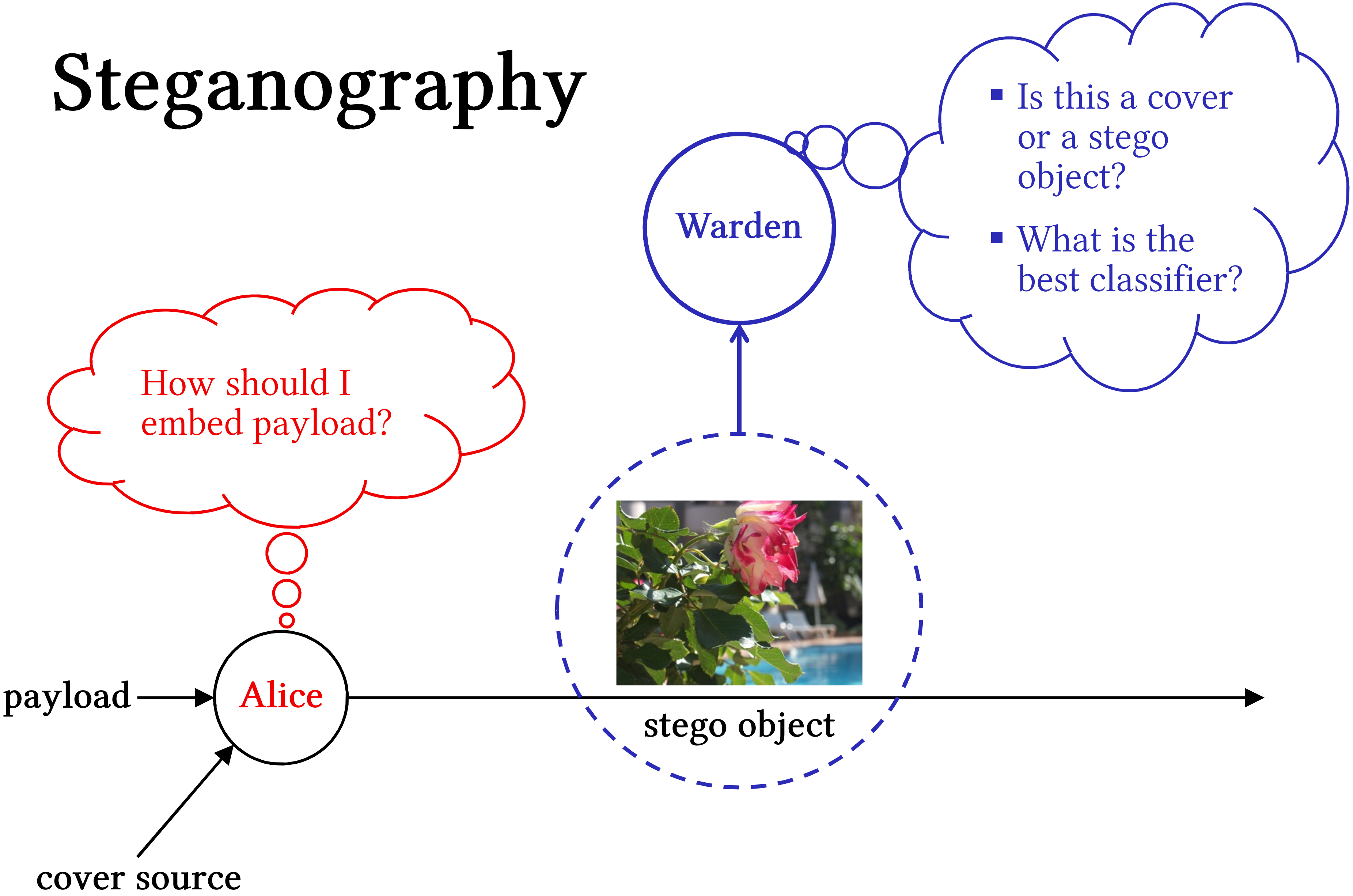
Tomáš Pevný

pevnak@gmail.com

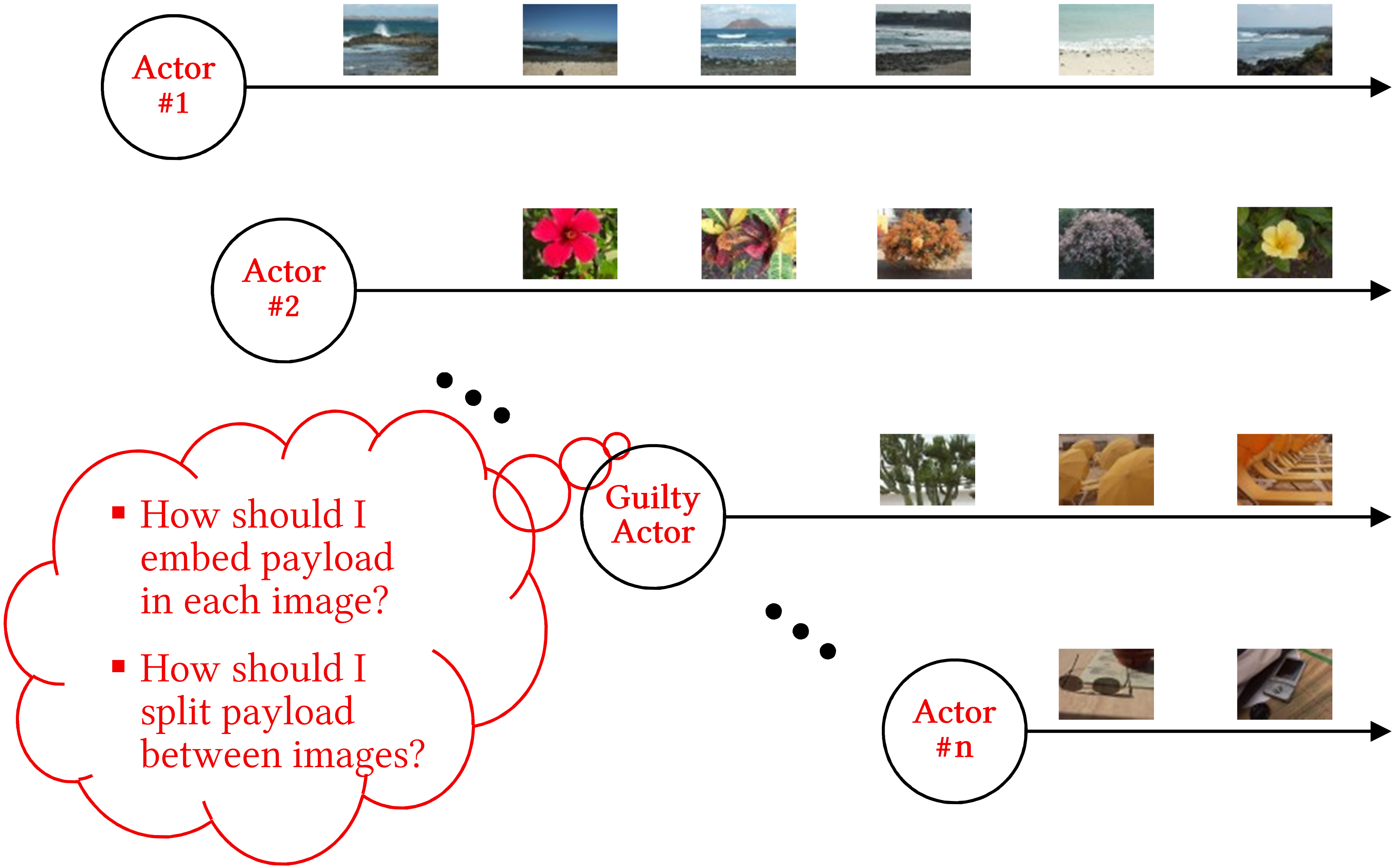
Agent Technology Center, Czech Technical University in Prague

14th ACM Multimedia & Security Workshop, Warwick University, 6 Sept 2012

