AFRL-AFOSR-UK-TR-2014-0043

Universal Batch Steganalysis

Tomas Pevny

CZECH TECHNICAL UNIVERSITY IN PRAGUE
ZIKOVA 4
PRAHA 6, 16636
CZECH REPUBLIC

EOARD Grant 11-3035

Report Date: June 2014

Final Report from 01 July 2011 to 30 June 2014

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515, APO AE 09421-4515

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
30 June 2014	Final Report	1 July 2011 – 30 June 2014

4. TITLE AND SUBTITLE

Universal Batch Steganalysis

5a. CONTRACT NUMBER

FA8655-11-1-3035

5b. GRANT NUMBER

Grant 11-3035

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Tomas Pevny

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CZECH TECHNICAL UNIVERSITY IN PRAGUE
ZIKOVA 4
PRAHA 6, 16636
CZECH REPUBLIC

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515
APO AE 09421-4515

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOE (EOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2014-0043

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The overall theme of this project was to bring steganalysis into practical use, by proposing methods to identify a `guilty' user (of steganalysis) in large-scale datasets such as might be obtained by monitoring a corporate network or social network. Identifying guilty actors, rather than payload-carrying objects, is entirely novel in steganalysis, and we propose new methodologies to rank actors by their level of suspicion, without requiring any training data. We identified that modern so-called `rich' (large-dimensional) steganalysis features are not well-suited to unsupervised learning of this type, and developed novels ways to collapse large-dimensional data to reduce noise while retaining most of the evidence. These methods have been evaluated using large-scale experiments (in total over a million images tested in well over a billion combinations) on images crawled from genuine social networks. We also developed new implementations of existing steganalysis feature extractors, which were necessary for work on such a scale. We also examined a source of difficulty in all kinds of steganalysis: mismatch between actors caused by different cameras and post-processing. We proposed ways to mitigate such mismatch. Finally, we proposed a new method for attacking a single stego object by exhausting a secret key.

15. SUBJECT TERMS

EOARD, Nano particles, Photo-Acoustic Sensors

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James H Lawton, PhD
UNCLAS	UNCLAS	UNCLAS	SAR	18	19b. TELEPHONE NUMBER *(Include area code)* (703)696-5999

FA8655-11-1-3035
FA8655-13-1-3020

Universal Batch Steganalysis

Tomáš Pevný
Andrew D. Ker

1 July 2011-30 June 2014
1 Jan 2013-30 June 2014

Contents

1 Summary

We report on scientific progress made in the above-named grants, which formed a collaborative programme between Tomáš Pevný (CTU Prague) and Andrew Ker (Oxford). The project generated eight peer-reviewed scientific papers (seven at leading conferences, and one journal paper) and this scientific report does not duplicate their content. It functions as an extended abstract of those publications, pointing to their key contributions, along with excerpts from their experimental results.

The overall theme of the project has been to bring steganalysis into practical use, by proposing methods to identify a 'guilty' user (of steganalysis) in large-scale datasets such as might be obtained by monitoring a corporate network or social network. Identifying guilty actors, rather than payload-carrying objects, is entirely novel in steganalysis, and we propose new methodologies to rank actors by their level of suspicion, without requiring any training data. We identified that modern so-called 'rich' (large-dimensional) steganalysis features are not well-suited to unsupervised learning of this type, and developed novels ways to collapse large-dimensional data to reduce noise while retaining most of the evidence. These methods have been evaluated using large-scale experiments (in total over a million images tested in well over a billion combinations) on images crawled from genuine social networks. We also developed new implementations of existing steganalysis feature extractors, which were necessary for work on such a scale.

We also examined the a source of difficulty in all kinds of steganalysis: mismatch between actors caused by different cameras and post-processing. We proposed ways to mitigate such mismatch. Finally, we proposed a new method for attacking a single stego object by exhausting a secret key.

2 Introduction

Steganography is the hiding of information within an innocent 'cover': such a technology is becoming increasingly desirable for those who wish to evade network monitoring or to disguise their circle of correspondents. On the other hand, it presents a challenge for those with a legitimate need to monitor networks. The aim of steganalysis is to decide whether internet traffic contains hidden data, and the literature has developed rapidly in the last ten years. (For full literature surveys, see the papers [1–8].) However, it has developed in a particularly specialised direction, to binary classification (innocent cover vs. guilty stego) of single objects using machine learning methods, applied almost exclusively to grayscale images. The practical application of such a detector is uncertain, as it requires potentially unrealistic knowledge by the detector: the exact embedding method used, (usually) the size of the embedded payload, and training data from the same source as that being examined. Furthermore, it does not generalise well to detection of multiple objects.

This research project lies within the general theme of making steganalysis more practically usable. In a survey paper with other authors, we identified a number of open research problems aiming towards this goal [2]. This research project addresses some of those open problems. In the following we summarise the problems addressed, grouping our progress and publications by theme.

The central part of this project has been the development of an entirely new kind of steganalysis, *universal batch steganalysis*. It proposes a new kind of hidden data detection where a steganalyst is monitoring a large network, with multiple users and many potentially suspect communications. We identified four requirements:

Universality. The steganalyst need not know what steganography algorithm is being used. Most existing steganalysis methods do not have this property.
Robustness. There should be no requirement for reliable training data from sources identical to those being monitored. Again, existing steganalysis methods generally fail this condition.
Multiple actor. The network has multiple users, some (most) of them innocent of steganographic embedding, but each with a slightly different cover source. The detector needs to determine *who* is guilty, not necessarily which of their objects specifically contain payload. In this research, we aim to rank the actors in order of likeliness of guilt.
Multiple object. Each actor emits many objects. For innocent actors, all of their objects are plain covers. For guilty actors, some (not necessarily all) of them contain payload.

Our publication [4] proposed the method, with subsequent testing on large-scale social network images and analysis of an interesting phenomenon of nonlinear distortion of features in [3]. Further experiments, and exploration of detection parameters and different detection kernels, were added to make the journal paper [6]. This work addressed the following open problems from [2]:

> '**Open Problem 12**: Theoretically well-founded, and practically applicable, detection of payload of unknown length.'
>
> '**Open Problem 19**: Any detector for multiple objects, or based on sequential hypothesis tests.'
>
> '**Open Problem 17**: Unsupervised universal steganalysis.'

