Extracting Hidden Messages in Steganographic Images

By

Tu-Thach Quach

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2014 USA

Denver, CO (Aug 3rd - 6th)

Digital Investigation 11 (2014) S40–S45

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Extracting hidden messages in steganographic images☆

Tu-Thach Quach

Sandia National Laboratories, Albuquerque, NM, USA

Keywords:
Steganography
Steganalysis
Payload location
Message extraction
Embedding key
Logical order

A B S T R A C T

The eventual goal of steganalytic forensic is to extract the hidden messages embedded in steganographic images. A promising technique that addresses this problem partially is steganographic payload location, an approach to reveal the message bits, but not their logical order. It works by finding modified pixels, or residuals, as an artifact of the embedding process. This technique is successful against simple least-significant bit steganography and group-parity steganography. The actual messages, however, remain hidden as no logical order can be inferred from the located payload. This paper establishes an important result addressing this shortcoming: we show that the expected mean residuals contain enough information to logically order the located payload provided that the size of the payload in each stego image is not fixed. The located payload can be ordered as prescribed by the mean residuals to obtain the hidden messages without knowledge of the embedding key, exposing an inherent vulnerability in these embedding algorithms. Experimental results are provided to support our analysis.

Introduction

Digital image steganography hides messages into cover images to produce stego images that appear innocuous to an unintended observer. Popular algorithms include simple least significant bit (LSB) steganography, group-parity steganography, and matrix embedding. To embed the payload, some pixels[1] in the cover image must be modified by an embedding operation so that the resulting stego image conveys the message bits. Two popular operations are LSB replacement and LSB matching. In LSB replacement, odd pixels are decremented and even pixels are incremented. In LSB matching, pixels are either incremented or decremented as necessary. By storing the payload in the LSBs, the resulting stego image looks similar to the cover image, making it difficult to detect by steganalysis detectors. Despite being careful, if the number of changes is sufficiently large, it is still possible to detect stego images as suggested by the square root law of steganographic capacity (Ker, 2007; Filler et al., 2009; Ker, 2009).

Once a stego image is detected, further processing is needed to extract the hidden message, which is inherently a difficult problem. One approach is to search for the embedding key (Fridrich et al., 2004). This technique is applicable when the key space is small and details about the embedding software are known. The advantage is that once the key is found, the hidden message can be extracted readily. An alternative approach that may lead to the eventual message extraction is to locate the payload using a number of stego images where each stego image has the payload at the same locations (Ker, 2008; Ker and Lubenko, 2009; Quach, 2011a, 2011b, 2012, 2014). This could happen if the naive steganographer reuses the embedding key and the stego images are the same size. As an example, the steganographer takes several pictures using his digital

☆ Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

E-mail address: tong@sandia.gov.

[1] We use spatial domain pixels, but the same concept applies to other transformed domains such as quantized JPEG coefficients.

http://dx.doi.org/10.1016/j.diin.2014.05.003
1742-2876/

camera and then embeds messages into these pictures using the same password, e.g., embedding key. Payload location can be used in this scenario.

Payload location can be effective against simple LSB steganography as well as group-parity steganography. Payload location algorithms generally rely on computing the residuals of pixels indicating whether they have been modified in the embedding process. Payload location, however, can only reveal the message bits; the messages themselves remain hidden as we do not know the logical order of the located payload. In order to extract the messages, we need to establish the correct order on the located payload. The current work addresses this fundamental problem. Specifically, we show that if the size of the payload in each stego image is not fixed, the expected mean residuals impose the correct logical order on the located payload. In particular, the expected mean residual of logical payload bit i is strictly greater than the expected mean residual of logical payload bit j for $i < j$. The hidden messages, therefore, can be obtained by ordering the located payload in descending mean residual.

Following a brief overview of steganography and payload location in Section Background, our main result is presented in Section Message extraction. Experimental results to further support our analysis are presented in Section Experiments. Concluding thoughts are provided in Section Discussion and conclusion.