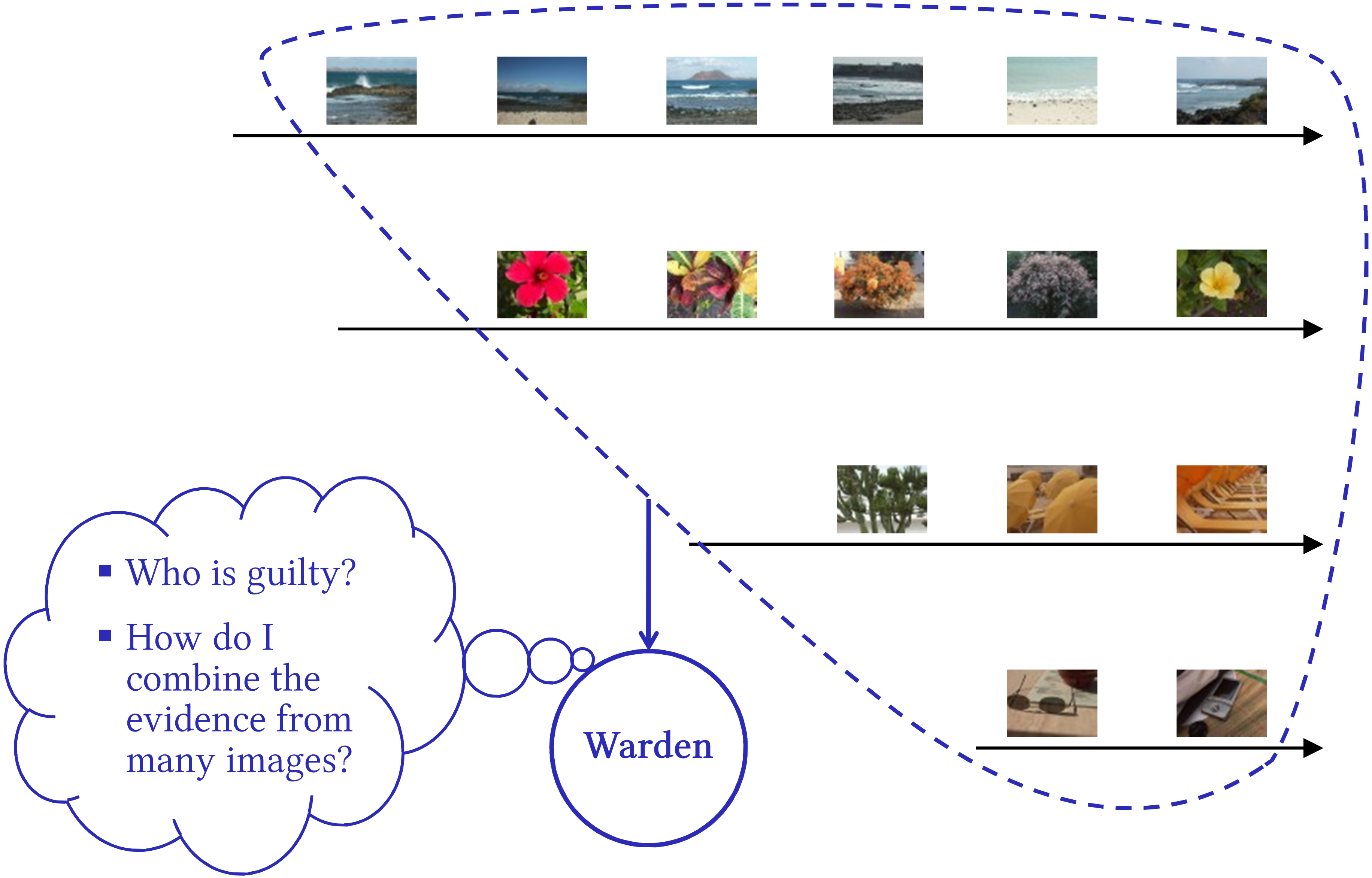
Steganography



Batch steganography



Pooled steganalysis



Setting

Little work published on these problems:

- Some game theoretic work on highly abstracted versions,
- No practical implementations.

