Some Notes on Steganography

Ron Crandall

Friday, December 18, 1998
xrc@ieee.org

Abstract. This essay presents some observations concerning low rate steganography in images where the amount of covert information is below one bit per pixel. This discussion is restricted to paletted images, such as color GIF pictures. Much of this essay is relevant to other cover messages, but the application is left to the reader. It includes information and techniques for reducing the capability of an attacker to detect steganographic alterations in a cover image. This discussion assumes the covert message is compressed and/or encrypted to have its statistics appear random or nearly random.

Covert Message Extraction

This essay proposes a general approach where the extraction process is viewed as multiple applications of a function. Call the input to the function a "cell". Cells can be a single-pixel, multiple pixels, or fractionating, such as eleven bit blocks ignoring pixel boundaries. Cells may also be of varying size or non-contiguous bits. This essay uses the term "extractor" to describe the function which derives covert data from the cells in the image. A simple example of an extractor function is one that returns the parity of the bits in the cell. Extractor functions over multiple pixels that create a single bit of output are the basis of pixel selection and are discussed below.

Since cells may be larger than a single pixel, it is reasonable for the output of an extractor function to be more than a single bit, such as a ternary digit or two bits, etc. An example of a multiple bit output function is one that takes five pixels as input and returns the overall parity as one bit and the XOR of five lookup values as another bit. The encoding process can select a strategy that produces the least conspicuous change to the image. For example, it can change a single pixel to force both bits or change two or more pixels instead. Another possibility is to use a secure hash function, such as SHA, on five pixels and select two bits of the result. Encoding in this case consists of trial and error starting with the least conspicuous change.

Improvements to Traditional Steganography

The traditional extractor has a single-pixel input and a single bit output which is a copy of the LSB. There are two major problems with this practice:

- An attacker can concentrate on the statistics of the LSB.
- In paletted images, forcing the LSB to the desired value may result in substantial color shifts, depending on the ordering of the palette.

In non-paletted images, changing the LSB by incrementing or decrementing the original value (with caution for overflow and underflow), or by forcing the LSB ensures one of the two closest possible values. See the paper "Practical invisibility in digital communication", by Tuomas Aura, for a discussion of the LSB statistics of grey scale images.

In a paletted image the LSB does not necessarily have the least visual impact. The first recommendation of this essay is to rely on parity, or some other function of the entire cell, rather than the LSB. The use of such a function eliminates the ability of an attacker to concentrate on the statistics of the LSB and forces a much more difficult analysis of the overall characteristics of the cell values. In general, an extractor can be any function that partitions the set of possible input values into disjoint sets. These sets should be nearly equal in size. Another example of such an extractor returns a bit derived by lookup from a randomized table. In any event, the corresponding encoder must pick the best possible palette value from the subset of palette values having the correct extractor output.

Encoding During Color and Spatial Reduction

Traditional steganography has generally used simple techniques for encoding a single-bit into a pixel. The low rate techniques being discussed in this paper allow the encoder more choices, such as a choice of which pixels in a cell to change. When possible, the encoding process should be done during the color reduction of a 24 bit image and/or any size reduction. This requires a more involved analysis and coding process, but it allows the encoder more information on which to base decisions.

Conceptually, the encoder examines an area of the image and weighs each of the options that allow it to embed the desired bits in that area. It scores each option for how conspicuous it is and chooses the option with the best score. Weighting factors in the score might include:

- The color difference between the encoded pixel and its true 24-bit color.
- The gradient near the pixel. Reasonable values for pixels in high gradient areas have a wider margin because a microscopically different position of the image on the detector would lead to a different pixel value anyway. However, caution must be exercised that a sharp edged regular feature is not distorted.
- Try to reduce the net color shift in an area. This is a form of error diffusion.
- Consider known physiological effects to modify the best encoding. For instance, people seem to be relatively insensitive to changes between red and magenta (i.e. the human visual system has a lower blue sensitivity).

Encoder algorithms are an open research area.

Lower Rate Techniques—General

The most obvious way to reduce the possibility of detection is by reducing the density of changes in the cover message and consequently the information density. The current method of choice uses some method of scattering the message bits throughout the image. Previous techniques often left the message bits contiguous at the beginning of the cover message. However, the alternatives described here are more effective.