Background

In simple LSB steganography, each payload bit is determined by a single pixel. Given cover image $c = (c_1,\ldots, c_n)$[2] and message $m = (m_1, m_2,\ldots, m_m)$, where $m_i \in \{0, 1\}$ and $m \leq n$, a naive algorithm would use the pixels in their natural order to embed m to produce stego image $s = s_1, s_2,\ldots, s_n$ so that $LSB(s_1) = m_1$, $LSB(s_2) = m_2$, etc., where $LSB(.)$ returns the least significant bit. This is not recommended due to the fact that the introduced stego noise is concentrated in the first m pixels of the stego image, possibly making it easier to detect. More importantly, the message can be extracted by examining the LSBs of the first m pixels. To overcome these undesirable characteristics, a key is often used in the embedding process to distribute the payload over the entire image. Both embedding processes are illustrated in Fig. 1.

The embedding key determines which pixels are used to embed the payload. If the same key is used to embed another payload of m bits into a cover image of n pixels, it is possible that the same set of pixels will be used to carry the payload. This is the fundamental exploit behind payload location. More specifically, given a cover-stego image pair, the residual of pixel i is $r_i = |c_i - s_i|$. In other words, residual r_i indicates whether pixel i has been modified in the embedding process. By averaging the residuals over a number of these cover-stego pairs, we can identify the payload pixels as those with non-zero mean residuals. On average, the payload can be located with approximately $\log_2 m$ pairs (Quach, 2011a, 2011b).

Payload location can also be used against group-parity steganography. In this scheme, each payload bit is determined by a group of k pixels. Specifically, each payload bit is determined by the modulo-2 addition of the LSBs of the k pixels in that group. A payload of m bits requires km pixels. The task of payload location is more complex as it is no longer sufficient to find these km pixels, but also to group them into the correct m groups of k pixels each. The approach consists of constructing and partitioning a weighted complete graph with pixels as nodes. On average, the payload can be located with approximately $8k^2\log_e(km)$ cover-stego image pairs (Quach, 2012).

Both simple LSB embedding and group-parity steganography can be seen as special cases of the popular class of matrix embedding based on Hamming codes (Fridrich and Soukal, 2006). In this class, k pixels are used to embed q bits. The popular (3, 2) matrix embedding uses $k = 3$ pixels to embed $q = 2$ bits. From this perspective, simple LSB embedding is simply (1, 1) matrix embedding and group-parity steganography is $(k, 1)$ matrix embedding. As a consequence, payload location techniques apply to the general class of matrix embedding. The key difference here is that each located group of k pixels represents q bits rather than a single bit.

Message extraction

It is important to recognize that payload location only reveals the message bits, not the message itself. In order to obtain the message, we must arrange the located payload in their logical order. The primary reason why payload location fails to establish this order is due to the fact that it assumes each stego image carries a fixed payload of size m. By relaxing this constraint so that the size of each payload can vary between 1 and m, we show that the mean residuals contain enough information to logically order the located payload to obtain the hidden messages. The next two sub-sections establish this fundamental result for simple LSB steganography and group-parity steganography, respectively.

Simple LSB steganography

Let R_i be the indicator (Bernoulli) random variable for the event that *logical* pixel i needs to be modified to embed message bit m_i. Let $L \sim f_L(l)$ be the random variable corresponding to the size of the payload. We assume that the probability mass function, $f_L(l)$, satisfies the assumption that $f_L(l) > 0$ for all $l \in \{1,\ldots, m\}$ and zero everywhere else. Let $F_L(l) = \sum_{i=1}^{l} f_L(i)$ be the cumulative mass function for L and note that it is strictly increasing in l for $l \in \{1,\ldots, m\}$. We now show that $E[R_i] > E[R_j]$ for all payload pixels i, j where $i < j$. First note that for a payload of size l,

$$p(R_i = 1 | L = l) = \begin{cases} \frac{1}{2}, & \text{if } l \geq i, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

[2] For notational convenience, we represent an image as a one-dimensional sequence.