[Ker & Pevný, 2011-12] finally proposes a method for pooled steganalysis.

Now we test batch steganography methods against it:

- different payload sizes,
- different hiding methods for individual images,
- different strategies for allocating payload.

'Batch steganography in the real world'

We limit ourselves to practically available methods and real-world JPEG images.

Hiding methods

Guilty Actor

- How should I embed payload in each image?

Freely-available steganography methods for JPEG images:

- ‘F5’ [Westfeld, 2001]
- ‘JP Hide&Seek’ [Upham, 2001?]
- ‘Steghide’ [Hetzl &c, 2005]
- ‘OutGuess’ [Provos, 2001]

A reference method from the literature, which is not freely available:

- ‘nsF5’ [Kodovský &c, 2007]

Embedding strategies

Guilty Actor

- How should I split payload between images?

A theoretical ‘optimum’ exists...

*use **Gibbs embedding** [Filler 2010] to minimize total distortion*

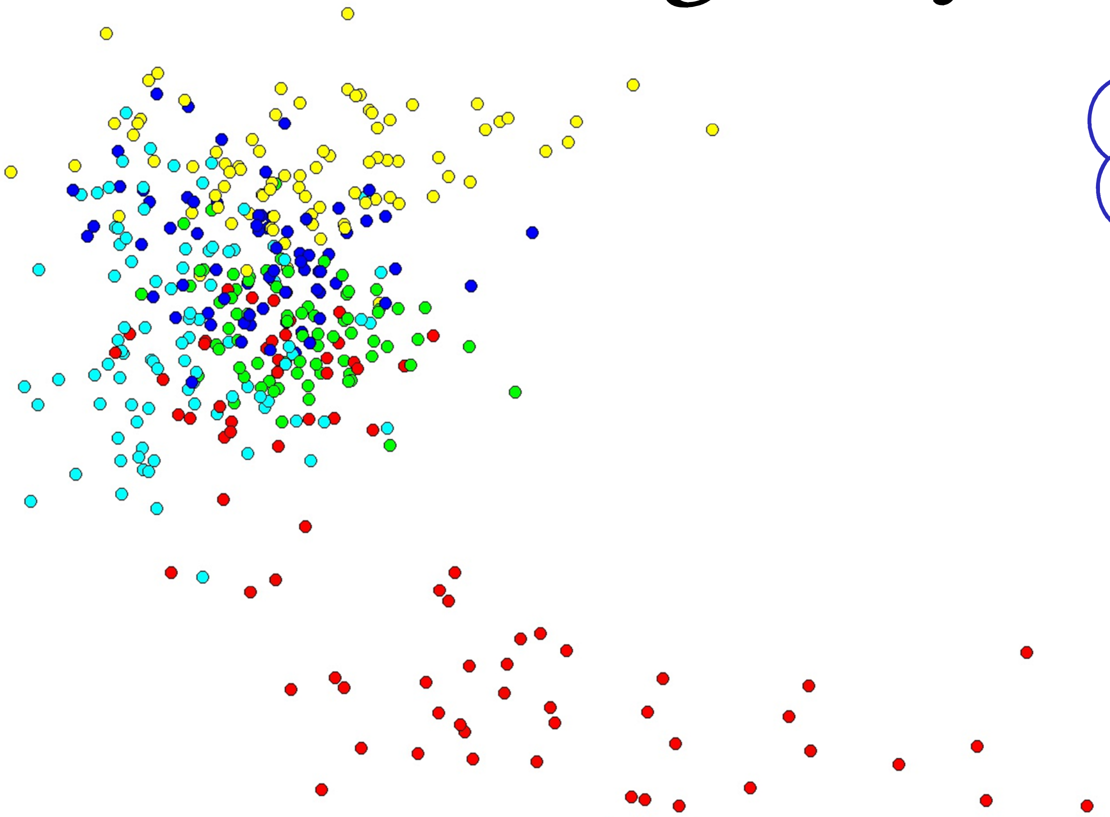
... but has caveats and is not freely implemented.

Naïve options

Let individual image **capacities** be $(c_1, \ldots, c_n)$; the total payload is M, and the amount embedded in each image is $(m_1, \ldots, m_n)$.