For the purposes of this essay, I refer to a data "rate" where 100 % is one bit of message for each pixel of cover message. Purists might object that this is actually 12.5 %. However, making the reference relative to the commonly implemented 1-bit-per-pixel is arguably more useful in discussing alternatives. I also refer to the proportion of altered pixels as the "change density". Even in the conventional technique with a rate of 100 % the change density is normally 50 % (because in the random case 50 % of the pixels already convey the correct information). This conventional technique may also be viewed as coding two bits of message per changed pixel.

Lower Rate Techniques—Pixel Selection

The first proposal in this category is for the extractor function input to be a "cell" of more than a single-pixel. An example of an effective extractor function is letting the message bit be the parity of the entire cell. As noted earlier, it is possible to use non-adjacent pixels and/or a non-binary output from the extractor. If the encoding for a group is wrong, choose any member of the group, or combination of members when appropriate, to change. Choose the variation which has the least impact on the image. The key to a good result is the ability to identify the pixels having the least visual and statistical effect on the cover image.

Consider a design that uses 3 pixels in a cell to give a rate of 33 % and a change density of 17 %. This design provides the same rate and change density as if every third pixel had been used for the message. However, now it is possible to change those pixels that are least noticeable. The effects of the reduced change density and the ability to select a "best choice" for change have a cumulative effect on the difficulty of detecting the presence of a hidden message.

Spreading Functions

Cells may be made up of pixels (or bits) that are not necessarily spatially close. Scattered data makes it easier to skip pixels that might be visually more prominent in favor of changing a pixel in a less conspicuous area. Scattering functions should not generate duplicate locations because it complicates the encoder implementation. An example of a simple function is to use an integer that is relatively prime to the size of the image and multiply it by the cell ordinal (mod the image size). Also, as long as processing is serial, a Linear Feedback Shift Register

(LFSR) algorithm can be used to cycle through all possible cells. Choose a LFSR with a cycle just larger than the image size and ignore outputs beyond the size of the image to eliminate multiple use of one cell. A normal LFSR does not generate a zero, so compensate accordingly. Cryptographicly strong scattering functions are not discussed here.

Lower Rate Techniques—Matrix Encoding

The next proposal is best introduced by way of an example. There is a high level function to produce the actual covert message output and a low level extractor of the kind already discussed. This example partitions the cover message into blocks of 3 cells and encodes two message bits into each block. Start the encoding by deriving extractor values for each of the three cells in an unmodified block and call these values a, b, and c respectively. Now consider the two message bits. Message bit 1 is (a xor c), and message bit 2 is (b xor c). If the three cells are already correct (probability one quarter) do nothing. If (a xor c) is correct for message bit 1, but (b xor c) is not correct for message bit 2, change cell b. If (a xor c) is not correct for message bit 1, but (b xor c) is correct for message bit 2, change cell a. If both message bits are wrong, change cell c. Assuming one pixel cells, this method has a rate of 67 %, and a change density of 25 %.

This paper refers to these designs by an ordered triple containing the number of cell changes allowed in a block, the number of cells in a block, and the number of bits conveyed. The above description is a $(1, 3, 2)$ design. Designs can be derived for a variety of rates and change density. To avoid unnecessary complication, rates and change densities in the following descriptions are based on one pixel cells. Simple coding designs as described in the preceding paragraph (where only one cell is changed in each block) exist for 7 cells carrying 3 bits (rate=42 % change density=12.5 %, $(1, 7, 3)$), 15 cells carrying 4 bits (rate=27 %, change density=6 %, $(1, 15, 4)$), 31 cells carrying 5 bits (rate=16 %, change density=3 %, $(1, 31, 5)$), etc. The $(1, 31, 5)$ design is conveying 5 message bits for every cell changed! The penalty, of course, is the low rate.

This family of designs uses the same decoding matrix H as the Hamming error correcting codes (R. W. Hamming. Error detecting and error correcting codes. Bell Sys. Tech. Journal, 29, 1950. A modern tutorial is in Coding and Information Theory, Richard W. Hamming). In general, the purpose of error control codes is to ensure code words have maximal distance from each other. For this application a special class of codes (covering codes) is required to ensure every word can be generated from a minimum number of code words. The Hamming codes (and the Golay codes mentioned later) are the only examples of what are known as perfect error correcting codes and are also good covering codes.

Slightly better rates are possible in designs where more than one cell in a block can be changed. A practical design exists that changes no more than 2 cells in a block of 13 to convey 6 information bits $(2, 13, 6)$. This design has a rate of 46 % and a change density of 13.5 %. Compare this to the similar $(1, 7, 3)$ where two blocks occupy 14 cells, have two cells changed, and convey 6 bits (rate

42 %, change density 12.5 %). Both designs convey about three bits for each cell changed.

