Steganography Hiding Data Within Data

Gary C. Kessler

An edited version of this paper with the title "Hiding Data in Data" originally appeared in the April 2002 issue of Windows & .NET Magazine

Cryptography — the science of writing in secret codes — addresses all of the elements necessary for secure communication over an insecure channel, namely privacy, confidentiality, key exchange, authentication, and non-repudiation. But cryptography does not always provide safe communication.

Consider an environment where the very use of encrypted messages causes suspicion. If a nefarious government or Internet service provider (ISP) is looking for encrypted messages, they can easily find them. Consider the following text file; what else is it likely to be if not encrypted?

```
qANQR1DBwU4D/TlT68XXuiUQCADfj2o4b4aFYBcWumA7hR1Wvz9rbv2BR6WbEUsy
ZBIEFtjyqCd96qF38sp9IQiJIKlNaZfx2GLRWikPZwchUXxB+AA5+lqsG/ELBvRa
c9XefaYpbbAZ6z6LkOQ+eE0XASe7aEEPfdxvZZT37dVyiyxuBBRYNLN8Bphdr2zv
z/9Ak4/OLnLiJRk05/2UNE5Z0a+3lcvITMmfGajvRhkXqocavPOKiin3hv7+Vx88
uLLem2/fQHZhGcQvkqZVqXx8SmNw5gzuvwjV1WHj9muDGBY0MkjiZIRI7azWnoU9
3KCnmpR60VO4rDRAS5uGl9fioSvze+q8XqxubaNsgdKkoD+tB/4u4c4tznLfw1L2
YBS+dzFDw5desMFSo7JkecAS4NB9jAu9K+f7PTAsesCBNETDd49BTOFFTWWavAfE
gLYcPrcn4s3EriUgvL3OzPR4P1chNu6sa3ZJkTBbriDoA3VpnqG3hxqfNyOlqAka
mJJuQ53Ob9ThaFH8YcE/VqUFdw+bQtrAJ6NpjIxi/x0FfOInhC/bBw7pDLXBFNaX
HdlLQRPQdrmnWskKznOSarxq4GjpRTQo4hpCRJJ5aU7tZO9HPTZXFG6iRIT0wa47
AR5nvkEKoIAjW5HaDKiJriuWLdtN4OXecWvxFsjR32ebz76U8aLpAK87GZEyTzBx
dV+lH0hwyT/y1cZQ/E5USePP4oKWF4uqquPee1OPeFMBo4CvuGyhZXD/18Ft/53Y
WIebvdiCqsOoabK3jEfdGExce63zDI0=
=MpRf
```

The message above is a sentence in English that is encrypted using Pretty Good Privacy (PGP), probably the most commonly used e-mail encryption software today. Besides being nonsensical to a casual reader, the other indication that this is encrypted is that the characters comprising the message appear more-or-less at random and do not adhere to the relative frequency counts that one would expect in a non-encrypted message. Encrypted data sticks out like a sore thumb.

Steganography is the science of hiding information. Whereas the goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party. In this article, I will discuss what steganography is, what purposes it serves, and will provide an example using available software.

Steganography

There are a large number of steganographic methods that most of us are familiar with (especially if you watch a lot of spy movies!), ranging from invisible ink and microdots to secreting a hidden message in the second letter of each word of a large body of text and spread spectrum radio communication. With computers and networks, there are many other ways of hiding information, such as:

- Covert channels (e.g., Loki and some distributed denial-of-service tools use the Internet Control Message Protocol, or ICMP, as the communications channel between the "bad guy" and a compromised system)

- Hidden text within Web pages

- Hiding files in "plain sight" (e.g., what better place to "hide" a file than with an important sounding name in the c:\winnt\system32 directory?)

- Null ciphers (e.g., using the first letter of each word to form a hidden message in an otherwise innocuous text)

Steganography today, however, is significantly more sophisticated than the examples above suggest, allowing a user to hide large amounts of information within image and audio files. These forms of steganography often are used in conjunction with cryptography so that the information is doubly protected; first it is encrypted and then hidden so that an adversary has to first find the information (an often difficult task in and of itself) and then decrypt it.

There are a number of uses for steganography besides the mere novelty. One of the most widely used applications is for so-called digital watermarking. A watermark, historically, is the replication of an image, logo, or text on paper stock so that the source of the document can be at least partially authenticated. A digital watermark can accomplish the same function; a graphic artist, for example, might post sample images on her Web site complete with an embedded signature so that she can later prove her ownership in case others attempt to portray her work as their own.