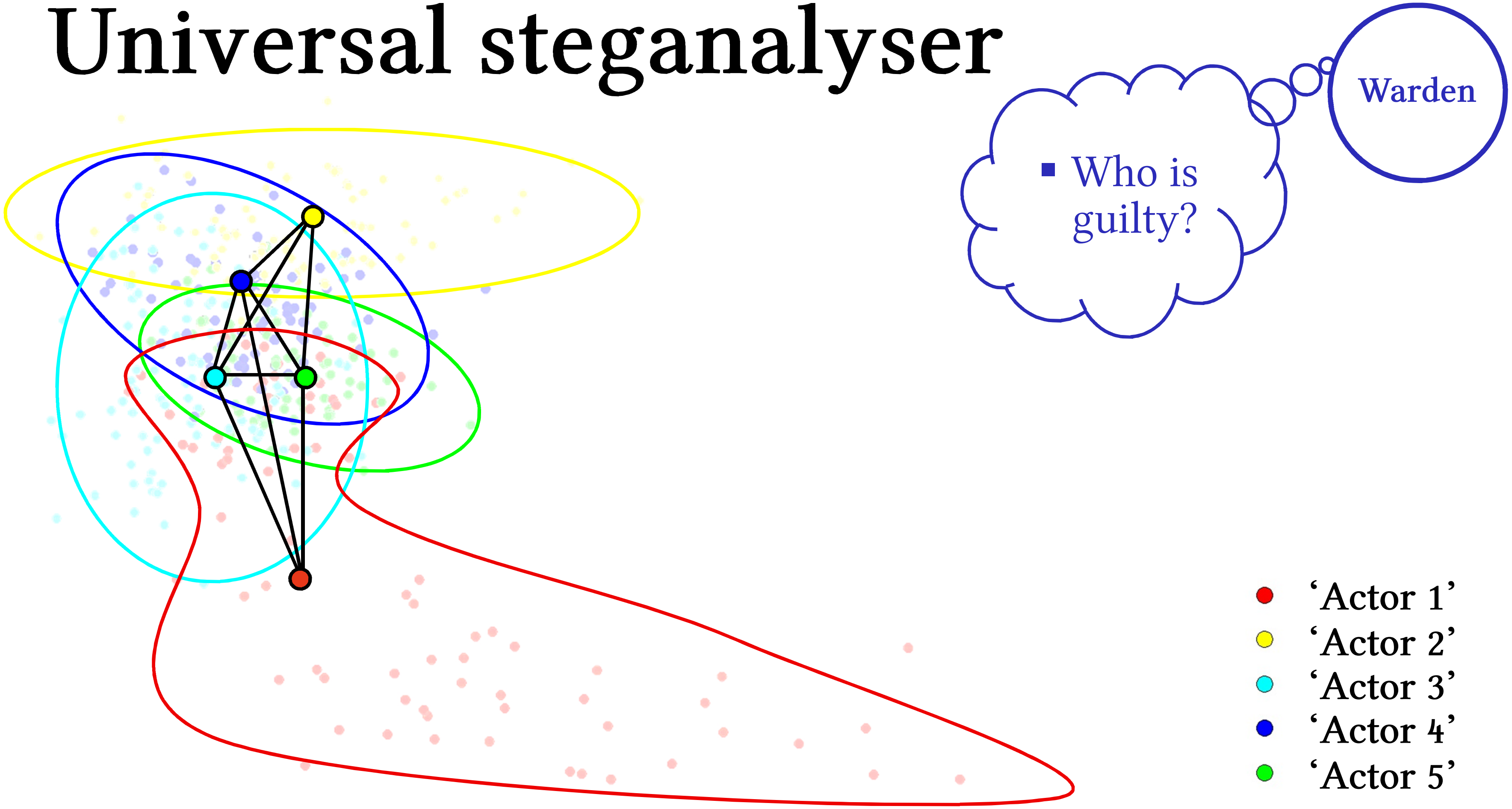
- ‘even’ — m_i constant
- ‘linear’ — $m_i \propto c_i$
- ‘max-random’ — $m_i = c_i$ for enough covers, selected randomly
- ‘max-greedy’ — $m_i = c_i$ for enough covers, with highest capacity

Universal steganalyser



- Many actors, transmitting many objects each.
- Different actors' sources have different characteristics: **model mismatch is guaranteed!**

Universal steganalyser



1. Extract features.
   Use each actor's output to estimate their overall distribution.
2. Compute a **distance** between each pair of actors.
3. Identify the steganographer(s).

Universal steganalyser

Features

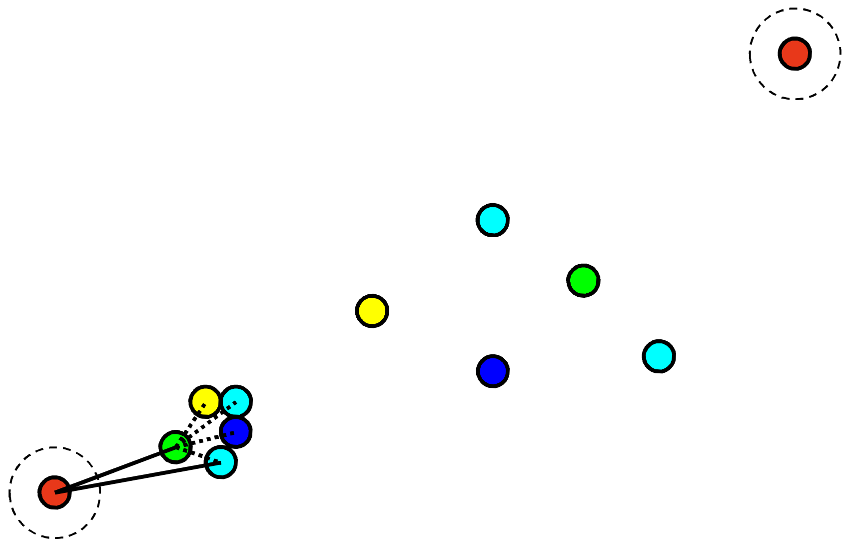
- 'PF274' features: 274-dimensional features for JPEGs.
- All features whitened (PCA) and rescaled ($\mu = 0$, $\sigma^2 = 1$).

Distance between actors

- Maximum Mean Discrepancy: $D(X, Y) = \sup_f E[f(X)] - E[f(Y)]$.
- Linear kernel: MMD = distance between actor's feature centroids.

Identification of steganographer(s)

- Local outlier factor.
  Compares local density with density around k-nearest neighbours.
- Ranks actors by level of suspicion.



Realistic, heterogeneous data set

On a leading social networking site…

- some users permit global access to images they appear in;
- we can click <u>next image</u> or <u>see more of user</u> (if user permits).

Automated process of following links, restricted to ‘Oxford University’ users, resulted in 4,051,928 images from 78,107 uploaders.

Ethics

- All data anonymized.
- Kept only images, grouped by ‘owner’, no personal information.
- All images globally visible at the time of download.

