Locating Steganographic Payload via WS Residuals

Andrew D. Ker
Oxford University Computing Laboratory
Parks Road
Oxford OX1 3QD, UK
adk@comlab.ox.ac.uk

ABSTRACT

The literature now contains a number of highly-sensitive detectors for LSB replacement steganography in digital images. They can also estimate the size of the embedded payload, but cannot locate it. In this short paper we demonstrate that the Weighted Stego-image (WS) steganalysis method can be adapted to locate payload, if a large number of images have the payload embedded in the same locations. Such a situation is plausible if the same embedding key is reused for different images, and the technique presented here may be of use to forensic investigators. As long as a few hundred stego images are available, near-perfect location of the payloads can be achieved.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures—*information hiding*

General Terms

Security, Algorithms

Keywords

Forensics, Steganalysis, Weighted Stego-Image

1. INTRODUCTION

There can be no doubt that replacement of least significant bits (LSBs) in digital images is a poor choice for steganography. Nonetheless, it remains popular in free steganography software, perhaps because of the mistaken assumption that visual imperceptibility implies undetectability.

Broadly, the literature contains two leading classes of detector for LSB replacement. The first, termed the *structural* detectors in [5], includes payload estimators found in [1, 3, 5, 6, 7, 9]; they analyse explicitly the combinatorial structure of LSB replacement in pixel groups. The second, known as the *Weighted Stego-image* (WS) detectors, are found in [2, 8], and involve filtering a stego image to estimate the cover, before using some properties of bit flipping. At the present time, the most recent WS detectors and payload estimators seem somewhat more accurate than the most recent structural detectors, but both exhibit astonishingly sensitive performance: depending on cover type, payloads using only of the order of 1% of capacity can often be detected with high reliability.

However, none of these detectors go beyond estimating the size of the payload (nor, to the author's knowledge, do less-sensitive detectors found in other literature): they cannot locate or determine the payload. It is somewhat curious that it is possible to make near-perfect estimates of the number of payload-carrying pixels without learning anything about which pixels they are.

Our aim here is to adapt the WS method to locate payload. Our method will not work on a single image, but instead assumes that the steganalyst possesses a number of stego images each containing payload at the same locations. We argue that such a scenario is not implausible, for example if the different stego objects are the same size, each contain the same amount of payload, and the same embedding key was used. By applying the WS method in an unusual way, it will be possible to determine with high accuracy which pixels carry the payload.

One other technique for locating payload is found in the literature [4]; there, the steganographic key space is tested exhaustively and stego-signatures in the histogram are used to determine the correct key from a single stego image. However, this method requires that the complete steganographic scheme is known (and the key space must not be too large). Our technique requires many stego images which (by reason of using the same embedding key, or by defect in the embedding method) locate the payload in the same pixels, but it does not require us to know anything more than that LSB replacement was used. This may have potential applications in image forensics: information about payload location could be a key step in identifying the embedding software used, with a technique such as [4] applied subsequently. Location of payload pixels is the first step to the eventual aim of decoding the payload.

The paper is structured as follows. In Sect. 2 we will briefly summarise the key points of the WS method, but alter the presentation to highlight what we call the *WS residuals*. In Sect. 3 we demonstrate how the residuals can be used to locate payload, and test the locator in Sect. 4. Sect. 5 considers the limitations of this technique and suggests directions for further research.

MM&Sec'08, September 22–23, 2008, Oxford, United Kingdom.

2. RESIDUALS AND THE WS METHOD

The Weighted Stego-image method was first described in [2] and substantially re-engineered and improved in [8]. We will present the core of the method in a slightly different way, making the *residuals* explicit.

We first fix the notation, to be used throughout the paper. Let us suppose that a cover image consists of a vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ of n pixel intensities, and that a corresponding stego image $\boldsymbol{s} = (s_1, \ldots, s_n)$ is created by replacing the LSBs of proportion p of the cover pixels; the total payload size is therefore np. Throughout the paper we will use the notation $\widetilde{x}$ to indicate the integer x with LSB flipped (more usually $\bar{x}$, but we want to avoid confusion with a sample mean, used extensively here).