The first part of steganalysis is to reduce each object examined to a set of 'features'. In the above research we used well-established features known as PF-274 (rather than repeat citations here, we refer the reader to the bibliographies of our published papers) which are 274-dimensional. However, recent developments in steganalysis have produced 'rich' features that are more powerful (can detect small payloads more accurately) but have many thousands of dimensions. We investigated one such feature set (the 7850-dimensional features known as $\mathcal{CF}^*$), in conjunction with the universal batch steganalyzer, in [7], where it exhibited *worse* performance because noise accumulates across the many dimensions and it cannot be removed by unsupervised methods. We examined methods for reducing dimensionality, and proposed a new procedure known as Calibrated Least Squares (CLS) which finds projections that reduce noise and enhance steganographic signal. Further large-scale tests demonstrated that that $\mathcal{CF}^*$ features, reduced in dimension using CLS (even when that reduction is done using an incorrect embedding algorithm) can determine a guilty actor almost uniquely. This work addressed from [2]:

> '**Open Problem 18**: Design of features suitable for universal steganalysis.'

The computational demands of the large-scale experiments required new implementations of existing steganalysis feature extractors. A particularly challenging feature set, so-called Projected Spatial Rich Model (PSRM), was implemented on a Graphic Processing Unit (GPU) and this research is published in [1].

We then turned to the failure cases, attempting to quantify the effect of natural differences between difference actors' image sources (due to camera, storage format, postprocessing, and other unknown factors). This work was carried out in the context of supervised binary classification. Using a different data set, we correlated detection inaccuracy with differences in features and found a number of significant factors, some

of which can be corrected for at the detector. We also proposed and tested 'abstaining classifiers' which report an 'unknown' classification on testing data not near to any training data. Novel benchmarks were also proposed. This work is reported in [5] and addresses the problem from [2]:

'**Open Problem 15**: Attenuate the problems of cover source mismatch.'

Finally, we did a small piece of research which addresses the other end of a steganalysis forensics problem: once we have focused on a single image that we are sure contains data, how do we determine the embedding key? By combining statistical evidence with a form of brute-force search, we proposed a novel Bayesian inference method reported in [8] (winner of the conference best paper award). This addresses in part the open problem from [2]:

'**Open Problem 23**: Is there a statistical approach to key brute-forcing, for adaptive steganography?'

3 Methods, Assumptions, Procedures

Grouping our research progress by theme, we summarise the methods we have proposed. For a proper context, and full technical details, we refer to the publications from which these descriptions are excerpted.

3.1 Methods

Unsupervised batch steganalysis

Our proposed detector identifies actors that significantly deviate from the majority. We assume that the detector who knows which user (actor) emitted which object (image) on the network. The detector works in three steps: extracting steganalytic features from all objects; calculating distances between each pair of actors from the feature points they have emitted; identifying actors deviating from the majority.

The steganalyst's first design decision is to select a feature set. A detector should work with any steganalytic features sensitive to embedding changes and relatively insensitive to image content. In our experiments of [3, 4, 6], which use JPEG images, we used so-called PF-274 features, for reasons explained in [6]. An important step is to pre-process the features to make their scales comparable. In [6] we determined that a global whitening works best, so that features are uncorrelated and they have unit variance in each direction.

We proposed to measure distance between actors using an empirical Maximum Mean Discrepancy (MMD), which is a measure of similarity between probability distributions. It can be estimated robustly, even in high dimensions, from relatively little data. Assuming n samples $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ and $\{y_i \in \mathbb{R}^d\}_{i=1}^n$, pre-processed feature vectors from actors X and Y, a sample estimate of the MMD distance has the following simple form

$$MMD(X,Y) = \frac{1}{\binom{n}{2}} \sum_{1 \le i < j \le n} \kappa(x_i, x_j) - \kappa(x_j, y_i) - \kappa(x_i, y_j) + \kappa(y_i, y_j)$$

where $\kappa(x, y) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is a kernel function. We investigated linear, polynomial, and Gaussian kernels in [6].

For detecting deviating actors, we chose the local outlier factor (LOF) method because it detects outliers in uneven probability distributions, and the provided anomaly score is interpretable. Given a set P of points (actors) with a metric $d : P \times P \to [0, \infty)$ and an integer parameter $1 < k < |P|$, the LOF is calculated as follows. The **reachability distance of point p from q**, $r_k(p, q)$, is the greater of $d(p, q)$ and $d(q, q')$, where q' is q's k-nearest neighbour. Fix a point p, and write P_k for the k-nearest neighbourhood of p in P. The **local reachability density of p** is defined as an inverse of the average reachability distance of point p from all points $q \in P_k$,

$$lrd_k(p) = \left(\frac{1}{k} \sum_{q \in P_k} r_k(p, q) \right)^{-1},$$

and the **local outlier factor (LOF) of** p is

$$lof_k(p) = \frac{1}{k} \sum_{q \in P_k} \frac{lrd_k(q)}{lrd_k(p)}.$$

Thus $lof_k(p)$ captures the degree to which p is further from its k-nearest neighbours than they are from theirs. Defining it as a relative number means that it does not depend on absolute values of distances $d(p, q)$ and it allows us to rank actors from most anomalous (most likely to be guilty) to least. We investigated the effect of the parameter k in [6].

Dimensionality reduction

Our aim here was to project large dimensional features to fewer dimensions, while retaining as much evidence of steganalysis as possible. We suppose the existence of training data $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ extracted from n images are arranged in a matrix $\mathbf{X} \in \mathbb{R}^{n,d}$. It is assumed that the mean of each column has been set to zero. The notation $\mathbf{X}^c$ / $\mathbf{X}^s$ denotes features extracted exclusively from cover / stego images respectively. The corresponding steganographic payload rates form a column vector $\mathbf{Y}^s \in \mathbb{R}^{n,1}$.

Some existing methods of dimensionality reduction are as follows.

Principal Component Transform (PCT) can be expression as an iterative algorithm, where in k^{th} iteration one seeks a projection vector $w_k \in \mathbb{R}^d$ best explaining the data (maximising the variance) and being orthogonal to all previous projections $\{w_i\}_{i=1}^{k-1}$. This can be formulated as

$$w_k = \arg\max_{\|w\|=1} w^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} w$$

subject to

$$w^{\mathrm{T}} w_i = 0, \ \forall i \in \{1, \ldots, k-1\}.$$

The Maximum CoVariance (MCV) method finds a direction maximising the *covariance* between the projected data $\mathbf{X}^s w$ and the dependent variable $\mathbf{Y}^s$. Again, the vector w_k found in k^{th} iteration should be orthogonal to previous projections $\{w_i\}_{i=1}^{k-1}$. It finds

$$w_k = \arg\max_{\|w\|=1} \mathbf{Y}^{s\mathrm{T}} \mathbf{X}^s w$$

subject to the same condition.