Realistic, heterogeneous data set

On a leading social networking site…

- some users permit global access to images they appear in;
- we can click next image or see more of user (if user permits).

Automated process of following links, restricted to 'Oxford University' users, resulted in 4,051,928 images from 78,107 uploaders.

Data set

- Selected 200 images from each of 4000 uploaders (actors).
- Filtered only for triviality and standard JPEG quality factor.
- Very challenging to work with.

Experiments

- Select {20, 50, 100, 200} random images from each of {100, 400, 1600} random actors.

- One is the **guilty** steganographer.

- Various total payloads,
  embeded using {nsF5, F5, JPH&S, Steghide, OutGuess},
  with strategy {even, linear, max-random, max-greedy}.

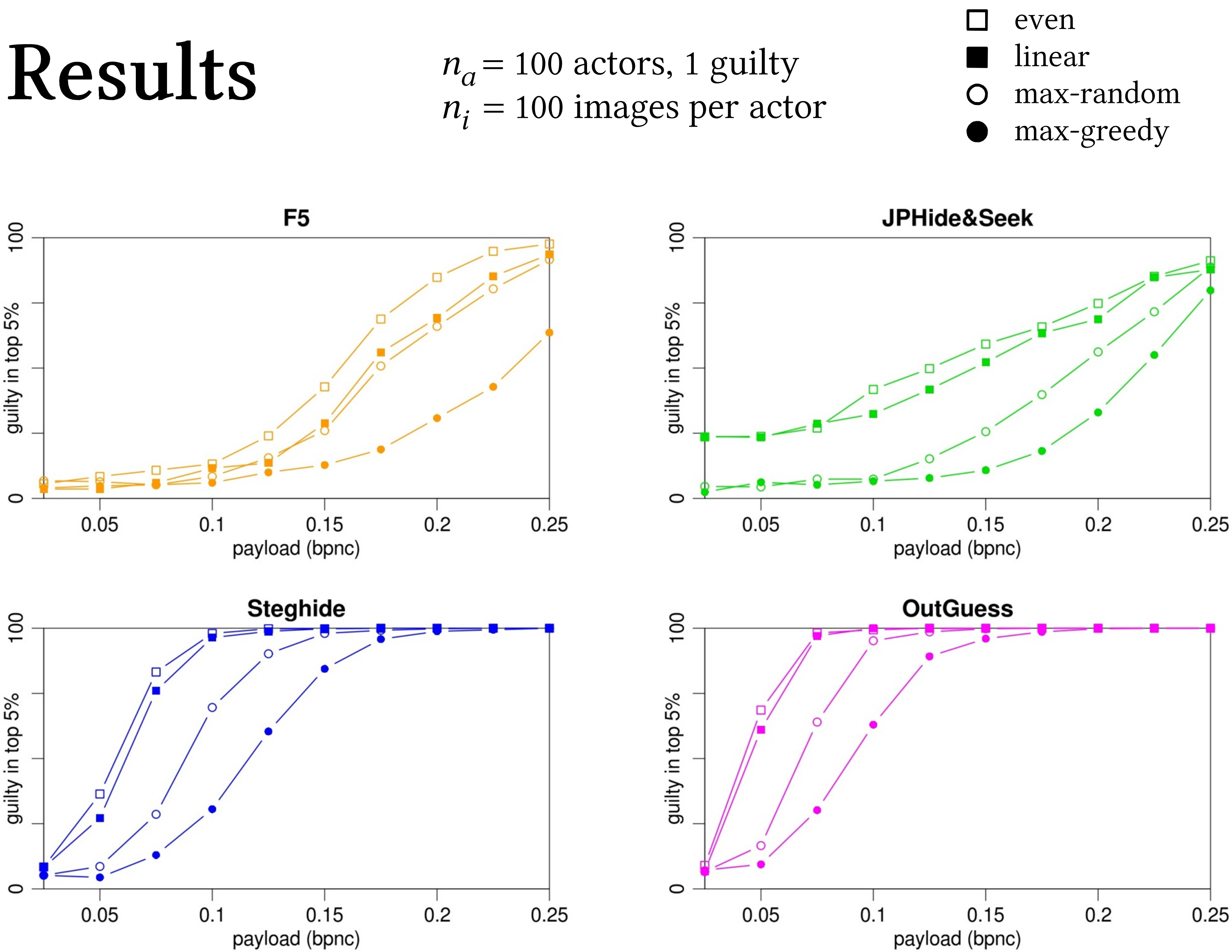
- Rank actors by suspiciousness according to our steganalyser.

Repeat $\times$ *500*

- How often does **guilty** actor appear in top 5% most suspicious?