Stego can also be used to allow communication within an underground community. There are several reports, for example, of persecuted religious minorities using steganography to embed messages for the group within images that are posted to known Web sites.

Steganographic Methods

The following formula provides a very generic description of the pieces of the steganographic process:

cover_medium + hidden_data + stego_key = stego_medium

In this context, the cover_medium is the file in which we will hide the hidden_data, which may also be encrypted using the stego_key. The resultant file is the stego_medium (which will, of course. be the same type of file as the cover_medium). The cover_medium (and, thus, the stego_medium) are typically image or audio files. In this article, I will focus on image files and will, therefore, refer to the cover_image and stego_image.

Before discussing how information is hidden in an image file, it is worth a fast review of how images are stored in the first place. An image file is merely a binary file containing a binary representation of the color or light intensity of each picture element (pixel) comprising the image.

Images typically use either 8-bit or 24-bit color. When using 8-bit color, there is a definition of up to 256 colors forming a palette for this image, each color denoted by an 8-bit value. A 24-bit color scheme, as the term suggests, uses 24 bits per pixel and provides a much better set of colors. In this case, each pix is represented by three bytes, each byte representing the intensity of the three primary colors red, green, and blue (RGB), respectively. The Hypertext Markup Language (HTML) format for indicating colors in a Web page often uses a 24-bit format employing six hexadecimal digits, each pair representing the amount of red, blue, and green, respectively. The color orange, for example, would be displayed with red set to 100% (decimal 255, hex FF), green set to 50% (decimal 127, hex 7F), and no blue (0), so we would use "#FF7F00" in the HTML code.

The size of an image file, then, is directly related to the number of pixels and the granularity of the color definition. A typical 640x480 pix image using a palette of 256 colors would require a file about

307 KB in size (640 • 480 bytes), whereas a 1024x768 pix high-resolution 24-bit color image would result in a 2.36 MB file (1024 • 768 • 3 bytes).

To avoid sending files of this enormous size, a number of compression schemes have been developed over time, notably Bitmap (BMP), Graphic Interchange Format (GIF), and Joint Photographic Experts Group (JPEG) file types. Not all are equally suited to steganography, however.

GIF and 8-bit BMP files employ what is known as lossless compression, a scheme that allows the software to exactly reconstruct the original image. JPEG, on the other hand, uses lossy compression, which means that the expanded image is very nearly the same as the original but not an exact duplicate. While both methods allow computers to save storage space, lossless compression is much better suited to applications where the integrity of the original information must be maintained, such as steganography. While JPEG can be used for stego applications, it is more common to embed data in GIF or BMP files.

The simplest approach to hiding data within an image file is called least significant bit (LSB) insertion. In this method, we can take the binary representation of the hidden_data and overwrite the LSB of each byte within the cover_image. If we are using 24-bit color, the amount of change will be minimal and indiscernible to the human eye. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

Now suppose we want to "hide" the following 9 bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in bold have been changed):

10010101	0000110**0**	11001001
1001011**1**	0000111**0**	1100101**1**
10011111	00010000	11001011

Note that we have successfully hidden 9 bits but at a cost of only changing 4, or roughly 50%, of the LSBs.

This description is meant only as a high-level overview. Similar methods can be applied to 8-bit color but the changes, as the reader might imagine, are more dramatic. Gray-scale images, too, are very useful for steganographic purposes. One potential problem with any of these methods is that they can be found by an adversary who is looking. In addition, there are other methods besides LSB insertion with which to insert hidden information.