Ordinary Least Square regression (OLS) finds a single direction w minimising the total square error between the projected data $\mathbf{X}^s w$ and the dependent variable $\mathbf{Y}^s$. To avoid numerical error, a small diagonal matrix is added to prevent non-singularity and increase the stability of the solution, and the optimisation problem can be formulated in a similar way to the preceding, as

$$w = \arg\max_{w \in \mathbb{R}^d} 2\mathbf{Y}^{s\mathrm{T}} \mathbf{X}^s w - w^{\mathrm{T}} \mathbf{X}^{s\mathrm{T}} \mathbf{X}^s w - \lambda \|w\|^2.$$

The parameter λ acts as a regularisation constraining the complexity of the solutions and we fixed $\lambda = 10^{-7}$. It is indeed possible to turn OLS into an iterative algorithm, adding the same orthogonality constraint as in PCT and MCV, but doing so does not give us new directions.

In [7] we proposed a novel method for dimensionality reduction called Calibrated Least Squares (CLS) Regression. The projections are *sensitive* to embedding changes yet *insensitive* to the image content. This means that the covariance between the projection of stego features and their embedding change rate should be high, while the variance of projection of cover features should be low. In the k^{th} iteration, the algorithm solves following problem

$$w_k = \arg\max_{w \in \mathbb{R}^d} 2\mathbf{Y}^{s\mathrm{T}} \mathbf{X}^s w - w^{\mathrm{T}} \mathbf{X}^{c\mathrm{T}} \mathbf{X}^c w - \lambda \|w\|^2,$$

subject to

$$w^{\mathrm{T}} w_i = 0, \ \forall i \in \{1, \ldots, k-1\}.$$

This is similar to the problem being solved in OLS, but it reflects more closely our goal.

Feature implementation

The Projected Spatial Rich Model (PSRM) features are the most recent development in steganalysis of still images. They can be applied to both spatial-domain (uncompressed image) and transform-domain (JPEG) steganography. However, they require tremendous amounts of computation, over 10^{12} floating point operations (1 TFLOPs) for a 1 megapixel image. We designed a new implementation using Compute Unified Device Architecture (CUDA) on NVIDIA graphics cards. The key to good performance is to combine computations so that registers are not exhausted, because memory access has high latency on such devices, and to write code that can be optimized for register use. The feature definition was slightly adjusted to enable certain optimization, and we demonstrated that this does not impact their detection performance. Our implementation reduced computation time from approximately 25 minutes per megapixel to 2.6 seconds. This work was fully reported in [1].

We also created a custom implementation of the popular JPEG Rich Model (JRM) feature set, which does not require a GPU, that we can use in large scale experiments. It reduces feature computation time from approximately 15 seconds per megapixel to 0.5 seconds.

Once the code is cleaned up, these implementations will be made available to other researchers.

Mismatch mitigation

In this part of the project, we explore potential causes of *cover source mismatch*, which occurs when a supervised steganalysis algorithm is trained on images from one source but tested on another. This requires new benchmarks, and in [5] we proposed a number of novel metrics. Given error rates $P_{\mathrm{E}}^1, \ldots, P_{\mathrm{E}}^k$ from k different classifiers with different combinations of training and testing data, we use the *mean error rate* $\mu_1 = \frac{1}{k}\sum_i P_{\mathrm{E}}^i$, the *root mean square error rate* $\mu_2 = \sqrt{\frac{1}{k}\sum_i \left(P_{\mathrm{E}}^i\right)^2}$, and the *maximum error rate* $\mu_\infty = \max_i P_{\mathrm{E}}^i$. They are justified theoretically in [5].

We propose methods to measure certain types of discrepancy caused by different image sources: difference in location of cover features, difference in direction of stego features from covers, and difference in rates of movement with respect to payload. In our data the first two are significant, the last one is not. We propose methods for mitigating such discrepancies: (i) centering actors' images if a small amount of matched training data is available; (ii) using an ensemble of classifiers, trained on different sources, with equal weights; (iii) using an ensemble where the classifiers are weighted according to their correspondence with the object being studied, where correspondence is estimated by re-embedding a small additional payload.

In the same paper we also proposed a novel *abstaining* classifier, which is not subject to false certainty. We propose to combine two one-class detectors, one trained on covers and the other on stego images. If the cover detector returns *positive* and the stego detector returns *negative* we have a negative detection (false negative probability denoted P_{FN}). If the cover detector returns *positive* and stego detector returns *negative* we have a negative detection (false positive detections is still denoted P_{FP}). If both detectors return *negative* then we have a 'don't know' arising from a lack of data: the classified object is in a novel region for which there was not sufficient training data (probability denoted P_{DN}). If both detectors return *positive* it is also a 'don't know', in this case arising from contradictory evidence: the classified object would probably have been near the decision boundary of a 2-class classifier. (We denote the probability P_{DP}.)

Again, we propose novel metrics for measuring the accuracy of abstaining classifiers. In [5] we justify theoretically the measure

$$d_p = \frac{\left(1 - P_{\mathrm{FP}} - P_{\mathrm{FN}}\right)\sqrt{1 - P_{\mathrm{DP}} - P_{\mathrm{DN}}}}{\sqrt{P_{\mathrm{FP}}(1 - P_{\mathrm{FP}})} + \sqrt{P_{\mathrm{FN}}(1 - P_{\mathrm{FN}})}}$$

where the error rates P_{FP}, etc., are pooled over all matched or mismatched training experiments. Another metric is

$$d_\infty = \max d_i$$

where the d_i is calculated as d_p separately for each experiment.

Key exhaustion

In this work [8] we consider the extraction process, decoding a steganographic payload from a password p: an embedding key is derived using a key derivation function: $k = \mathrm{KDF}(p)$; k locates and decodes a small block of metadata, the *decoding parameters*; the decoding parameters and the embedding key are used to extract the payload. The **exhaustion attack** (on p) simply tries all possible passwords in turn until a recognisable plaintext is recovered. We examine ways to blend statistical steganalysis with the exhaustion attack, for cases where the plaintext is not recognisable. Typically we can make use of the fact that not all decoding parameters are possible, and further that some are more likely than others.