Results

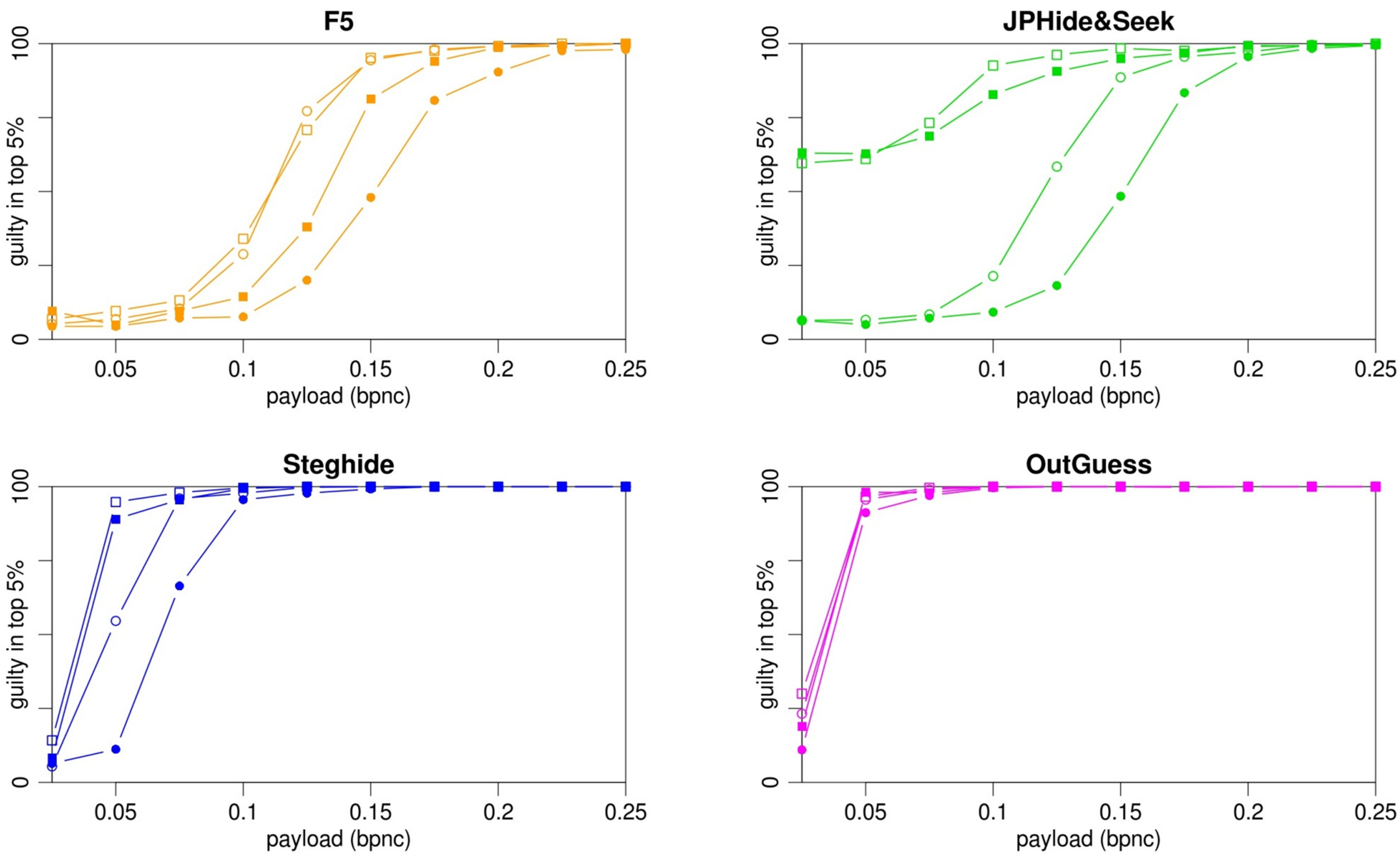
$n_a = 100$ actors, 1 guilty
$n_i = 100$ images per actor



Results

$n_a = 1600$ actors, 1 guilty
$n_i = 100$ images per actor

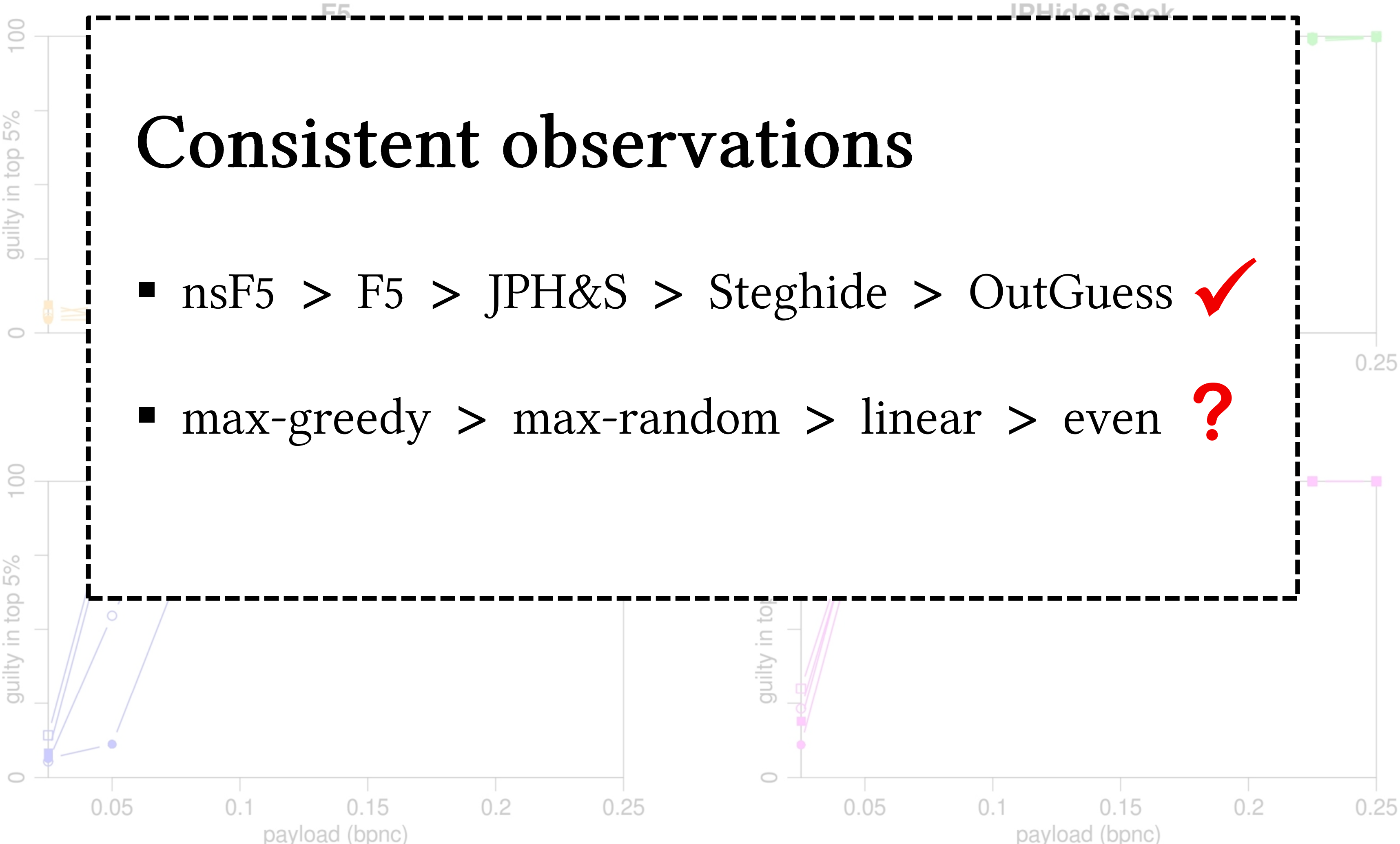
□ even
■ linear
○ max-random
● max-greedy