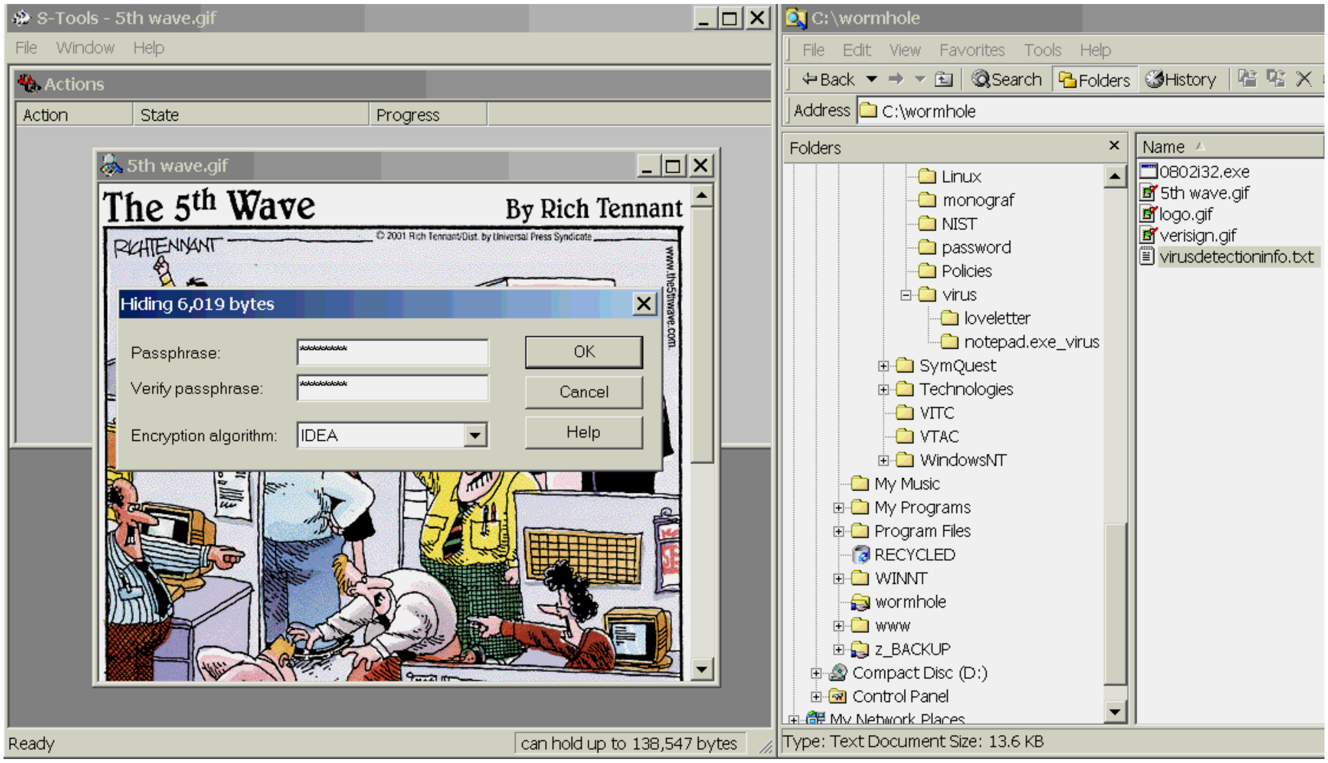
Without going into any detail, it is worth mentioning steganalysis, the art of detecting and breaking steganography. One form of this analysis is to examine the color palette of a graphical image. In most images, there will be a unique binary encoding of each individual color. If the image contains hidden data, however, many colors in the palette will have duplicate binary encodings since, for all practical purposes, we can't count the LSB. If the analysis of the color palette of a given file yields many duplicates, we might safely conclude that the file has hidden information.

But what files would you analyze? Suppose I decide to post a hidden message by hiding it in an image file that I post at an auction site on the Internet. The item I am auctioning is real so a lot of people may access the site and download the file; only a few people know that the image has special information that only they can read. And we haven't even discussed hidden data inside audio files! Indeed, the quantity of potential cover files makes steganalysis a Herculean task.

A Steganography Example

There are a number of software packages that perform steganography on just about any software platform; readers are referred to Neil Johnson's list of steganography tools at http://www.jjtc.com/Steganography/toolmatrix.htm. Some of the better known packages for Windows NT and Windows 2000 systems include:

- Hide4PGP (http://www.heinz-repp.onlinehome.de/Hide4PGP.htm)
- MP3Stego (http://www.cl.cam.ac.uk/~fapp2/steganography/mp3stego/)
- Stash (http://www.smalleranimals.com/stash.htm)
- Steganos (http://www.steganos.com/english/steganos/download.htm)
- S-Tools (available from http://www.webattack.com/download/dlstools.shtml)



Hiding 6,019 bytes

Passphrase: ********

Verify passphrase: ********

Encryption algorithm: IDEA

can hold up to 138,547 bytes

Address C:\wormhole

Type: Text Document Size: 13.6 KB

FIGURE 1
The cover_image (5th wave.gif), hidden_data file (virusdetectioninfo.txt), and stego_key.

The following examples come from Andy Brown's S-Tools for Windows. S-Tools allows users to hide information into BMP, GIF, or WAV files. The basic scheme of the program is straight-forward; you drag an image or audio file into the S-Tools active window to act as the cover_medium, drag the hidden_data file onto the cover_medium, and then provide a stego_key for encryption. The result is the stego_medium. All of this is shown in Figure 1:

1. I highlighted the GIF image file 5th wave.gif and dragged it to the S-Tools active window. Note that S-Tools reports that up to 138,547 bytes can be hidden in this image file.

