Non-shared selection channel. Steganalysis

Hristo Kostadinov

Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences

Non-shared selection channel

The steganographic security can be measured with the KL - divergence between the distributions of cover and stego images. One principle for minimizing the divergence is the principle of minimal embedding impact, which starts with the assumption that each cover element, i, can be assigned a numerical value, $\rho[i]$, that expresses the contribution to the overall statistical detectability if that cover element was to be changed during embedding. If the values $\rho[i]$ are approximately the same across all cover elements, minimizing the embedding impact is equivalent to minimizing the number of embedding changes.

Non-shared selection channel

Let Alice has a raw, never-compressed image and wants to embed information in its JPEG compressed form. When compressing the image, Alice can inspect the DCT coefficients after they have been divided by quantization steps but before they are rounded to integers and select for embedding those coefficients whose fractional part is close to 0.5.

For example, when rounding the coefficient -3.54, we can embed a bit by rounding it to -3 or to -4. The rounding distortion (rounding to -4) is 0.46. If embedding requires rounding to -3 instead, the combined rounding and embedding distortion is only slightly larger, 0.54.

Non-shared selection channel

Alice calculates for each pixel, i, in the cover image the variance, $\sigma^2[i]$, from all pixels in its local 3×3 neighborhood. Then, she sorts the variances $\sigma^2[i]$ from the largest to the smallest and embeds the payload of m bits using LSB embedding into the m pixels with the largest local variance. When Bob attempts to read the message, it may well happen that the m pixels with the largest local variance in the stego image will not be completely the same (or their order may not be the same) as those selected by Alice. Again, Bob is unable to read the message.

Writing on wet paper

Writing on wet paper. Imagine that the cover image $\mathbf{x}$ was exposed to rain and some of it pixels got wet. Alice is allowed only to slightly modify the dry pixels (the selection channel) but not the wet pixels. During transmission, the stego image $\mathbf{y}$ dries out and thus Bob has no information about which pixels were dry.

This problem is recognized in information theory as writing in memory with defective cells. A computer memory contains n cells out of which $n - k$ cells are permanently stuck at either 0 or 1. The device that writes data into the memory knows the locations and status of the stuck cells. The task is to write as many bits as possible into the memory so that the reading device, that does not have any information about the stuck cells, can correctly read the data.

Writing on wet paper

Let us assume that n cover elements are represented using a bit-assignment function, π, as a vector of n bits $\mathrm{x} \in \{0,1\}^n$. For example, one can think of $\mathbf{x}$ as the vector of LSBs of pixels. The sender forms a selection channel of k changeable elements $\mathrm{x}[j], j \in \mathcal{S} \subset \{1,\ldots,n\}, |\mathcal{S}| = k$, that can be modified during embedding. The remaining $n-k$ elements $\mathrm{x}[j], j \notin \mathcal{S}$, are not to be changed. Using the writing-on-wet-paper metaphor, S contains indices of *dry* elements (functioning memory cells) while the rest of the elements are *wet* (stuck cells). The sender's goal is to communicate $m < k$ message bits $\mathrm{m} \in \{0,1\}^n$ to the recipient who has no information about the selection channel S.

Writing on wet paper

The sender modifies some changeable elements in the cover image so that the bits assigned to the stego image, $\mathbf{y}$, satisfy

$$\mathbf{Dy} = \mathbf{m},$$

where $\mathbf{D}$ is an $m \times n$ binary matrix shared by the sender and the recipient. The recipient reads the message by multiplying the vector of stego image bits $\mathbf{y}$ by the matrix $\mathbf{D}$. While this appears identical to matrix embedding, there is one important difference because the sender is now allowed to modify only the changeable elements of $\mathbf{x}$.