Results

n_a = 1600 actors, 1 guilty
n_i = 100 images per actor

- □ even
- ■ linear
- ○ max-random
- ● max-greedy



Consistent observations

- nsF5 > F5 > JPH&S > Steghide > OutGuess ✓
- max-greedy > max-random > linear > even ?

Linear distortion

$\boldsymbol{f}$ features of a cover image
$\boldsymbol{f}_p$ features of a stego image with payload length p

$$\boldsymbol{f}_p \approx \boldsymbol{f} + p\,\boldsymbol{\delta}.$$

Expected because

- embedding changes are roughly additive,
- [Pevný &c, 2012] successfully trained a linear payload estimator.

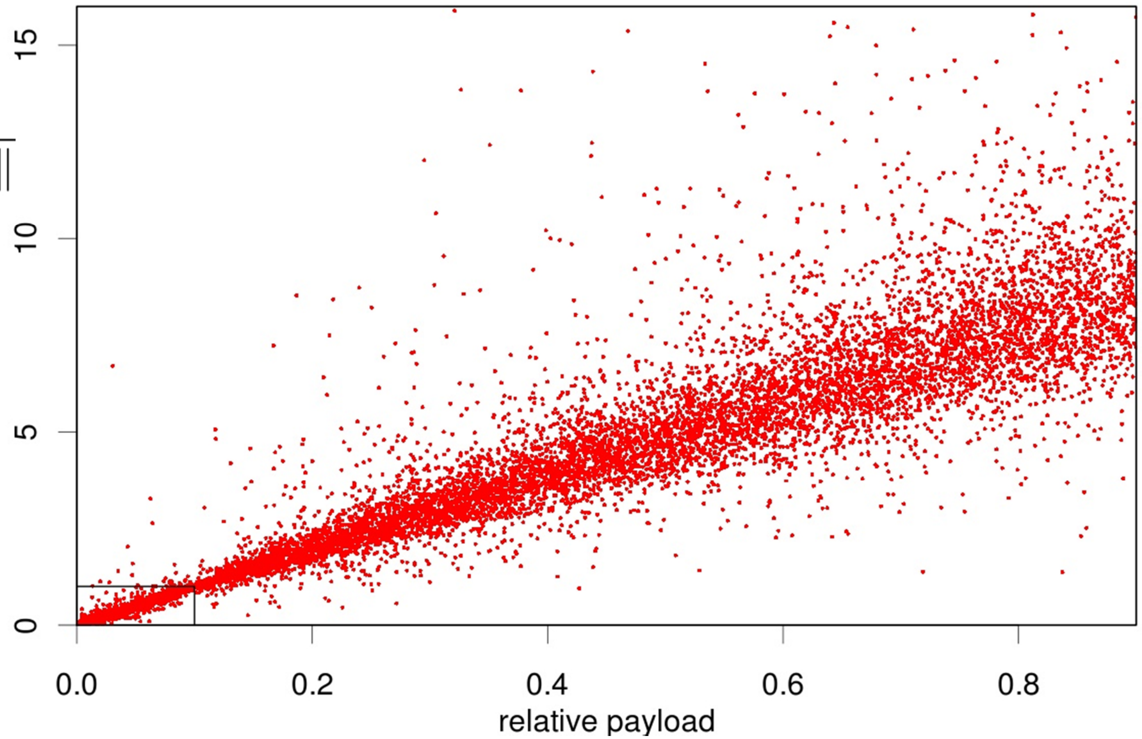
Linear distortion

$\boldsymbol{f}$ features of a cover image

$\boldsymbol{f}_p$ features of a stego image with payload length p

$$\boldsymbol{f}_p \approx \boldsymbol{f} + p\,\boldsymbol{\delta}.$$

$$\frac{\|\boldsymbol{f}_p - \boldsymbol{f}\|}{\|\boldsymbol{f}_{0.1} - \boldsymbol{f}\|}$$



10000 random images

Linear distortion

$\boldsymbol{f}$ features of a cover image
$\boldsymbol{f}_p$ features of a stego image with payload length p

$$\boldsymbol{f}_p \approx \boldsymbol{f} + p\,\boldsymbol{\delta}.$$

Expected because

- embedding changes are roughly additive,
- [Pevný &c, 2012] successfully trained a linear payload estimator.

Consequence: all strategies should be equally detectable.

(Detection depends on centroid of actors' feature clouds.)