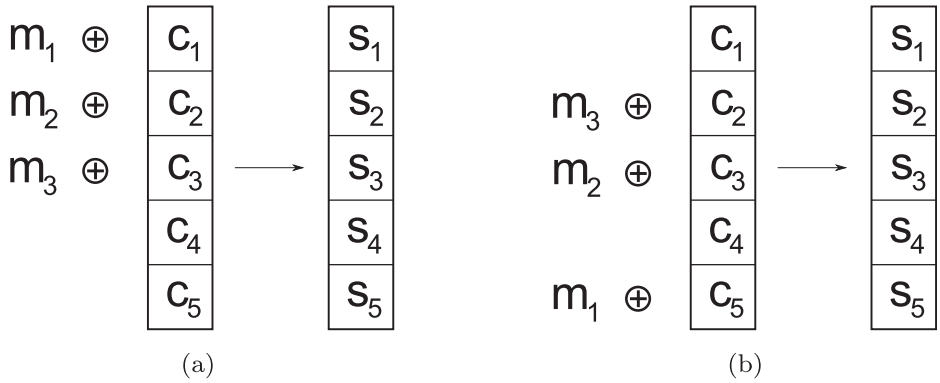

Fig. 1. Simple LSB embedding (a) without and (b) with an embedding key. The illustration shows the cover image consisting of 5 pixels and the message consisting of 3 bits. The embedding key serves to randomize the locations of the pixels that are used to embed the message. The plus sign represents the embedding operation, which is either LSB replacement or matching.

Now,

$$E[R_i] = \sum_{l=1}^{m} p(R_i = 1|L = l) f_L(l) \tag{2}$$

$$= \frac{1}{2} \sum_{l=i}^{m} f_L(l) \tag{3}$$

$$= \frac{1}{2}(1 - F_L(i-1)). \tag{4}$$

Since $F_L(l)$ is strictly increasing in l for $l \in \{1, \ldots, m\}$, it follows that $E[R_i] > E[R_j]$ for all payload pixels i, j where $i < j$. Therefore, to obtain the hidden messages, we simply order the located payload in descending mean residual.

For the specific case where $f_L(l)$ is uniform, $F_L(l) = \frac{l}{m}$ for $l \in \{1, \ldots, m\}$ and

$$E[R_i] = \frac{m + 1 - i}{2m}. \tag{5}$$

Note that the mean residual decreases linearly as a function of i. We will use this scenario in our experiments.

Group-parity steganography

The correct order can also be established for payloads located in group-parity steganography. Let G_i be the group of k pixels that determine message bit m_i. In other words,

$$\sum_{j \in G_i} LSB(s_j) \mod 2 = m_i. \tag{6}$$

Let R_i^j be the indicator random variable for the event that pixel j in group G_i needs to be modified to embed message bit m_i. First note that for a payload of size l,

$$p\left(R_i^j = 1 \middle| L = l\right) = \begin{cases} \frac{1}{2k}, & \text{if } l \geq i \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This is due to the fact that if the parity of the group does not match the message bit, only one of the k pixels needs to be modified. Using the same derivation for (4), we have

$$E\left[R_i^j\right] = \frac{1}{2k}(1 - F_L(i-1)). \tag{8}$$

With a slight abuse of notation, let $R_i = \sum_{j \in G_i} R_i^j$ be the residual of group G_i, we have

$$E[R_i] = \sum_{j \in G_i} E\left[R_i^j\right] = \frac{1}{2}(1 - F_L(i-1)). \tag{9}$$

Since $F_L(l)$ is strictly increasing in l for $l \in \{1, \ldots, m\}$, it follows that $E[R_i] > E[R_j]$ for all payload groups G_i, G_j where $i < j$. The located payload bits obtained from the payload groups can be arranged in descending group mean residual to reveal the hidden messages.

Similarly, if $f_L(l)$ is uniform,

$$E[R_i] = \frac{m + 1 - i}{2m}. \tag{10}$$

As mentioned earlier, group-parity steganography is a special case of matrix embedding. It is straightforward to extend our analysis to this class of embedding algorithms. The only difference is that we can only order groups of q bits instead of each individual bit.

We note that in both cases (simple LSB embedding and group-parity steganography), it is not necessary that $f_L(l) > 0$ for all $l \in \{1, \ldots, m\}$. We can relax this assumption to allow $f_L(l) \geq 0$. In this situation, only a partial order can be obtained.