Public-key steganography

In public-key cryptography, there exist two keys - an encryption key E and a decryption key D. The encryption key is made public, while the decryption key is kept private. Public-key encryption schemes have the important property that, knowing the encryption key, it is computationally hard to derive the decryption key. Thus, giving everyone the ability to encrypt messages does not automatically give the ability to decrypt. When Alice wants to send an encrypted message, **m**, to Bob, she can send him $E_B(D_A(\mathrm{m}))$, where E_B and D_A stand for the public (encryption) and private (decryption) key of Bob and Alice, respectively. Bob reads the message by $\mathrm{m} = E_A(D_B(E_B(D_A(\mathrm{m}))))$.

Public-key steganography

Imagine that Alice uses a steganographic scheme with a public selection channel but encrypts her payload using a public-key encryption scheme. Upon receiving an image from Alice, Bob suspects steganography, extracts the payload, and decrypts it to see whether there is a secret message from Alice.

The encrypted message is publicly available to Eve (the warden) but, as long as the cryptosystem is strong, Eve will not be able to tell whether the extracted bit stream is a random sequence or ciphertext. The fact that the selection channel is public, may give Eve a starting point to mount a steganalytic attack. This problem can be eliminated by using selection channels that are *completely random* implemented using wet paper codes with a public randomly generated matrix $\mathbf{D}$.

$e + 1$ matrix embedding

Let us assume that we have a matrix embedding scheme with $\pi(x) = \mathrm{LSB}(x)$ and binary code C with embedding efficiency e. This means that the sender can embed m bits in n pixels on average by making m/e embedding changes. Let us denote the set of modified pixels as S, $E[|\mathcal{S}|] = m/e$. If Alice modifies pixels in S by ± 1. She has a choice to adjust the second LSB of every pixel in S to either 0 or 1. Since Bob will not know the set S, Alice has to use wet paper codes, this time with the second LSB as the bit-assignment function and S as the set of changeable pixels.

$e + 1$ matrix embedding

The recipient first extracts m message bits from LSBs of the image using the parity-check matrix of the code C and then extracts an additional m/e bits from the second LSB using wet paper codes. Because a total of $m + m/e$ bits is embedded using m/e changes, the embedding efficiency of this scheme is

$$\frac{m + m/e}{m/e} = e + 1.$$

To summarize, if we allow the sender to modify the pixels by ± 1 rather than flip their LSBs, the embedding efficiency of the original binary matrix embedding scheme can be increased by 1. On the other hand, ± 1 embedding changes allow application of ternary codes that enjoy a higher bound on embedding efficiency.

Steganalysis

In the prisoners' problem, Alice and Bob are allowed to communicate but all messages they exchange are closely monitored by warden Eve looking for traces of secret data that may be hidden in the objects that Alice and Bob exchange. Eve's activity is called **steganalysis** and it is a complementary task to steganography.

The steganalyst is successful in attacking the steganographic channel if she can distinguish between cover and stego objects with probability better than random guessing. Note that. The important task of extracting the secret message from an image once it is known to contain secretly embedded data belongs to **forensic steganalysis**.

Typical scenarios

The steganographic channel consists of five basic elements: channel used to exchange data between Alice and Bob, cover source, message source, data embedding and data extraction algorithms, source of stego keys. We assume that the warden is passive. The physical channel used to exchange information is lossless and has no impact on steganalysis or steganography.

Although there certainly exist many different situations with various levels of detail available to Eve about the individual elements of the channel, we highlight two typical and very different scenarios:

- traffic monitoring;
- analysis of a seized computer.

Steganalysis as detection problem

It is typically achieved through some simplified model of the cover source obtained by representing images using a set of numerical features. Depending on the scope of the features, we recognize two major types of statistical steganalysis – targeted and blind.

The cover source can be described by a probability distribution, P_c, on the space of all possible cover images, C. The value $\mathrm{P}_c(\mathcal{B})$ is the probability of selecting cover $x \in \mathcal{B} \subset \mathcal{C}$ for hiding a message. A given stegosystem assumes on its input covers $x \in \mathcal{C}, x \sim \mathrm{P}_c$, stego keys, and messages (both attaining values on their sets according to some distributions), the distribution of stego images is P_s.