The first step of the WS method is to estimate the cover image by filtering the stego image; we use the notation $\widehat{\boldsymbol{c}}$ for the estimate of $\boldsymbol{c}$ obtained from $\boldsymbol{s}$. In [2], the estimate of each cover pixel is simply the average of the neighbouring four stego pixels, but this is generalized in [8] to convolution by an arbitrary linear filter, thus

$$\widehat{\boldsymbol{c}} = \boldsymbol{f} * \boldsymbol{s} \tag{1}$$

where $\boldsymbol{f}$ is the filter. (This constitutes a slight abuse of notation, as $*$ is intended to indicate a 2-dimensional convolution taking into account horizontal and vertical structure in the image, even though we have modelled images as 1-dimensional vectors).

In this work we will define the vector of *residuals*

$$r_i = (s_i - \widetilde{s_i})(s_i - \widehat{c_i})$$

which indicate the difference between stego object and estimated cover, with the sign adjusted to take into account the asymmetry in LSB replacement (even pixels could only be incremented, and odd pixels decremented, by overwriting the LSB). If $\widehat{c_i}$ is an unbiased estimator for c_i, the estimation error is independent of the parity of c_i, and the payload is independent of the cover, then the residuals r_i satisfy

$$\mathrm{E}[r_i] = \begin{cases} 0, & \text{if } s_i = c_i, \\ 1, & \text{if } s_i = \widetilde{c_i}. \end{cases} \tag{2}$$

We can define the *mean residual* $\overline{r} = \frac{1}{n}\sum r_i$; from (2), $2\overline{r}$ is an unbiased estimator for p. The factor of 2 is because, on average, replacing a LSB only flips it with probability 1/2.

The residuals themselves have quite a dispersed distribution, in comparison with a shift of 1 caused by LSB flipping. For a set of images acquired from a digital camera (of which more in Sect. 4), in which no payload was embedded and no LSBs flipped, we computed the residuals for each pixel and each image, and display their histogram in Fig. 1. The observed mean was -0.000179, the standard deviation 5.30 (3 sig. fig.), and the distribution was significantly leptokurtic (fatter tails than Gaussian). But a single digital image is likely to contain hundreds of thousands, or millions, of pixels and so the mean residual $\overline{r}$ will have a much lower variance. This is why sensitive detection, and payload size estimation, of LSB replacement is possible by the WS method.

The version of WS described above is equivalent to the simplest payload estimator in [2]. More sophisticated estimators, with even more accurate payload size estimation, can be found in [2] and [8]: additional techniques include using better pixel predictors than the simple 4-neighbour average above, optimizing the pixel predictor by training it,

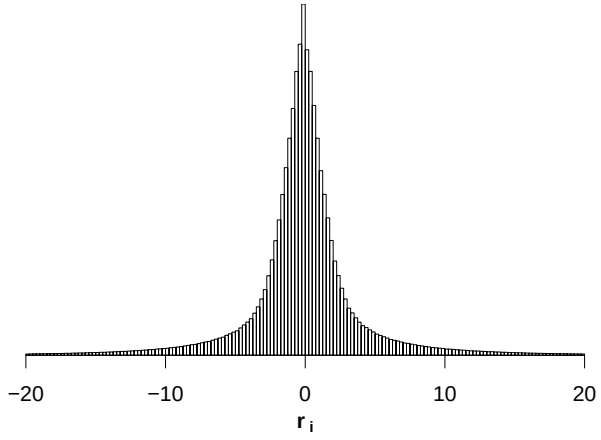

Figure 1: Histogram of WS residuals for pixels with no payload.

forming a *weighted* WS estimator which amounts to taking a weighted average residual – with areas of more confident cover prediction being given higher weights – and correcting for bias caused by parity co-occurrence between neighbouring cover pixels. We will not repeat these here, and some of them are not applicable to payload location, but we will return briefly to improved pixel predictors and weighting in Sect. 4.

3. LOCATING PAYLOAD IN MULTIPLE COVERS

Suppose that we have a number of stego images, which contain different payloads but locate the payloads at the same pixel positions. This is not completely inconceivable: some embedding schemes (particularly those foolish enough to choose LSB replacement) use fixed payload locations, and even when the location is varied by a secret key shared between steganographer and recipient it is quite possible that the same key is used for a batch of communications, which would therefore contain payload in the same locations. We will attempt to identify the location of these payload-carrying pixels, by summing WS residuals *between*, instead of *within*, images.