Cover estimation

The above analysis shows that it is possible to extract the hidden messages provided that the cover images are known so that the residuals can easily be computed. This is the scenario where the forensic analyst has access to the steganographer's computer and other digital media. Even if the naive steganographer deletes the cover images from his computer or digital camera, it is still possible to recover them using file carving techniques (Garfinkel, 2007).

There may be cases, however, where the cover images cannot be recovered. This poses a difficult problem for the analyst as the residuals cannot be computed immediately. In these cases, we have to estimate the cover images. This is the same problem encountered in payload location when the cover images are not available.

The cover estimation problem is formulated as follows. Given stego image s, find the most likely cover image:

$$\widehat{c} = \arg\max_{c} \; p(c|s).$$

A recent cover estimator uses Markov random fields (MRF) and shows good results in locating steganographic payload (Quach, 2014). More specifically, the MRF model estimates the cover image by assigning label y_i to pixel i, $\forall i$. The label indicates whether a pixel has been changed or how it has been changed. It expresses the above conditional distribution in the form of a Gibbs distribution:

$$p(y|s;w) = \frac{1}{Z(s;w)} e^{-E(y|s;w)}, \tag{11}$$

where $w = (w_1, w_2)$ are the model parameters (or weights), Z is the normalization term, and E is an energy function of the form

$$E(y|s;w) = w_1 \sum_{i\in\mathcal{V}} f_i(y_i|s) + w_2 \sum_{ij\in\mathcal{E}} f_{ij}(y_i, y_j|s). \tag{12}$$

Here, $\mathcal{V}$ corresponds to the pixels and $\mathcal{E}$ represents neighboring pixel pairs in a four-connected grid. The terms f_i and f_{ij} are unary and pairwise costs, respectively, that depend on the embedding operation. Intuitively, they can be viewed analogously as the likelihood and prior probabilities, respectively.

For LSB replacement,

$$f_i(y_i|s) = \begin{cases} -\log(1-\rho) & \text{if } y_i = 0, \\ -\log(\rho) & \text{if } y_i = 1. \end{cases} \tag{13}$$

The parameter ρ indicates the proportion of modified pixels and can be estimated using available techniques (Fridrich & Goljan, 2004; Ker and Böhme, 2008; Pevný et al., 2009; Kodovský and Fridrich, 2013).

Denote by $\tilde{s}$ the LSB flipped version of s. Pairwise cost f_{ij} is defined as

$$f_{ij}(y_i, y_j|s) = \begin{cases} -\log p(s_i, s_j) & \text{if } y_i = 0 \text{ and } y_j = 0, \\ -\log p\left(s_i, \widetilde{s_j}\right) & \text{if } y_i = 0 \text{ and } y_j = 1, \\ -\log p\left(\widetilde{s_i}, s_j\right) & \text{if } y_i = 1 \text{ and } y_j = 0, \\ -\log p\left(\widetilde{s_i}, \widetilde{s_j}\right) & \text{if } y_i = 1 \text{ and } y_j = 1. \end{cases} \tag{14}$$

The joint probabilities, p, are learned from known cover images. The obtained labels, y, indicate which pixels have been modified, e.g., $y = 1$ indicates the LSB of pixel i has been flipped. This corresponds precisely to residual r_i.

For LSB matching,

$$f_i(y_i|s) = \begin{cases} -\log(1-\rho) & \text{if } y_i = 0, \\ -\log\left(\frac{\rho}{2}\right) & \text{if } 1 \le s_i + y_i \le 254 \text{ and } y_i \neq 0, \\ -\log(\rho) & \text{if } (s_i = 1 \text{ and } y_i = -1) \text{or } (s_i = 254 \text{ and } y_i = 1), \\ \infty & \text{otherwise.} \end{cases} \tag{15}$$

and

$$f_{ij}(y_i, y_j|s) = -\log p(s_i + y_i, s_j + y_j). \tag{16}$$

Pixel i is modified if $y_i \in \{1, -1\}$. Therefore, $r_i = |y_i|$.