Steganalysis as detection problem

Let us assume that Eve can collect sufficiently many cover images and estimate P_c. If her knowledge of the steganographic channel allows her to estimate the P_s, we speak of steganalysis of a known stegosystem, which is a detection problem that leads to simple hypothesis testing

$$H_0 : x \sim P_c$$

$$H_1 : x \sim P_s$$

If Eve has no information about the stego system, the steganalysis problem becomes

$$H_0 : x \sim P_c$$

$$H_1 : x \nsim P_c$$

which is a composite hypothesis-testing problem and is in general much more complex.

Steganalysis as detection problem

Obviously, there are many other possibilities that fall in between these two formulations depending on the information about the steganographic channel available to Eve. For example, Eve may know the embedding algorithm but not the message source. Then, P_s, will depend on an unknown parameter, the change rate β, and Eve faces the composite one-sided hypothesis-testing problem

$$H_0 : \beta = 0,$$

$$H_1 : \beta > 0.$$

Eve may employ appropriate tools, such as the generalized likelihood-ratio test, or convert this problem to simple hypothesis testing by considering β as a random variable and making an assumption about its distribution.

Modeling images using features

The models used in steganalysis are usually obtained by representing images using a set of numerical features. Each image, $\mathrm{x} \in \mathcal{C}$ is mapped to a $d-$dimensional feature vector $\mathrm{f} = (f_1(\mathrm{x}), \ldots, f_d(\mathrm{x})) \in \mathbb{R}^d$, where each $f_i : \mathcal{C} \to \mathbb{R}$. The random variables representing the cover source, $\mathrm{x} \sim P_c$, and the stego images, $\mathrm{y} \in P_s$, are thus transformed into the corresponding random variables $\mathrm{f}(\mathrm{x}) \sim p_c$ and $\mathrm{f}(\mathrm{y}) \sim p_s$ on $\mathbb{R}^d$. For accurate detection, the features need to be chosen so that the clusters of $\mathrm{f}(\mathrm{x})$ and $\mathrm{f}(\mathrm{y})$ have as little overlap as possible.

Modeling images using features

The models used in steganalysis are usually obtained by representing images using a set of numerical features. Each image, $\mathrm{x} \in \mathcal{C}$ is mapped to a $d-$dimensional feature vector $\mathrm{f} = (f_1(\mathrm{x}), \ldots, f_d(\mathrm{x})) \in \mathbb{R}^d$, where each $f_i : \mathcal{C} \to \mathbb{R}$. The random variables representing the cover source, $\mathrm{x} \sim P_c$, and the stego images, $\mathrm{y} \in P_s$, are thus transformed into the corresponding random variables $\mathrm{f}(\mathrm{x}) \sim p_c$ and $\mathrm{f}(\mathrm{y}) \sim p_s$ on $\mathbb{R}^d$. For accurate detection, the features need to be chosen so that the clusters of $\mathrm{f}(\mathrm{x})$ and $\mathrm{f}(\mathrm{y})$ have as little overlap as possible.

Optimal detectors

Any steganalysis algorithm is a detector, which can be described by a map $F : \mathbb{R}^d \to \{0, 1\}$, where $F(\mathrm{x}) = 0$ means that x is detected as cover, while $F(\mathrm{x}) = 1$ means that x is detected as stego. The set $\mathcal{R}_1 = \{x \in \mathcal{R}^d | F(\mathrm{x}) = 1\}$ is called the critical region because the detector decides "stego" if and only if $\mathrm{x} \in \mathcal{R}_1$. The critical region fully describes the detector.

The detector may make two types of error - false alarms and missed detections. The probability of a false alarm, P_{FA}, is the probability that a random variable distributed according to p_c is detected as stego, while the probability of missed detection, P_{MD}, is the probability that a random variable distributed according to p_s is incorrectly detected as cover.

