Steganography: messages hidden in pictures

Can a picture say a thousand words?

Gareth Dwyer
garethdwyer@gmail.com
October 16, 2014

Abstract

Codes and ciphers have been around for millennia. Julius Caesar famously used a simple alphabet substitution cipher to have important messages sent through potentially dangerous territory, which is known as the 'Caesar cipher'. The creation of the German 'Enigma', and the efforts of the cryptologists who eventually managed to break it, played an enormous part in the outcome of World War II. Cryptography remained important throughout the world, and hit a new peak with the advent of the Internet. Recently, cryptography and encryption have become buzzwords in media and in conversation, as ordinary people fervently discuss topics involving the leaks by whistle-blowers such as Edward Snowden and Julian Assange. Government organisations and three-letter agencies have been violating people's privacy in the name of halting terrorism – and not everyone is happy about it. With doubt arising over the security of what was previously considered 'unbreakable' encryption, those who want to communicate privately are becoming more inventive in their methods. This essay takes a look at steganography, or more specifically at the practice of hiding messages in images – a method with which it is surprisingly easy to securely share private messages. Although steganography predates the digital age, computers and the proliferation of digital imagery have made it far easier to achieve.

Introduction

Let's imagine three people: Alice, Bob, and Eve. Alice wants to send a private message to Bob, while Eve wishes to intercept this message, probably with malicious intent. If you know anything about computer security, these names will be familiar, for they are used to explain many communication concepts, from basic emails to quantum encryption. The problem for Alice and Bob is that although communication over the Internet is lightning fast, and easily

achieved, the Internet was built from an unstable trust-based model, where a message moves from node to node until it reaches its destination. Any of the intermediary nodes have full access to the message – unless it is encrypted.

But the reliability of even state-of-the-art 'unbreakable' encryption has been called into doubt. Whether it is because the NSA may have played a dark part in creating intentional weaknesses in commonly used encryption algorithms, or because computers have become powerful enough to break even untampered, sophisticated encryption methods, Alice and Bob need to go to more extreme ends to ensure a method of communication which is interpretable by each of them, but not by Eve. One of the most fundamental challenges of encryption is that in order to create a truly unbreakable cipher, both the sender and the receiver should ideally have shared a completely private 'secret' or 'key'. But this is not always possible, for if their messages are being intercepted, how do they share the key? One way of sharing secret messages without having already shared a key is by hiding them in images. This is known as steganography, and although it has been around in some forms since as early as 440 B.C. it is becoming easier and more popular in our digital world.

Images and text

Images are prolific. Proud parents send countless badly-shot cellphone snaps of their new-borns to relatives, who pretend to appreciate them; companies' logos are displayed prominently on their home-pages; and adverts showing pictures of scantily clad women are used to convince the unwary to divulge their credit card information to strangers. A single image is made up of millions of pixels; a standard smart-phone these days is likely to have a two mega-pixel camera, or better, which means that every picture it takes, uncompressed, contains about two million pixels. A colour pixel is most simply represented by a group of three numbers between 0 and 255, one number each for the red, blue, and green values of the pixel. Using just these three numbers, any colour on the Red-Green-Blue or RGB colour scale can be represented.[1]

Digital text, like images, is also represented by numbers. Using one of the most basic text encoding methods, ASCII, each character is represented by a number between 0 and 255. Thus a computer, using ASCII encoding, sees an 'A' as 65, a 'B' as 66, and a 'Z' as 90. A lowercase 'a' is 97, and other numbers are assigned for special characters.

This means that we can represent up to three characters in every single pixel of a picture: (84, 104, 101) could be both a dirty turquoise colour as well as the word 'The'. In a normal uncompressed photograph, we can therefore

[1]Note that although this simple format for images exists, known as PPM, it is quite uncommon. Uncompressed formats such as Bitmap are more common, but throughout this essay we use PPM as our example format, for reasons of simplicity.