Experiments

We provide the following experiments to further support the above analysis. We use images from the BOSSbase 0.92 database (Filler et al., July 2010), which consists of 9074 grayscale cover images of size 512×512 in the raw PGM format. In each experiment, we generate stego images using the same key. Each stego image carries a random payload of size l uniformly distributed between 1 and 32, e.g., $1 \le l \le 32$. We then compute residuals and order the located payload accordingly, e.g., in descending mean residual. The obtained order is compared against the ground-truth order. To quantify the similarity between the two, we use the minimum edit distance. This metric quantifies the minimum number of operations needed to make two sequences identical. The operations are insertion, deletion, and substitution. If two sequences are identical, the distance is zero. To make the results more precise, we average the minimum edit distances over 10 different runs.

Known cover images

The experiments in this sub-section assumes that the cover images are available, possibly by having access to the steganographer's digital media in combination with using file carving techniques.

We generate two sets of stego images using two different embedding algorithms: simple LSB steganography and group-parity steganography. In both algorithms, we use LSB replacement to modify the pixels. The choice of the embedding operation (LSB replacement or LSB matching) in this scenario does not change our results due to the fact that the cover images are known. In other words, similar results are obtained when we use LSB matching instead of LSB replacement. As we will see later, this is not the case when we have to estimate the cover images.

The mean residuals are computed using these image pairs. We plot the mean residuals of the payload pixels in their logical order using all cover-stego image pairs for simple LSB steganography in Fig. 2. The plot also shows the group mean residuals for group-parity steganography. The lines in both plots, although not perfectly straight, still indicate a linear decrease in mean residual as expected by (5) and (10). This suggests that the hidden messages can be extracted by ordering the located payload in descending mean residual.

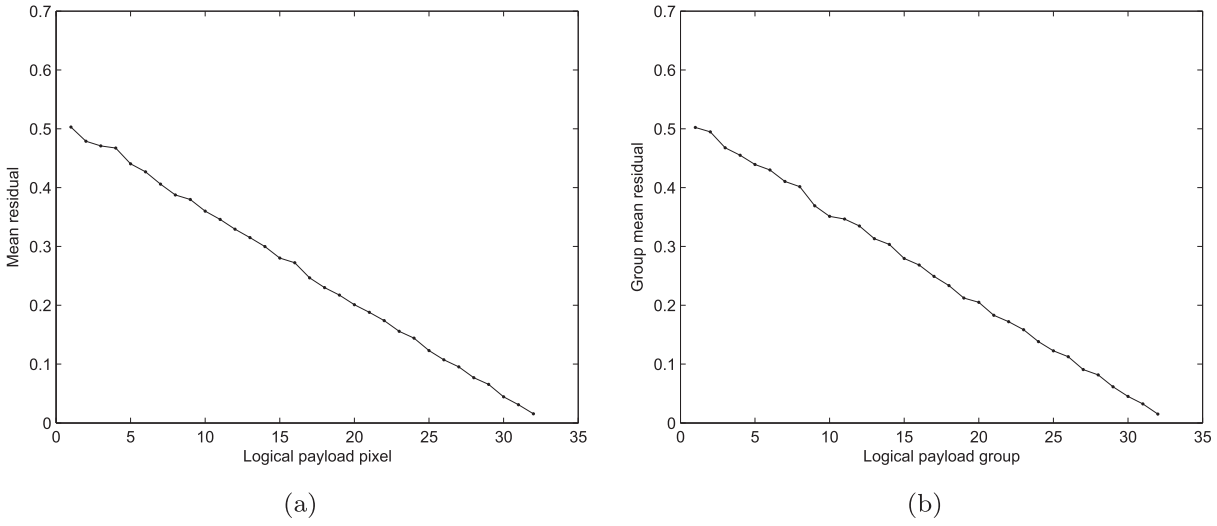

Fig. 2. Plot of the mean residuals of the payload pixels (groups for group-parity) in their logical order for (a) simple LSB steganography and (b) group-parity steganography. The linear decrease in mean residual as suggested by (5) and (10) is clear.