Suppose that there are N images, each of n pixels. Following the WS method, we can estimate each cover image by filtering the corresponding stego images. Then denote the residual of pixel i in image j by

$$r_{ij} = (s_{ij} - \widetilde{s_{ij}})(s_{ij} - \widehat{c_{ij}}).$$

The conventional WS method estimates the proportion of flipped LSBs in image j by taking the mean $\overline{r_{\cdot j}} = \frac{1}{n}\sum_{i=1}^{n} r_{ij}$. Instead, we can estimate *the number of images in which pixel i is flipped* by

$$\overline{r_{i\cdot}} = \frac{1}{N}\sum_{j=1}^{N} r_{ij}.$$

Given (2), if pixel i is not used for payload (by assumption, it will not have been flipped in any of the stego images) then $\mathrm{E}[\overline{r_{i\cdot}}] = 0$. On the other hand, if pixel i is used for payload (by assumption, it will be overwritten in each of the images) then $\mathrm{E}[\overline{r_{i\cdot}}] = 0.5$. Of course, the observations of $\overline{r_{i\cdot}}$ will

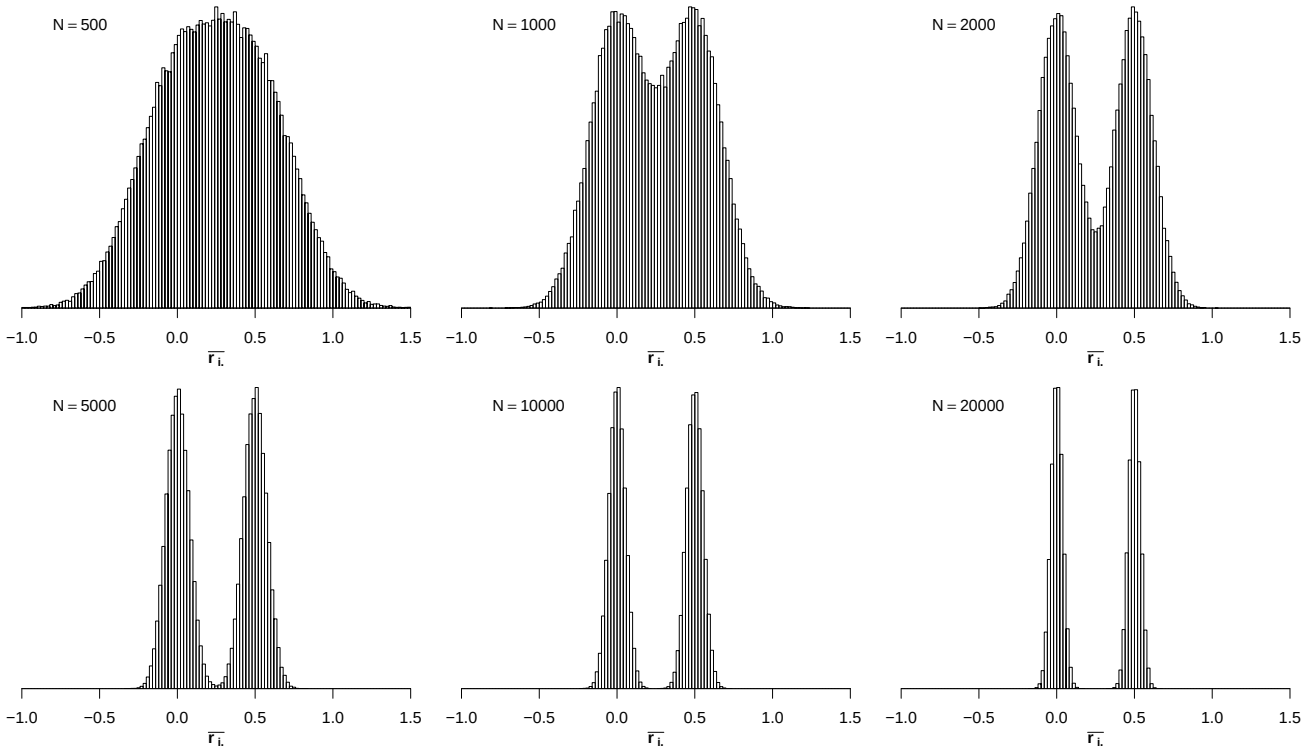

Figure 2: Histograms of the mean residuals of each pixel, when half of the pixels carry payload, for six different values of N.

inherit high variance from the residuals unless N is large, and we cannot reasonably expect N to be of the order of 10^6 in the same way as n in standard WS. However, neither do we need as low a variance if our only aim is to distinguish $\overline{r_{i\cdot}} \approx 0$ from $\overline{r_{i\cdot}} \approx 0.5$. So, given sufficient stego images, it should be possible to separate the payload-carrying pixel locations from the rest.