We suppose that a number of different stego objects have been intercepted, that were embedded using the same password p. In the **intersection attack** the attacker maintains a list of possible passwords p. For each image received, they remove all keys which produce impossible decoding parameters. In [8] we explain why impossible parameters are common, and countermeasures that implementations should use.

We also propose a more sophisticated attack known as **Bayesian key inference**. If $p(k)$ represents the prior probability that the key is k, and $x(k)$ the payload length decoded by key k, then the posterior $p(k|y)$ after one observation of a stego image with estimated payload y is given by

$$p(k|y) = \frac{\mathrm{P}\big(y|x(k)\big)p(k)}{\sum_{k'} \mathrm{P}\big(y|x(k')\big)p(k')} \propto \mathrm{P}\big(y|x(k)\big)p(k),$$

where $\mathrm{P}(y|x)$ denotes an estimate of the probability that true payload length x produces a quantitative steganalytic estimate y. Simplifying, the unscaled posterior can therefore be written

$$\log p(k|y_1, \ldots, y_n) \propto \log p(k) + \sum_{i=1}^{n} \log \mathrm{P}\big(y_i|x(k)\big).$$

Bayesian inference is still possible if the key k does not imply a specific payload length, for example when it determines decoding parameters with respect to a syndrome code. Instead, each k implies *bounds* on the payload size. Assuming a uniformly random number of changes x between these two limits, we can still compute

$$P(y|k) = \sum_{x} P(y|x)P(x|k)$$

and apply inference as before. In [8] we showed that the true key should dominate the posterior and can therefore be identified with a known level of certainty.

3.2 Assumptions

Our large-scale experiments have been carried out using images from real social networking sites (see the next subsection for details). We have used them as cover images and simulated 'guilty' actors and images using existing steganography tools, measuring our ability to detect the guilty actor or image accurately. Throughout all the experiments we have made the assumption about the ground truth, that none of the download images contains steganographic payload. We argue that this is likely because steganography tools are not easy to use, particularly not with social networking sites that re-compress uploaded images. If the assumption is incorrect, our results will *understate* the accuracy of our detection methods, because we would attribute a false positive to an actor who is in fact a true positive.

In most of the anomaly detection work [3, 4, 6, 7] we simulated situations where exactly one actor (out of 100, 200, or 400) was guilty, and measured our ability to rank the guilty actor amongst the most suspicious. This makes the assumption that only one actor is guilty, and we briefly consider the case of more guilty actors in [6]. It was not within the scope of this project to determine whether *any* actors are guilty.

3.3 Procedures

We briefly describe the procedures we used to obtain the images for our experiments, and the structure of the most important experiments.

For the large-scale experiments of [3, 4, 6, 7] we obtained images from a leading social network site, which is popular for sharing pictures. We wrote scripts to crawl pages of *public* images from users who identified themselves as members of Oxford University, obtaining more than 4 million images from more than 70 000 users. The *actors* in our experiments are the uploaders, which mimic well the behaviour of real-world actors: sometimes a single actor uses two or three cameras. Personally identifiable information was removed, and the files anonymized. In these papers we used a randomly selected subset of 4000 actors and 200 images for each actor, for a total of 800 000 images.

Apart from a fairly uniform size and completely uniform quality factor, the images in the database are very diverse, as they come from different sources (cameras, flatbed scanners), have varied content, and most likely underwent different image processing from acquisition to download. Some of them are not natural images at all, but synthetic images or mosaics. We did not remove such "impurities": they are there in practice and these images are a good prototype for what might be expected when monitoring a real network.

The images used in [5] came from a different source, a popular photo-only sharing site. We crawled images that were explicitly available under a creative commons license (again anonymizing completely) and for these experiments used 9000 photos from each of 9 uploaders. A subset of these images is used in [8].

For experiments to test the unsupervised batch stganalyzer, we randomly selected n actors out of the 4000 in our data set, and 100 images from each actor. Exactly one guilty actor is simulated, by using the chosen embedding algorithm to insert of payload of a certain number of bits per nonzero DCT coefficient, spread between the 100 images using different strategies. We then calculate features from each of the $100n$ images, MMD between each pair of actors, and LOF scores for each actor. For each combination of parameters, the experiment is repeated 500 times with a different selection of actors and guilty actor. We benchmark the detector by the average rank of the guilty actor across these 500 repetitions.

In experiments on cover source mismatch, we trained one Fisher Linear Discriminant (FLD) detector for each of the nine actor's images (using 6000 of their images). For each actor we tested (the remaining 3000 images) using an ensemble from the other eight detectors. The votes of the ensemble were aggregated in various ways. This was repeated 100 times.

4 Results and Discussion

Our experimental results are published in full in [1,3–8]. Here, we highlight some of the key results obtained during this project, grouping them by theme as before. For a fuller explanation of the results and their significance, see [1,3–8].

Unsupervised batch steganalysis

Figure 1 shows the average rank of a guilty actor as assessed by the proposed universal detector with respect to the payload emitted by the guilty actor. The evaluation is performed on images from the social network database for (a) the three different numbers of actors and (b) four strategies of spreading the message among images available to the guilty actor: the greedy / maximum strategy puts the message into as few images as possible with / without the knowledge of their capacity; the linear / even strategy divides the message evenly in images with / without the respect to their capacity. The embedding algorithm for the results displayed here was F5 (four others are used in [3, 6]). The steganalyzer used PF-274 features, pre-processed by principal component transformation removing all components with smaller eigenvalue than 0.01. The MMD kernel used to calculate distance between actors was linear (more on this below).

The figure shows that for payloads higher than 0.15bpp the universal steganalyzer identifies guilty actors correctly. It also shows that strategies that use information about image capacity (linear, greedy) are more secure. We might also conclude that the greedy strategy is more secure than the linear one. We investigated