Targeted steganalysis. Features

The features in targeted steganalysis are constructed from knowledge of the embedding algorithm. The features in blind steganalysis must be constructed in such a manner as to be able to detect every possible steganographic scheme, including future schemes. Both targeted and blind steganalysis can use multiple features and tools from machine-learning and pattern recognition. The main difference between them is the scope of their feature sets.

In targeted steganalysis the embedding mechanism of the stego system is known. It makes sense to choose as features the quantities that predictably change with embedding. While it is usually relatively easy to identify many such features, features that are especially useful are those that attain known values on either stego or cover images.

Targeted steganalysis. Features

Let f_β is the feature computed from a stego image obtained by changing the ratio of β of its corresponding cover elements.

- **Testing for stego artifacts.** Identify a feature that attains a specific known value, f_β, on stego images and attains other, different values on cover images. Then,

$$H_0 : f = f_\beta,$$

$$H_1 : f \neq f_\beta.$$

- **Calibration.** It is possible to estimate from the stego image what the value of a feature would be if it were computed from the cover image. This process is called calibration.

$$H_0 : \hat{\beta} = 0,$$

$$H_1 : \hat{\beta} > 0.$$

Blind steganalysis

The goal of blind steganalysis is to detect any steganographic method irrespective of its embedding mechanism. As in targeted steganalysis, we cannot work with the full representation of images and instead transform them to a lower dimensional feature space. Ideally, we would want the feature space to be complete, i.e., we do not lose on our ability to distinguish between cover and stego images by representing the images with their features.

Blind steganalysis. Features

The impact of embedding can be considered as adding noise of certain specific properties. Thus, many features are designed to be sensitive to adding noise while at the same time being insensitive to the image content.

- **Noise moments.** Transform the image to some domain, such as the Fourier or wavelet domain, where it is easier to separate image content and noise. Compute some statistical characteristics of the noise component (such as statistical moments of the sample distributions of transform coefficients). By working with the noise residual instead of the image, we essentially improve the signal-to-noise ratio (here, signal is the stego noise and noise is the cover image itself) and thus improve the features' sensitivity to embedding changes, while decreasing their undesirable dependence on image content.

Blind steganalysis. Features

- **Calibrated features.** Identify a feature that is likely to predictably change with embedding and calibrate it. For these features to work best, it is advisable to construct them in the same domain as where the embedding occurs. For example, when designing features for detection of steganographic schemes that embed data in quantized DCT coefficients of JPEG images, compute the features directly from the quantized DCT coefficients.
- **Targeted features.** Many features for blind steganalysis originated in targeted steganalysis. In fact, it is quite reasonable to include in the feature set the features that can reliably detect specific steganographic schemes because, this way, the blind steganalyzer will likely detect these steganographic schemes well.

Blind steganalysis. Classification

After selecting the feature set, Eve has at least two options to construct her detector. One possibility - mathematically describe the probability distribution of cover-image features, for example, by fitting a parametric model, $\hat{p}_c$, to the sample distribution of cover features and test

$$H_0 : x \sim \hat{p}_c,$$

$$H_1 : x \nsim \hat{p}_c,$$

The second option is to use a large database of images and embed them using every known steganographic method with uniformly distributed change rates β and then fit another distribution, $\hat{p}_s$, through the experimentally obtained data. Eve now faces a simple hypothesis-testing problem,

$$H_0 : x \sim \hat{p}_c,$$

$$H_1 : x \sim \hat{p}_s,$$

Blind steganalysis. Classification

In practice the detection is formulated as classification. Eve trains a classifier on features $f(x)$ for $\mathbf{x}$ drawn from a sufficiently large database of cover and stego images to recognize both classes.

Eve can construct her classifier in two different manners. The first option is to train a cover-versus-all-stego binary classifier on two classes: cover images and stego images produced by a sufficiently large number of stego algorithms and an appropriate distribution of message payloads. With this approach, there is always the possibility that in the future some new steganographic algorithm may produce stego images whose features will be incompatible with the distribution $\hat{p}_s$, in which case such images may be misclassified as cover.