A classification of which pixel locations do contain payload can be made in a number of ways. One could take the grand mean $\overline{r_{\cdot\cdot}} = \frac{1}{nN}\sum_{i,j} r_{ij}$ to estimate the number of payload locations by $M = 2n\overline{r_{\cdot\cdot}}$, and the M pixel locations with the highest mean residual $\overline{r_{i\cdot}}$ can be identified as containing payload. Alternatively, given symmetry of the residuals, locations with $\overline{r_{i\cdot}} > 0.25$ could be identified as containing payload.

Note that our assumptions include that the *amount* of payload in each image is also fixed. However, the method could also be adapted to unequal payload sizes, if payload is placed into a fixed sequence of locations: those locations at the start of the sequence would have the highest values of $\mathrm{E}[\overline{r_{i\cdot}}]$, while locations at the end would have the lowest. Simply ranking the observed values of $\overline{r_{i\cdot}}$ would estimate the pseudorandom path, but we will not pursue this here.

4. EXPERIMENTAL RESULTS

We now give some experimental data to measure how well this adapted WS technique locates payload, and to determine the necessary number of images to obtain reliable results. We began with a set of 1600 never-compressed digital images, 2000×1500 pixels, converted from RAW colour images obtained from a digital camera using the manufacturer's standard RAW-to-TIFF conversion software and then reduced to grayscale by taking the luminosity. However, 1600 images is not enough to test large values of N, and 2000×1500 pixel images are rather large if we need to store all residuals of each pixel, so we created a set of 20000 smaller images by repeatedly cropping random 400×300 regions from the larger originals. There will be some overlap between a few of these images, but that should not be a significant factor in these experiments. We chose a fixed set of pixels to carry payload of 50% capacity (i.e. 60000 locations), and there embedded a random payload by LSB replacement in each image.

For six different values of N, we selected N images at random and computed residuals for each pixel and each image. The cover predictor was the simple average of four neighbours, described in [2]. Finally, we computed the mean-per-pixel residuals $\overline{r_{i\cdot}}$, and display their histograms in Fig. 2. Observe that there is much noise in these residuals for small values for N, but for N at least 1000 distinct peaks at 0 and 0.5 – corresponding to pixels without and with payload – begin to appear. At $N = 10000$ there is complete separation between these two cases and so it is discovered exactly which pixels carried the payload.

To evaluate the accuracy of classification, we chose the simple method of identifying all pixels with $\overline{r_{i\cdot}} > 0.25$ as those carrying payload. We then compared our estimate against the true set of 60000 payload-carrying pixels. For some different values of N, we display accuracy (in terms of true positive, false positive, and false negative) in Tab. 1. The classification is near-perfect for $N = 5000$ and perfect for N at least 10000. Even for smaller values of N the clas-

Table 1: Accuracy of payload location, for seven different values of N. TP = true positives, FP = false positives (pixels incorrectly classified as carrying payload), FN = false negatives (missed pixels).

N	TP	FP	FN
100	40717	19042	19283
200	43850	15927	16150
500	49262	10674	10738
1000	55170	4861	4830
2000	58672	1290	1328
5000	59956	33	44
10000	60000	0	0

sification is more right than wrong, but it appears that this method is of limited use unless a forensic investigator has thousands of images, all embedded using the same key.

However, we can do better by boosting the performance of the WS method, adopting some of the methods of [8]. Instead of predicting the cover image using the simple fixed filter (1) we adjust the filter according to the image under consideration: it is trained on each individual stego image to determine the linear filter which best predicts the stego image itself. Following [8], we used a 5×5 filter pattern with horizontal, vertical, and diagonal symmetry (for correctness of (2), the central value of the filter must be fixed at zero). Second, we take into account a varying level of confidence in the predictor by weighting the residuals: each pixel receives a weight factor w_{ij} which depends on the local variance of the neighbourhood of the estimated pixel (in [8] the weights are chosen by $w_{ij} = 1/(5 + \sigma_{ij}^2)$, where σ_{ij}^2 is the local variance weighted by the same filter used by the predictor). Then the mean residual is a weighted sum: $\overline{r_{i\cdot}} = \left[\sum_{j=1}^{N} w_{ij} r_{ij}\right] / \left[\sum_{j=1}^{N} w_{ij}\right]$. In [8] it is shown that these changes reduce the variance of payload size estimates.