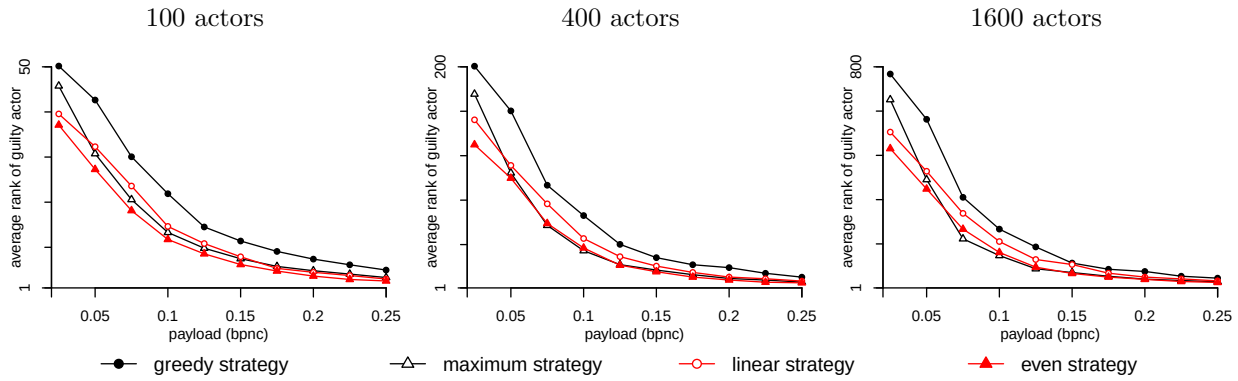

Figure 1: (Taken from [6].) Performance of unsupervised batch steganalysis: from left to right, different numbers of actors n (of which one is guilty); 100 images per actor in each case; lines in each chart denote different strategies for the guilty actor to allocate payload amongst their 100 images; each x-axis represents total payload (bpnc) and the y-axis represents the average rank of the truly guilty actor (lower is better; 1.0 represents perfect detection and $\frac{n+1}{2}$ random guessing). The detector parameters are: linear kernel, whitened features, LOF parameter $k = 10$.

the cause of this in [3], and visualized in Figure 2 the relationship between feature distortion and relative payload. The relative *distortion quotient* of features, from cover to stego images, was measured as

$$\frac{\|x_p - x_0\|}{\|x_{0.1} - x_0\|},$$

where x_0 are the cover features, $x_{0.1}$ features from the image with a fixed payload of 10% of its capacity, and x_p features from the image with a random-length payload, proportion p of its capacity. Figure 2 nicely demonstrates how PCT improves the sensitivity of features to the embedding operation, as the distortion is more correlated to the payload in the right image than in the left one.

We observe, though, that distortion in whitened features is non-linear with respect to the relative payload. In [3] we explained that this is due to noise in the whitened and normalized features and in [6] we demonstrated that such preprocessing is essential to the detector's performance. It explains why, with respect to the linear kernel, the greedy strategy is the hardest to detect: the total distortion accumulated from all guilty actor's images is smaller if the message is hidden in small number of images.

However, this observation only holds when MMD is calculated using a linear kernel. Other kernels are better able to detect large deviations, and in [6] we uncovered an interplay between the strategy steganographer uses to to distribute the message in images and the MMD kernel used by the detector. Figure 3 shows average rank of guilty actors for strategies interpolating between the greedy strategy (left) towards linear strategy (right) for steganalyzers using linear (centroid), polynomial, cubic, and Gaussian kernel. We can observe that against a greedy strategy the detector is better to use the quadratic kernel, but against a linear strategy it is better to use the Gaussian kernel. Game theory can be used to find the optimal strategies for the steganographer and steganalyst in this game, which we have demonstrated in [6].

Dimensionality reduction

Table 1 compares the proposed Calibrated Least-Squares (CLS) supervised feature reduction method to prior methods Maximum Co-Variance (MCV), Ordinary Least-Square (OLS), and unsupervised Principal Component Transformation (PCT). The comparison has been done on the same database and the same detector used in previous experiments, with the only change that the features used are $\mathcal{CF}^*$. To study the generalization of supervised feature-extraction methods (MCV, OLS, and CLS) to unknown steganographic algorithms, we used five different steganography methods, testing each combination of different methods

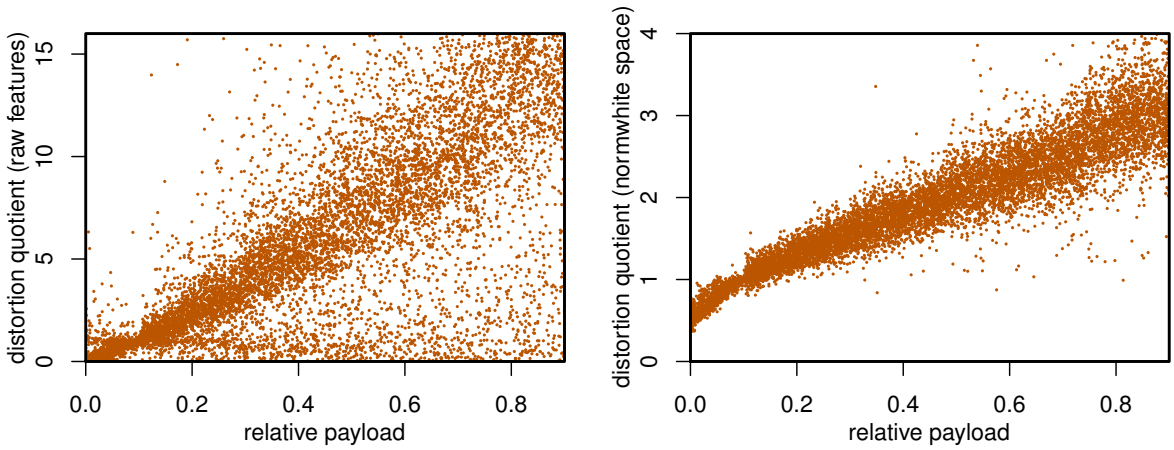

Figure 2: (Taken from [3].)Left: relative payload versus feature distortion in individual images. Right: the same, but the distortion calculated for features transformed by PCT.

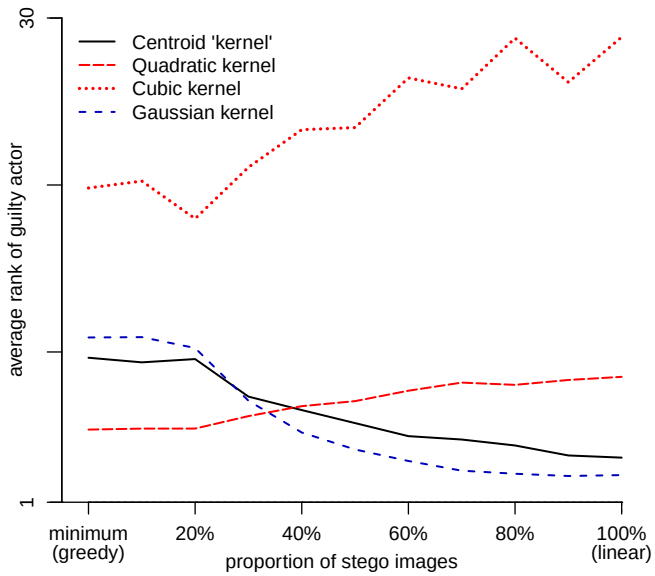

Figure 3: (Taken from [6].) The x-axis denotes strategies interpolating smoothly between greedy (left) and linear (right), the y-axis the average rank of the truly guilty actor. The different lines indicate different kernels. In this experiment there are 100 images from each of 100 actors, the payload is 0.2 bpnc, and the embedding is by nsF5, the features were whitened, and the LOF parameter is $k = 10$.