(a) Alice in Wonderland

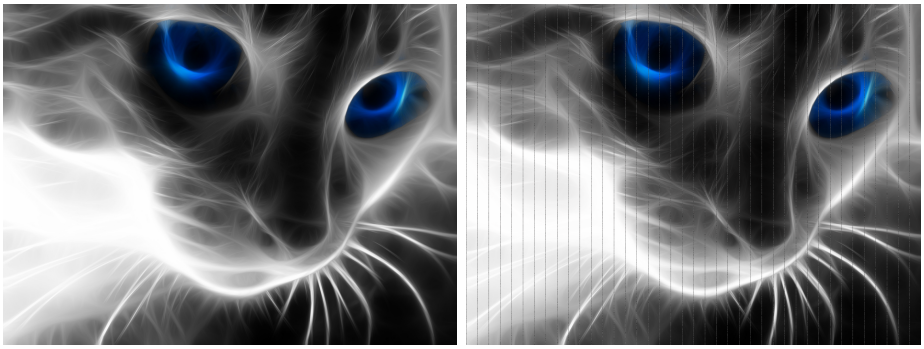
Figure 1: Alice in Wonderland as an image

fit about six million characters. The full text of *Alice in Wonderland* by Lewis Carroll is 26 000 words or about 150 000 characters, so we can see that there is the potential to send substantial amounts of text in a single image.

Keeping the secret

If Alice sends Bob a message, encoding it as described above and formatting it as an image, there is still the possibility that Eve will intercept the image, and manage to decipher it. This possibility is exacerbated by the fact that text encoded as an image as described above doesn't look like your average family snap, but more like white noise on a television. Figure 1 shows the whole of Alice in Wonderland hidden in a 300x300 pixel image with black pixels being used for the extra space, and we can agree that Eve would probably realise that there was more to the image than meets the eye. Assuming that Eve has some technical knowledge (a safe assumption, considering that we already assume that she has the knowledge needed to intercept digital communications), it probably won't take her long to work out how to decode the message. So how can Alice be more discreet?

One solution is to use an ordinary looking image, and then overwrite *some* of the pixels. As long as Bob knows which pixels represent the message, he will be able to decode it, and instead of random noise, we'll get something that resembles an actual image. In Figure 2, we can see an original image, and then the image again with *Alice in Wonderland* hidden within, using only every 40th number (each number being a third of a pixel). Although the lines are noticable, these are more likely to be overlooked as part of the image than the pure 'noise' seen in Figure 1. Because of the inefficiency of using only every Nth pixel, our image needs to be a bit larger, but in



(a) Original (b) With Alice in Wonderland

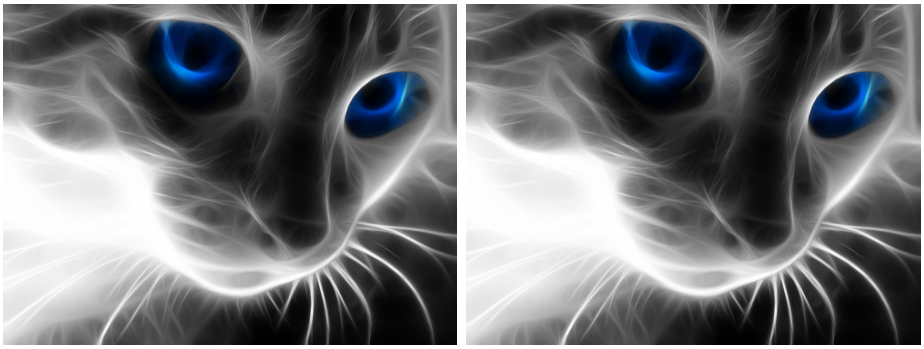
Figure 2: With and without Alice in Wonderland

the example the full text of *Alice in Wonderland* still easily fits in a two mega-pixel image.

Again, however, Eve is likely to notice that something is amiss, and spend some time and effort 'cracking the code'. Can we do better still?

It turns out we can. Instead of representing our text in decimal ASCII, where an 'A' is 65, we can represent it in binary. Binary, or base 2, is a base in which only the numbers 0 and 1 are used. In our standard base 10 or decimal system, the last digit of a number represents 1s, the second-last 10s, the third-last 100s, and so forth. The number 111 represents one 100, one 10, and one 1. Adding these together we get one-hundred and eleven. In base 2, the last number represents 1s, the second-last, 2s, the third last 4s, the fourth-last 8s, and so on, doubling with each added digit. Therefore, 111 in binary represents one 1, one 2, and one 4, and we would usually write it as 7. Let's see how we can use this to hide our messages even more securely.