The accuracy of payload location using the improved WS is shown in Tab. 2, for comparison with Tab. 1. It is apparent that the variance-reducing methods from [8] make a huge improvement to this application too. In fact, perfect classification is achieved with $N \geq 1500$, and 99% accurate classification for $N \geq 600$. Even $N = 100$ yields mostly correct classification.

Table 2: Accuracy of payload location, when the WS method is enhanced by a trained pixel predictor, and weighting.

N	TP	FP	FN
100	51439	8505	8561
200	55289	4435	4711
500	59089	936	911
1000	59959	48	41
2000	60000	0	0
5000	60000	0	0
10000	60000	0	0

The large difference between results in Tabs. 2 and 1 is because, more than simply reducing variance, the techniques of [8] dramatically cut down outliers in the residuals. A further technique in [8], correction for additive bias, is not applicable to this case because the bias is only caused by parity co-occurrence between neighbouring pixels: when the residuals summed are from different images, it is not reasonable to suppose that such correlations exist.

5. CONCLUSIONS

The aim of this paper has been to identify the WS residuals, and to demonstrate that they can be used to locate LSB replacement payload if enough stego images place it in the same pixels. With the WS method presented in the form in Sect. 2, the payload location method is almost absurdly simple yet, as long as the steganalyst has a few hundred images, the payload can be located almost precisely. We have also demonstrated that enhancements to the WS payload size estimator also improve accuracy of the location estimator. Knowledge of the payload location might help the investigator to apply specialised detectors (e.g. for sequentially-placed payload [8]) or to determine the exact embedding software, with the eventual aim of extracting the payload itself.

It is not completely implausible to imagine that such evidence might be available to a forensic investigator, as any steganographer sufficiently ignorant to use LSB replacement could make further mistakes by placing payload nonrandomly or reusing an embedding key. It has long been known that re-using secret keys can compromise the security of cryptosystems, and it is also known that digital watermarks can be estimated and removed if the same watermark is used in multiple objects. The primary contribution of this work is to demonstrate that something similar is true in steganography: even when the payloads are different, their locations must not be kept constant.

As well as needing a large number of stego images, this technique is also limited by dependence on LSB replacement embedding. However, it may be possible to extend the technique to other spatial-domain steganography (perhaps LSB matching): although we could not expect to see residuals with higher *mean* at pixels where alternative embedding was used, we might observe higher *variance*. However, a detector based on this property is likely to be weak.

More generally, we could look for correlations between residuals in different stego images. Even if the location of the payload is not identical, this could tell us if there are some similarities between payload locations in different images. Searching for correlations between large numbers of vectors of huge dimensionality would be a challenge with a data mining perspective.

6. ACKNOWLEDGMENTS

The author is a Royal Society University Research Fellow.

7. REFERENCES

[1] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via sample pair analysis. *IEEE*

Transactions on Signal Processing, 51(7):1995–2007, 2003.

[2] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In *Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306 of *Proc. SPIE*, pages 23–34, 2004.

[3] J. Fridrich, M. Goljan, and R. Du. Detecting LSB steganography in color and grayscale images. *IEEE Multimedia*, 8(4):22–28, 2001.

[4] J. Fridrich, M. Goljan, and D. Soukal. Searching for the stego-key. In *Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306 of *Proc. SPIE*, pages 70–82, 2004.

[5] A. Ker. A general framework for the structural steganalysis of LSB replacement. In *Proc. 7th Information Hiding Workshop*, volume 3727 of *Springer LNCS*, pages 296–311, 2005.

[6] A. Ker. Fourth-order structural steganalysis and analysis of cover assumptions. In *Security, Steganography and Watermarking of Multimedia Contents VIII*, volume 6072 of *Proc. SPIE*, pages 25–38, 2006.

[7] A. Ker. A fusion of maximum likelihood and structural steganalysis. In *Proc. 9th Information Hiding Workshop*, volume 4567 of *Springer LNCS*, pages 204–219, 2007.

[8] A. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In *Security, Forensics, Steganography and Watermarking of Multimedia Contents X*, volume 6819 of *Proc. SPIE*, 2008.

[9] P. Lu, X. Luo, Q. Tang, and L. Shen. An improved sample pairs method for detection of LSB embedding. In *Proc. 6th Information Hiding Workshop*, volume 3200 of *Springer LNCS*, pages 116–127, 2004.