for feature extraction and evaluation. In Table 1 each row corresponds to the guilty actor using a different embedding algorithm, the columns corresponds to the algorithm used in feature extraction. The PCT, MCV, OLS, and CLS methods are compared for every combination of feature-extraction/embedding algorithm, and the one with best performance (measured by average rank of guilty actor) highlighted in boldface. For the MCV and CLS methods, the number of projections selected is the value up to 50 which gives the best testing accuracy and is displayed in parentheses below the average rank; OLS always creates one projection.

From Table 1 we can see that the unsupervised PCT does not perform well, especially if steganographer uses the greedy strategy, and its accuracy is close to random guessing. Of the supervised methods, the proposed CLS performs the best as it is (a) robust to cover mismatch which is demonstrated by being superior in cases where feature-extraction and embedding algorithm match and (b) it does not overfit to the algorithm used in the feature extraction.

Mismatch mitigation

Table 2 shows the performance, in terms of binary detection error, for various methods that we proposed to ameliorate cover source mismatch in [5]. Rows correspond to different ways of pooling the votes from the ensemble of linear FLD base learners, which have been trained on the first 6000 of each actor's images. "Equal weight" refers to majority aggregation of FLD votes; "Weight by $cos\alpha_i$" / "Weight by s_i" refers to weighted voting, with weights determined by the base learners whose projection vector most closely matches that of the observed image's trajectory under re-embedding. (To understand the formulae for $\cos\alpha_i$ and s_i, see [5].) "Only $\arg\max\cos\alpha_i$ " / "Only $\arg\max s_i$" corresponds to to ensemble returning the vote of the only classifier with the best match. The first half of the table shows the results without any calibration while the second shows the results when images of each actor were centred by subtracting mean of its cover images.

The errors in the second part of the table are lower than those in the first half, demonstrating that centring on covers does mitigate cover mismatch. This suggests that part of the cover mismatch is due to different actors being centred on different parts of the feature space, but the embedding operation shifts images in the same direction. Weighted voting helps on non-centered images while on centered images it is not as effective, which suggests that the trajectory of features under re-embedding can be used as a proxy for knowing the location of cover centres. These results are similar whether one considers accuracy on individual actors' testing data, or by the aggregate measures we proposed in [5].

In our pilot study on abstaining classifiers, we implemented one-class detectors as one-class support vector machines (1-SVM) with Gaussian kernel on a $\mathcal{CF}^*$ feature set reduced by CLS to 20 dimensions. The accuracy of this detector has been compared to that of standard two-class SVM (2-SVM) by benchmarks described earlier. The comparison has been carried on the same data as before. The hyper-parameters of the 1-SVMs can be optimized with respect to metrics for matched or mismatched data.

Error and accuracy rates estimated from the training data are summarized in Table **??**. First, the abstaining classifier frequently outputs "don't knows" (high rates of P_{DP} and P_{DN}) on mismatched sources and only provides a decision on images where it is sure, which is demonstrated by having lower P_{FP} and P_{FN} than the standard binary classifier. Interestingly, the abstaining classifier outputs frequently "don't knows" on matched cases, which allows a very small probability of error. This is an advantage in the case of pooled steganalysis, as the interest is to have a low false positive rate.

Key exhaustion

Figure 4 demonstrates the efficacy of the Bayesian inference based key-exhaustion attack described in [8]. The experiment used images embedded by Outguess with a random payload and PF-274 features, and was repeated 1000 times with up to 50 different stego images selected from 9000. The graph shows how the entropy of the keyspace decreases with the number of observed images, and demonstrates that if steganographer uses on average six images embedded with the same key, the key can be uniquely determined by the proposed attack. Results for other scenarios can be found in [8].

	PCT	MCV trained on					OLS trained on					CLS trained on				
		F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	40.3	23.4 (4)	28.4 (4)	31.6 (5)	30.7 (50)	28.9 (47)	22.2	4.4	28.1	41.0	25.6	**1.6** (1)	**1.9** (1)	**8.8** (1)	**6.6** (4)	**4.5** (3)
nsF5	38.0	25.3 (4)	26.6 (4)	29.8 (5)	31.8 (42)	28.9 (48)	31.6	5.8	30.2	41.2	33.8	**1.8** (1)	**2.1** (1)	**10.1** (1)	**10.9** (4)	**10.5** (3)
JP	38.4	34.7 (4)	36.2 (4)	27.2 (5)	33.8 (50)	34.4 (47)	38.8	35.5	6.9	47.1	47.6	**8.9** (1)	**7.2** (2)	**1.7** (1)	**15.5** (2)	**10.5** (2)
OG	26.5	23.2 (44)	19.2 (50)	18.4 (50)	31.6 (4)	3.2 (6)	26.4	23.0	44.9	2.4	1.3	**3.7** (1)	**3.0** (6)	**11.8** (2)	**1.2** (1)	**1.1** (1)
SH	23.0	21.9 (44)	18.7 (50)	17.5 (50)	30.3 (4)	2.6 (6)	31.4	31.7	45.9	3.5	1.3	**5.2** (1)	**3.2** (6)	**9.1** (2)	**1.2** (1)	**1.1** (1)

(a) The guilty actor embeds total payload of 0.1bpnc, using the greedy strategy.