The accuracy of the obtained order depends on the number of image pairs. In general, the accuracy improves with more images. We report the average minimum edit distance between the ground-truth order and the obtained order as a function of the number of cover-stego image pairs for both algorithms in Table 1. With 1000 image pairs, the obtained order for most bits are already correct. The results confirm that the residuals contain enough information to obtain the correct logical order as shown in our analysis. The results for both algorithms are very similar. This is expected as (5) and (10) are identical. When the edit distance is non-zero, the obtained order can still reveal partial information about the hidden messages, e.g., consecutive sequences of bits, which may be invaluable to the forensic analyst.

Unknown cover images

The following experiment assumes that the cover images are unavailable and use the MRF cover estimator to estimate them to compute residuals. We use the default parameter setting: $w_1 = 1$, $w_2 = 1$, and $\rho = 0.25$. In addition, we must also provide the joint cover pixel probabilities for the pairwise cost functions. This is accomplished by learning from the set of cover images. However, to utilize all images in our experiment, we leave out the cover image that corresponds to the current stego image that is being estimated. This makes the experiment fair as no knowledge about the actual cover image is known. In practice, the analyst may use the same digital camera or capturing device the steganographer uses to generate good cover images.

Table 1
Average minimum edit distance between the ground-truth order and the obtained order as a function of the number of cover-stego image pairs for simple LSB steganography and group-parity steganography.

Images	Simple LSB	Group-parity
1000	8.0	9.5
2000	5.6	4.2
3000	3.3	2.8
4000	2.0	1.8
5000	1.6	1.2
6000	1.0	0.8
7000	1.0	0.0
8000	0.6	0.0
9000	0.0	0.0

Unlike the previous experiment, using cover estimators is not only noisy, but the results depend on the embedding operation as well. It is well-known that LSB replacement is easier to estimate than LSB matching due to its asymmetry. To this end, we perform the experiment using both algorithms: simple LSB replacement and simple LSB matching.

We plot the mean residuals of the payload pixels in their logical order using all estimate-stego image pairs for simple LSB replacement in Fig. 3. The residuals still exhibit a linear decrease. They are, however, noisier than the case when the cover images are available. This is expected as the estimator cannot produce perfect estimates. As a consequence, it is difficult to obtain the correct order using these residuals. We report the average minimum edit distance between the ground-truth order and the obtained order as a function of the number of estimate-stego image pairs for both embedding algorithms in Table 2. The accuracy against LSB matching is lower (larger edit distance) than LSB replacement, reflecting the fact that stego images modified by LSB matching are harder to estimate. The obtained order improves with more images and can still reveal partial information about the hidden messages.

Discussion and conclusion

The eventual goal of steganalytic forensic is not just to detect whether an image contains steganographic content, but also to extract the hidden message. Payload location offers an interesting approach that brings steganalytic

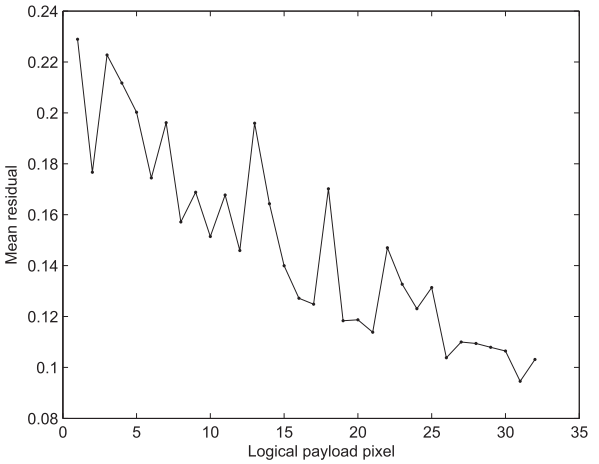

Fig. 3. Plot of the mean residuals of the payload pixels in their logical order for simple LSB replacement steganography. The residuals are computed from cover estimates. The residuals are noisy, but still exhibit a linear decrease.

research closer to realizing this goal. The main assumption in payload location is that the size of the payload is fixed. This assumption may be unnecessary and unrealistic except in special applications that require a fixed payload. After all, we would expect the steganographer to hide messages of various lengths. In these cases, the approach presented here shows that the messages can be extracted by ordering the located payload in descending mean residual. We note again that the presented approach also applies to other transformed domains such as quantized JPEG coefficients.