Influence of cover source on steganalysis

One should avoid using covers with little redundancy, such as images with a low number of colors represented in palette image formats, because there the spatial distribution of colors is more predictable. Among the most important attributes of the cover source that influence steganalysis accuracy, we name the color depth, image content, image size, and previous processing.

Images with a higher level of noise or complex texture have a more spreadout distribution than images that were compressed using lossy compression or denoised. Scans of films or analog photographs are especially difficult for steganalysis because high-resolution scans of photographs resolve the individual grains in the photographic material, and this graininess manifests itself as high frequency noise.

Influence of cover source on steganalysis

The image size has an important influence on steganalysis as well. Intuitively, it should be more difficult to detect a fixed relative payload in smaller images than in larger images because features computed from a shorter statistical sample are inherently more noisy. The effect of image size on reliability of steganalysis also means that JPEG covers with a low quality factor are harder to steganalyze reliably because the size of the cover is determined by the number of non-zero coefficients, which decreases with decreasing quality factor.

Influence of cover source on steganalysis

Image processing may play a decisive role in steganalysis. Processing that is of low-pass character (denoising, blurring, and even lossy JPEG compression to some degree) generally suppresses the noise naturally present in the image, which makes the stego noise more detectable. This is especially true for spatial-domain steganalysis. In fact, it is possible that a certain steganalysis technique can have very good performance on one image database and, at the same time, be almost useless on a different source of images. Thus, it is absolutely vital to test new steganalysis techniques on as large and as diverse a source of covers as possible.

Influence of cover source on steganalysis

JPEG images are less sensitive to processing that occurred prior to compression. Because the statistical distribution of DCT coefficients in a JPEG file is significantly influenced by the quality factor, for some steganographic schemes separate steganalyzers may need to be built for each quality factor to improve their accuracy. A complication for steganalysis of JPEG images is repeated JPEG compression with different quality factors because it leads to a phenomenon called "double compression" that may drastically change the statistical distribution of DCT coefficients. Double-compression artifacts may be mistaken by some steganalysis methods as an impact of steganography.

Influence of cover source on steganalysis

The process of calibration is especially vulnerable to double compression because, during calibration, the effects of both compressions are suppressed and the steganalyst does not estimate the doubly compressed cover but instead the cover singly compressed by the second quality factor. One possible approach here is to estimate the primary quantization matrix, and then calibrate by mimicking both compression processes.

There are some situations when a certain combination of covers and steganographic system is so unfortunate that it becomes possible to detect even a single modification.

Forensic steganalysis

Forensic steganalysis: A collection of tasks needed to identify individuals who are communicating in secrecy, the stegosystem they are using, its parameters (the stego key), and the message itself.

1. Identification of web sites, Internet nodes, or computers that should be analyzed for steganography.
2. Development of algorithms that can distinguish stego images from cover images.
3. Identification of the embedding mechanism, e.g., LSB embedding, ± 1 embedding, embedding in the frequency domain, embedding in the image palette, sequential, random, or content-adaptive embedding.
4. Determining the steganographic software.
5. Searching for the stego key and extracting the embedded data.
6. Deciphering the extracted data and obtaining the secret message.

Forensic steganalysis

The power of steganography is that it not only provides privacy in the sense that no one can read the exchanged messages, but also hides the very presence of secret communication. Thus, the primary problem in steganography detection is to decide what communication to monitor in the first place. Since steganalysis algorithms may be expensive and slow to run, focusing on the right channel is of paramount importance. Second, the communication through the monitored channel will be inspected for the presence of secret messages using steganalysis methods. Once an image has been detected as containing a secret message, it is further analyzed to determine the steganographic method.

Forensic steganalysis

The character of the embedding changes also leaks information about the embedding mechanism. If Eve can determine that the LSBs of pixels were modified, she can then focus on methods that embed into LSBs. Eventually, Eve may guess which stego method has been used and attempt to determine the stego key and extract the embedded message. If the message is encrypted, Eve then needs to perform cryptanalysis on the extracted bit stream.