	PCT	MCV trained on					OLS trained on					CLS trained on				
		F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	9.5	27.1 (50)	22.4 (48)	**20.0** (49)	18.7 (50)	18.1 (50)	55.9	2.4	41.6	48.3	24.0	**1.8** (1)	**1.8** (1)	25.9 (4)	**9.7** (50)	**5.4** (2)
nsF5	23.1	14.8 (4)	15.2 (4)	18.3 (5)	22.4 (46)	23.0 (48)	7.9	**1.4**	8.9	23.4	21.3	**1.4** (1)	1.5 (1)	**2.2** (1)	**5.3** (2)	**4.3** (2)
JP	16.2	33.8 (50)	30.2 (50)	37.4 (22)	30.9 (48)	35.3 (49)	29.3	36.7	21.3	44.1	39.3	**15.6** (3)	**11.4** (5)	**1.5** (4)	**11.2** (50)	**17.6** (44)
OG	5.7	25.8 (50)	18.4 (50)	13.5 (50)	26.9 (3)	1.6 (6)	13.6	18.3	35.4	1.7	1.2	**3.1** (1)	**2.6** (6)	**4.9** (2)	**1.1** (1)	**1.1** (1)
SH	4.7	19.6 (50)	15.0 (50)	11.2 (49)	27.5 (3)	1.5 (6)	16.2	15.6	31.4	1.7	1.3	**3.1** (3)	**2.7** (5)	**2.6** (1)	**1.1** (1)	**1.1** (1)

(b) The guilty actor embeds total payload of 0.1bpnc, using the linear strategy.

Table 1: (Taken from [7].) Average rank of guilty actor out of 100 actors (lower is better; 1.0 represents perfect detection and 50.5 random guessing), when $\mathcal{CF}^*$ features are condensed using PCT, MCV, OLS, and CLS methods. The PCT method is unsupervised and extracts all projections with eigenvalue at least 0.01, and the others create condensed features using data informed by each of the five different embedding algorithms (F5, nsF5, JPHide&Seek, OutGuess, Steghide) separately. The PCT, MCV, OLS, and CLS methods are compared for every combination of training/embedding algorithm, and the one with best performance highlighted in boldface.

	Testing actor									Aggregate error		
Voting	1	2	3	4	5	6	7	8	9	μ_1	μ_2	μ_∞
Equal weight	0.0151	0.0060	0.1366	0.0921	0.0064	0.0586	0.0261	0.0147	0.0394	0.0439	0.0627	0.1366
Weight by $\cos\alpha_i$	0.0128	0.0054	0.1160	0.0978	0.0074	0.0630	0.0259	0.0197	0.0416	0.0433	0.0593	0.1160
Weight by s_i	0.0133	0.0058	0.1109	0.0990	0.0079	0.0631	0.0251	0.0208	0.0419	0.0431	0.0584	0.1109
Only $\arg\max\cos\alpha_i$	0.0277	0.0140	0.1406	0.1156	0.0166	0.1041	0.0526	0.0409	0.0721	0.0649	0.0796	0.1406
Only $\arg\max s_i$	0.0412	0.0115	0.1201	0.0795	0.0160	0.0989	0.0342	0.0338	0.0680	0.0559	0.0675	0.1201
after centering:												
Equal weight	0.0274	0.0109	0.0776	0.0744	0.0058	0.0584	0.0344	0.0109	0.0519	0.0391	0.0479	0.0776
Weight by $\cos\alpha_i$	0.0201	0.0075	0.0806	0.0718	0.0059	0.0609	0.0289	0.0139	0.0454	0.0372	0.0468	0.0806
Weight by s_i	0.0235	0.0088	0.0790	0.0697	0.0059	0.0601	0.0285	0.0128	0.0470	0.0373	0.0463	0.0790
Only $\arg\max\cos\alpha_i$	0.0275	0.0135	0.1177	0.1156	0.0173	0.1036	0.0387	0.0352	0.0719	0.0601	0.0737	0.1177
Only $\arg\max s_i$	0.0403	0.0140	0.1101	0.0722	0.0169	0.0958	0.0397	0.0253	0.0754	0.0544	0.0648	0.1101

Table 2: (Taken from [5].) Error, when eight detectors are used to classify images on the remaining ninth actor. The first column denotes the voting strategy of the ensemble: a baseline method of equal weight, weighting all classifiers according to $\cos\alpha_i$ or s_i, or using the one classifier with best $\cos\alpha_i$ or s_i.

	matched cases						mismatched cases					
	pooled error rates				d_p	d_∞	pooled error rates				d_p	d_∞
	P_{FP}	P_{FN}	P_{DP}	P_{DN}			P_{FP}	P_{FN}	P_{DP}	P_{DN}		
2-SVMs	0.020	0.022	0.000	0.000	3.345	1.940	0.165	0.059	0.000	0.000	1.279	0.465
1-SVMs (a)	0.004	0.002	0.192	0.046	8.176	4.785	0.041	0.008	0.186	0.356	2.258	0.926
1-SVMs (b)	0.006	0.006	0.116	0.082	5.814	4.542	0.028	0.010	0.114	0.425	2.472	1.017

(a) optimized for matched data

(b) optimized for mismatched data

Table 3: (Taken from [5].) Comparison of 2-SVMs and 1-SVMs by pooled error rates and aggregated deflection metrics which appropriately value 'don't know' cases.

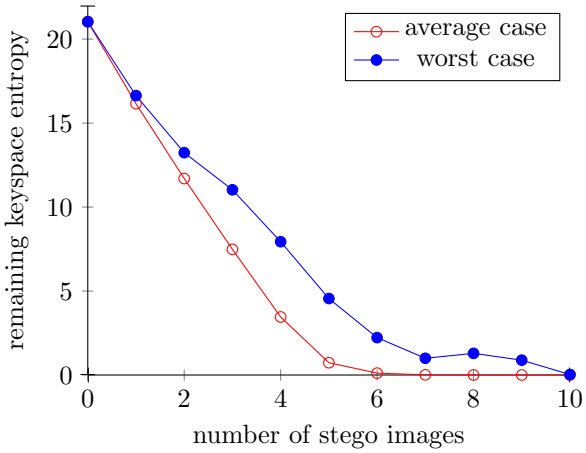

Figure 4: (Taken from [8].) Entropy of keyspace (y axis) after Bayesian Key Inference using message length metadata from multiple images (x axis) embedded with the same key.

5 Conclusions

The main goal of this project was to create and evaluate a universal steganalyzer, which identifies an actor who is guilty of communicating secret messages by using steganography. Our method turns the scale of the problem, where there are many actors each emitting many objects, to our advantage by using the innocent majority to calibrate the guilty. On a dataset of 800000 images from 4000 genuine social network actors, we have demonstrated that our steganalyzer can identify the guilty actor, or at least rank them as highly suspicious, without any prior knowledge of the embedding algorithm. The universality was demonstrated by detecting publicly available steganographic algorithms based on different embedding paradigms.