Universal steganalyser

Features

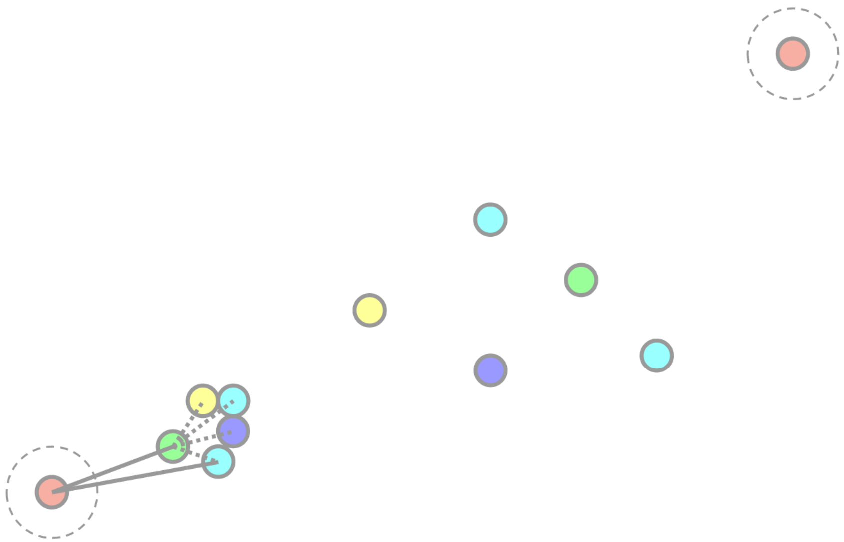
- 'PF274' features: 274-dimensional features for JPEGs.
- All features whitened (PCA) and rescaled ($\mu = 0$, $\sigma^2 = 1$).

Distance between actors

- Maximum Mean Discrepancy: $D(X,Y) = \sup_f E[f(X)] - E[f(Y)]$.
- Linear kernel: MMD = distance between actor's feature centroids.

Identification of steganographer(s)

- Local outlier factor.
  Compares local density with density around k-nearest neighbours.
- Ranks actors by level of suspicion.



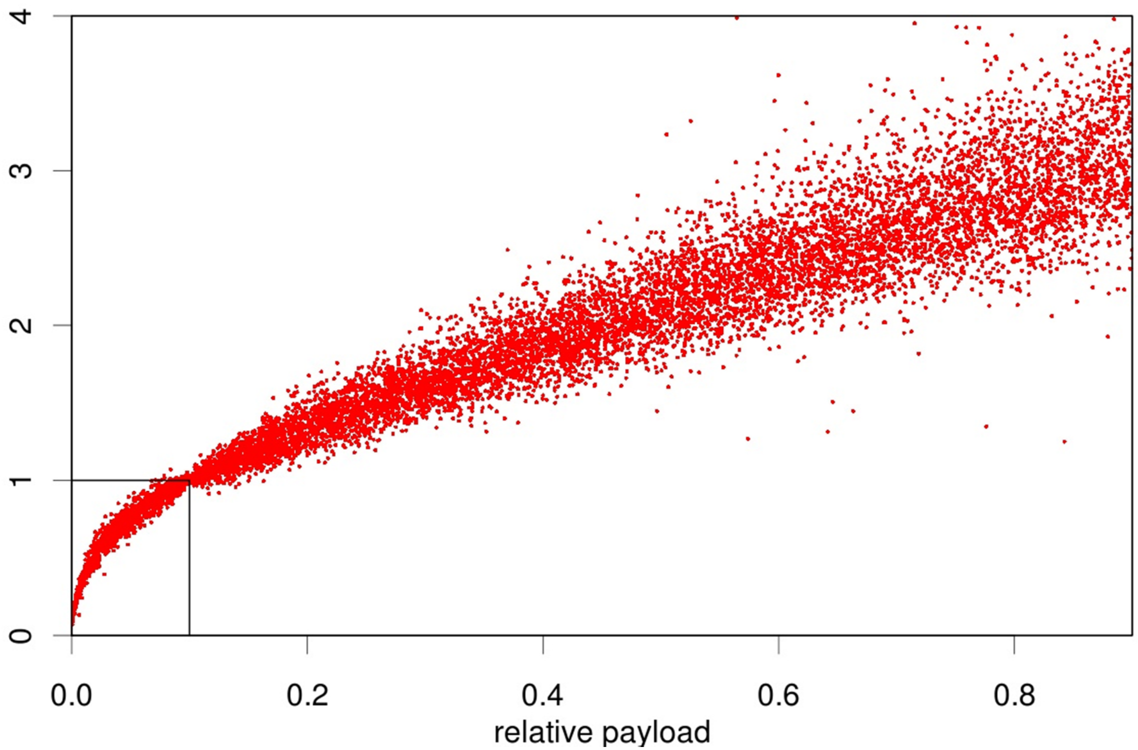
Nonlinear distortion

$\boldsymbol{f}$	features of a cover image
$\boldsymbol{f}_p$	features of a stego image with payload length p

$$\boldsymbol{f}_p \not\approx \boldsymbol{f} + p\,\boldsymbol{\delta}.$$

$$\frac{\|\boldsymbol{f}_p - \boldsymbol{f}\|}{\|\boldsymbol{f}_{0.1} - \boldsymbol{f}\|}$$

Whitened & normalized features



10000 random images

Nonlinear distortion

$\boldsymbol{f}$ features of a cover image

$\boldsymbol{f}_p$ features of a stego image with payload length p

$$\boldsymbol{f}_p \not\approx \boldsymbol{f} + p\,\boldsymbol{\delta}.$$

Whitened & normalized features

$$\|\boldsymbol{f}_p - \boldsymbol{f}\| = \sqrt{c_1(p)^2 + c_2(p)^2 + \cdots + c_n(p)^2}$$

some components are only noise

Conclusions

- The detector works in a wide range of situations.
  We confirm the relative security of hiding schemes,
  nsF5 > F5 > JPH&S > Steghide > OutGuess.

- We can learn about good batch steganography.
  Of the naïve embedding methods, greedy is best.

- The hider is exploiting a weakness in the detector…
  … (normalized) feature distortion is sublinear.

- This is a consequence of noisy (uninformative) feature components.
  Is it unavoidable in an unsupervised steganalyser?