One possibility is based on the fact that every number representing a third of a pixel has to have either an odd or an even value. We can interpret all odd pixel values as 1s, and all even values as 0s. Any given image can therefore be seen as a practically random string of 1s and 0s. To encode our text into the image, we simply convert our text to binary ASCII values, meaning our text is now a string of 1s and 0s. We then read each pixel of the image, and see if it 'matches' with the value we want (i.e. it is odd if we need a 1 or even if we need a 0). Half of the time, it will already be the value we need, and the other half, we simply modify the value by 1, making it odd or even as required. The final image is almost identical to the original image, as half of the pixel numbers are identical, and the other half have been modified by 1. Remembering that each value we read is actually only a third of a pixel (either the red, green, or blue measure), it shouldn't be surprising that this kind of modification is completely undetectable to the human eye, for adding a minuscule amount of one colour to a single pixel keeps it virtually identical.



(a) Original (b) With Alice in Wonderland

Figure 3: With and without Alice in Wonderland

Adding more layers

Using the methods explained above, Alice can send messages to Bob disguised as innocent pictures. But the system remains imperfect. Depending on their situation, Eve may still grow suspicious if she is expecting Alice to try to communicate with Bob, and sees a stream of images passing between them. And considering that Eve may be anyone from a casual eavesdropper, with nothing better to do, to an entire three-letter government agency, with practically unlimited resources and technology, Alice and Bob may want to add a final layer of security.

One way to achieve this is counter-intuitive. Instead of Alice sending the image through some supposedly private means of communication, such as e-mail, (which, although private, is not completely secure) Alice can leave the image in a public place for Bob to 'find'. Alice could upload the modified image to any number of public image sharing services, such as Instagram, or even more creatively, she could use a website such as 9gag.com where hundreds of 'funny' images are uploaded every day for the amusement of thousands of viewers. As long as Bob has some inkling of where to look for the image, he can then retrieve it, without alerting Eve to the fact that he and Alice are even communicating. Yet another option is to hide the message in an advertisement image, and pay for a third-party site to show it to their users.

With the number of images to be found around the world wide web, Eve's goal of finding the correct image, and still managing to decode the message it contains, becomes close to impossible.

Challenges

Of course, the more layers of security that Alice and Bob add to their communication, the higher the need becomes for prior contact in order to

correctly receive and interpret each other's messages. But this prior contact is still easier to achieve than for traditional encryption in which each needs to share a long private secret which is different for each message. And should Alice and Bob have had the opportunity to communicate privately in the past, and had the foresight to share the necessary keys, then traditional encryption can be combined with steganography for an even more layered approach to covert communication. That is, Alice can encrypt the message traditionally, and then hide the result of the encryption in the image, making extraction and code-breaking analysis more difficult for Eve.

Another challenge, especially if the method of storing the images in publicly accessible places is used, is that images are almost always compressed. Many compression schemes use so-called 'lossy' compression, in which less important pixels in an image are identified and either discarded or modified. A popular example of lossy compression is found in the ubiquitous JPEG format. Once an image has been compressed using one of these methods, it is impossible to reverse the process and retrieve the original image again, even though the result is indistinguishable to the human eye at normal zoom levels. And because there are many different compression methods, and public image sites do not usually specify which they are using, the subtleties introduced to an image by Alice or Bob may well be lost. Therefore Alice and Bob would need to either identify a suitable sharing platform where uncompressed images may be shared, or to examine the compression methods used and attempt to work around them.

Nonetheless, steganography remains a more than theoretical means for private communication. A few years ago, police in Berlin confiscated a password-protected memory card containing hidden files from a suspect who was undergoing questioning. After these files were decoded, they seemed to be pornographic videos, but after further investigation it turned out over a hundred documents relating to al Qaeda plots were encoded into the videos, using steganography. It took German investigators weeks of effort to discover and retrieve these documents, and it requires no stretch of the imagination to realise that many similar cases may have gone completely undetected. Cases like this necessarily prompt calls for more information about steganography, with the result that receiving funding to spend a few years 'analysing' pornography, strictly for academic purposes, is not unheard of.