While this work is of theoretical importance as it is the first to show that it is possible to extract the hidden messages without knowledge of the embedding key, it also points out the surmounting difficulty facing the forensic analyst. The analyst must have a sufficiently large set of stego images to be successful. In addition, it is beneficial to have access to the cover images. The latter is primarily due to the weakness of current cover estimators and may improve with future research. Even then, it is only realistic to expect partial message extraction, but that may be indispensable to the forensic analyst. From a security perspective, this work exposes a vulnerability in block-based embedding algorithms. A sophisticated steganographer, however, can easily evade this method by using a more advanced embedding algorithm that adapts to the message size as well.

On a practical note, digital computers operate on bytes instead of bits. The payloads are therefore in bytes. The work here still applies, but at the byte boundary, e.g., we can order groups of eight bits. In practice, the stego images may also have been generated using several keys instead of just one. In these cases, we must be able to separate these stego images by embedding key and use the presented technique on each set separately. We defer investigating this problem to our future work.

Table 2
Average minimum edit distance between the ground-truth order and the obtained order as a function of the number of estimate-stego image pairs for simpless LSB replacement and matching steganography.

Images	Replacement	Matching
1000	24.7	27.4
2000	24.7	27.3
3000	23.8	26.4
4000	23.3	26.3
5000	23.3	25.7
6000	23.0	25.7
7000	22.3	25.2
8000	21.9	25.2
9000	21.8	25.0

References

Filler T, Ker AD, Fridrich J. The square root law of steganographic capacity for Markov covers. In: Media forensics and securityVol. 7254. SPIE; 2009. p. 725408.

Filler T, Pevný T, Bas P. Break our steganography system http://boss.gipsa-lab.grenoble-inp.fr; July 2010.

Fridrich J, Goljan M. On estimation of secret message length in LSB steganography in spatial domain. In: Security, steganography, and watermarking of multimedia contents VI, Vol. 5306. SPIE; 2004. pp. 23–34.

Fridrich J, Soukal D. Matrix embedding for large payloads. IEEE Trans Inf Forensics Security 2006;1(3):390–4.

Fridrich J, Goljan M, Soukal D. Searching for the stego-key. In: Security, Steganography, and Watermarking of Multimedia Contents VI, Vol. 5306. SPIE; 2004. pp. 70–82.

Garfinkel SL. Carving contiguous and fragmented files with fast object validation. In: Digital forensics research workshopVol. 4S. DFRWS; 2007. pp. 2–12.

Ker AD, Böhme R. Revisiting weighted stego-image steganalysis. In: Security, forensics, steganography, and watermarking of multimedia contents X, Vol. 6819. SPIE; 2008. p. 681905.

Ker AD, Lubenko I. Feature reduction and payload location with WAM steganalysis. In: Media forensics and security, Vol. 7254. SPIE; 2009. 72540A.

Ker AD. A capacity result for batch steganography. IEEE Signal Process Lett 2007;14(8):525–8.

Ker AD. Locating steganographic payload via WS residuals. In: 10th Multimedia and security workshop. ACM; 2008. pp. 27–31.

Ker AD. The square root law requires a linear key. In: 11th Multimedia and security workshop. ACM; 2009. pp. 85–92.

Kodovský J, Fridrich J. Quantitative steganalysis using rich models. In: Media watermarking, security, and forensicsVol. 8665. SPIE; 2013. 86650O.

Pevný T, Fridrich J, Ker AD. From blind to quantitative steganalysis. In: Media forensics and securityVol. 7254. SPIE; 2009. 72540C.

Quach T-T. On locating steganographic payload using residuals. In: Media Watermarking, security, and forensics IIIVol. 7880. SPIE; 2011a. 78800J.

Quach T-T. Optimal cover estimation methods and steganographic payload location. IEEE Trans Inf Forensics Security 2011b;6(4):1214–22.

Quach T-T. Locating payload embedded by group-parity steganography. Digit Investig 2012;9(2):160–6.

Quach T-T. Cover estimation and payload location using Markov random fields. In: Media Watermarking, security, and forensicsVol. 9028. SPIE; 2014. 90280H.