The more complex design uses one less cell of cover. This means the rate and the change density have increased proportionately. Another example going to higher orders shows that changing no more than 3 cells in a block of 23 can convey 11 bits $(3, 23, 11)$. This design is based on the binary Golay error correcting code and has a rate of 48 % and a change density of 12.4 %.

Similar codes are possible in the non-binary case, but are not discussed here.

Attainable Rates

For a given number of cell changes (k) and message bits, it is readily seen that the number of cells in a block (n) is lower bounded by combinatorial considerations (n items taken k at a time plus n items taken $k-1$ at a time, etc.). I have posed the coding problem to Jürgen Bierbrauer of Michigan Technological University and he was able to derive tighter bounds on the attainable capacities, as well as providing other valuable information (private communications). For a change rate of 12.5 %, the limiting case (the best *possible* encoding) has a rate of about 54 %, for a change rate of 6.25 %, the best possible code has a rate of about 33 % and for a change rate of 0.8 %, the best possible code has a rate of about 6.6 %. The $(3, 23, 11)$ code has a change density of 12.4 % and a rate of 48 %, the $(1, 15, 4)$ code has a change density of 6.25 % and a rate of 26 %, and the $(1, 127, 7)$ code has a change density of about 0.8 % and a rate of 5.5 %. This demonstrates the improvement possible by increased coding complexity beyond these relatively simple schemes is small. To reduce the number of changed cells per bit the amount of cover text must be greatly increased. Still, it should be clear that this powerful technique is much better than just scattering a reduced number of conventional changes through an image.

Consider the capacity implications with some examples in a normal image. Assume a paletted image that is 640 × 480 or about 307K pixels. Assume that each cell is one pixel. A conventional design allows a 38 Kbyte hidden message with about 153K pixels changed. The $(3, 23, 11)$ design will convey an 18K message with 40K pixels changed. The $(1, 31, 5)$ design provides a 6 Kbyte secret message for a little under 10K pixels changed. Compared to the conventional design, this hidden message has 16 times less impact on the cover image, but is only 6 times smaller. If the information rate requirements are low, the $(1, 255, 8)$ design conveys 1200 bytes of information with 1200 pixels changed. Or, if the information rate requirements are very low, the $(1, 4095, 12)$ design conveys 112 bytes of information with only 75 pixels changed.

Additional Items

The pixel selection technique enables the encoder to select the pixel for alteration from a small group (the "cell"). This technique is interesting because of the combination of a reduced density of pixel changes and the ability to select the least conspicuous pixels for change. These two effects work synergistically to reduce the detectability. Matrix coding has the advantage of coding multiple message bits for each cell change. However, the location of each change is rigidly specified by the coding rules. In the case where a cell is a single-pixel, it may require a pixel in an adverse location to be changed. To accommodate this case, some other arbitrary change can be made followed by a reevaluation. Alternatively, you can gain the advantage of both techniques by using multiple pixel cells (at a considerably lower rate).

Another use for pixel selection is to hide multiple messages in the same image. They could be separately recovered using their unique keys without revealing the fact another message is present. Other people have suggested this method as a defense against "rubber-hose" cryptanalysis. A hidden but relatively "safe" cover message can be surrendered under duress. Study of the matrix method provided the key to this observation. If the pixel selections for multiple pixel extractor functions overlap in an appropriate way it is possible to code multiple messages simultaneously. Each message has its own key and there are no obvious clues that multiple messages are present.

A simple example would use two overlapping pixel selections. The first, an extractor function over 5 pixels using overall parity for the extractor output bit. The second, an extractor function over 3 pixels with table lookup for each output bit. It is always possible to select pixels and change values making both hidden messages attain their desired value. As another example, a completely unrelated steganographic technique using a fraction of the available pixels can be used to hide a message in an image. The image can then have another message hidden by pixel selection using knowledge from the first method about which pixels must not be changed.

Summary

This paper has demonstrated several methods for reducing the impact of a hidden message on the cover text. Additional topics for research are:

- the impact of non-binary coding. Example; an extractor over a block of 4 pixels where the output is the block mod 3
- the design of efficient encoders
- the best mechanism for applying these methods to images other than paletted images (e. g. JPEG) or other forms of cover text.