In course of the work on the project we have discovered and described a game between the steganographer (who choses a strategy to spread payload between multiple objects) and the detector (who chooses a kernel for the MMD calculation). We also discovered that there are problems when steganalytic features have large dimension. We have successfully solved this problem by deriving a novel method (called CLS) for extracting directions in feature space that are robust to change in the distribution of cover messages, but still contain evidence of steganography. CLS has greatly increased the accuracy of our universal steganalyzer without impairing its capability to detect new algorithms.

Diversity between cover images taken by different cameras and actors is the main factor of falsely accusing innocent users. Since we have focused on steganalysis for real-world use, we have studied the nature of this diversity and its effect on accuracy (particularly false positives). We proposed several solutions for different scenarios, decreasing the error by up to 50% of its previous value. In the same study we have presented a steganalyzer that can output "don't know", which again lowers the false positive rate since the steganalyzer does not answer on data it has never seen before, or on those where evidence is ambivalent.

Our steganalyzer represents the first step, providing intelligence on the guilty actor. The next logical step would be to prove that he is actually guilty, which can be done by breaking the key and extracting the message. We have created and demonstrated a method that does this, provided that the steganalyst possesses a few images embedded by the same key and he knows which algorithm has been used. Both conditions are not completely unrealistic, but lie further down the forensic chain than the rest of our work.

Our methods are likely to apply beyond the speciality of pure steganalysis. For example, it may be possible to perform blind intrusion detection in a similar manner. And we predict that the usefulness of CLS can be extended to many other situations when we test a simple null hypothesis (nothing is happening) against a compound alternative (something is happening, but we do not know how much).

Allocation of work

The work has been collaborative throughout, with ideas generated during visits of Tomáš Pevný to Andrew D. Ker. All papers were coauthored. Broadly speaking, code in MATLAB and for large-scale experiments in R was written by Pevný; high-performance code in C or CUDA, scripts for visualization in R, and scripts to crawl images from social networking sites, were written by Ker. The experiments were performed on computer systems in both Prague and Oxford, on a small cluster administered by Ker in Oxford, part of whose upkeep was funded by this grant, and on GPU and large clusters at Oxford's Advanced Research Computing facility.

References

[1] Andrew D. Ker. Implementing the projected spatial rich features on a gpu. In Adnan. M. Alattar, Nasir D. Memon, and Chad D. Heitzenrater, editors, *Media Watermarking, Security, and Forensics 2014*, volume 9028 of *Proc. SPIE*, pages 90280K–90280K–10. SPIE, 2014.

[2] Andrew D. Ker, Patrick Bas, Rainer Böhme, Remi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevný. Moving steganography and steganalysis from the laboratory into the real world. In *Proc. 1st ACM Workshop on Information Hiding and Multimedia Security*, pages 45–58. ACM, 2013.

[3] Andrew D. Ker and Tomáš Pevný. Batch steganography in the real world. In *Proceedings of the Workshop on Multimedia and Security*, MM&Sec '12, pages 1–10. ACM, 2012.

[4] Andrew D. Ker and Tomáš Pevný. Identifying a steganographer in realistic and heterogeneous data sets. In N.D. Memon, A.M. Alattar, and E.J. Delp III, editors, *Media Watermarking, Security, and Forensics XIV*, volume 8303 of *Proc. SPIE*, pages 83030N–83030N–13. SPIE, 2012.

[5] Andrew D. Ker and Tomáš Pevný. A mishmash of methods for mitigating the model mismatch mess. In Adnan. M. Alattar, Nasir D. Memon, and Chad D. Heitzenrater, editors, *Media Watermarking, Security, and Forensics 2014*, volume 9028 of *Proc. SPIE*, pages 90280I–90280I–15. SPIE, 2014.

[6] Andrew D. Ker and Tomáš Pevný. The steganographer is the outlier: Realistic large-scale steganalysis. *Information Forensics and Security, IEEE Transactions on*, 9(9):1424–1435, Sept 2014.

[7] Tomáš Pevný and Andrew D. Ker. The challenges of rich features in universal steganalysis. In Adnan. M. Alattar, Nasir D. Memon, and Chad D. Heitzenrater, editors, *Media Watermarking, Security, and Forensics 2013*, volume 8665 of *Proc. SPIE*, pages 86650M–86650M–15, 2013.

[8] Tomáš Pevný and Andrew D. Ker. Steganographic key leakage through payload metadata. In *Proceedings of the 2Nd ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '14, pages 109–114. ACM, 2014.

6 List of Symbols, Abbreviations, and Acronyms

1-SVM, 2-SVM one-class and two-class support vector machine classifier.

$\{x_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$ features from two actors (after whitening and normalization).

$\mathbf{X}^c$ matrix of feature vectors from cover images.

$\mathbf{X}^s$ matrix of feature vectors from stego images.

$\mathbf{Y}^s \in \mathbb{R}^{n,1}$ payload rates of images in $\mathbf{X}^s$.

$\kappa(x, y) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ a kernel function for the MMD calculation.

$d : P \times P \to [0, \infty)$ metric defined on space P.

d_p, d_∞ metrics for the accuracy of abstaining classifiers.

μ_1, μ_2, μ_∞ metrics for aggregate accuracy of classifiers trained and testing in different combinations.

$lof_k(p)$ the Local Outlier Factor of point p.

P_{FP}, P_{FP}, P_{E} false positive, negative, and equal-prior error rate of binary detectors.

P_{DP}, P_{DN} 'don't know' rates of abstaining classifiers.

$p(k)$, $p(k|y)$ prior and posterior probability of a key k, given evidence from one stego object y.

$\text{P}(y|x)$ distribution of the estimated payload size y, given an object with true payload x.

bpnc Bits per non-zero DCT coefficient (measure of payload size).

$\mathcal{CF}^*$ A set of 7850-dimensional steganalysis features.

CLS Calibrated Least Squares, supervised method of dimensionality reduction.

CUDA Compute Unified Device Architecture.

DCT Discrete Cosine Transform.

F5, nsF5, OutGuess, StegHide, JpHide&Seek Steganographic methods for JPEG images.

FLD Fisher Linear Discriminant.

GPU Graphical Processing Unit.

JPEG Image format for storing digital images, Joint photographic experts group.

JRM JPEG Rich Model. A set of 22510-dimensional steganalysis features.

KDF A key derivation function.

MCV Maximum CoVariance.

MMD Maximum Mean Discrepancy

OLS Ordinary Least Squares regression.

PCT Principal Component Transformation.

PF-274 A set of 274-dimensional steganalysis features.

PSRM Projected Spatial Rich Model, a set of 12870-dimensional steganalysis features.

TFLOP One trillion floating-point operations.