2. I next highlighted a 14 KB text file called virusdetectioninfo.txt and dragged it onto the image file in S-Tools.

3. A dialog box pops up telling me that I am hiding 6,019 bytes of data and asks for a passphrase with which to encrypt the hidden text; the default secret key crypto scheme used by S-Tools is the International Data Encryption Algorithm (IDEA).

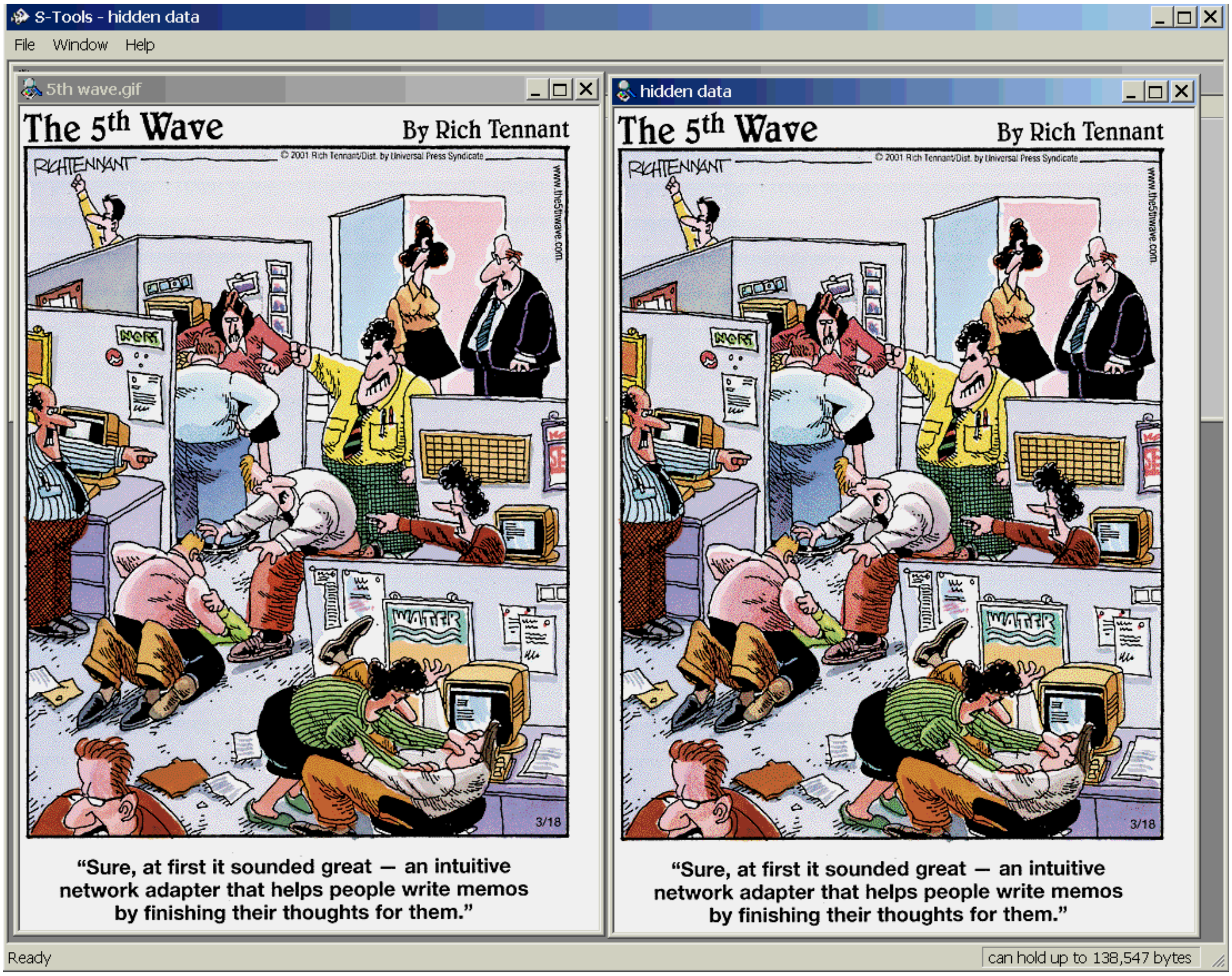
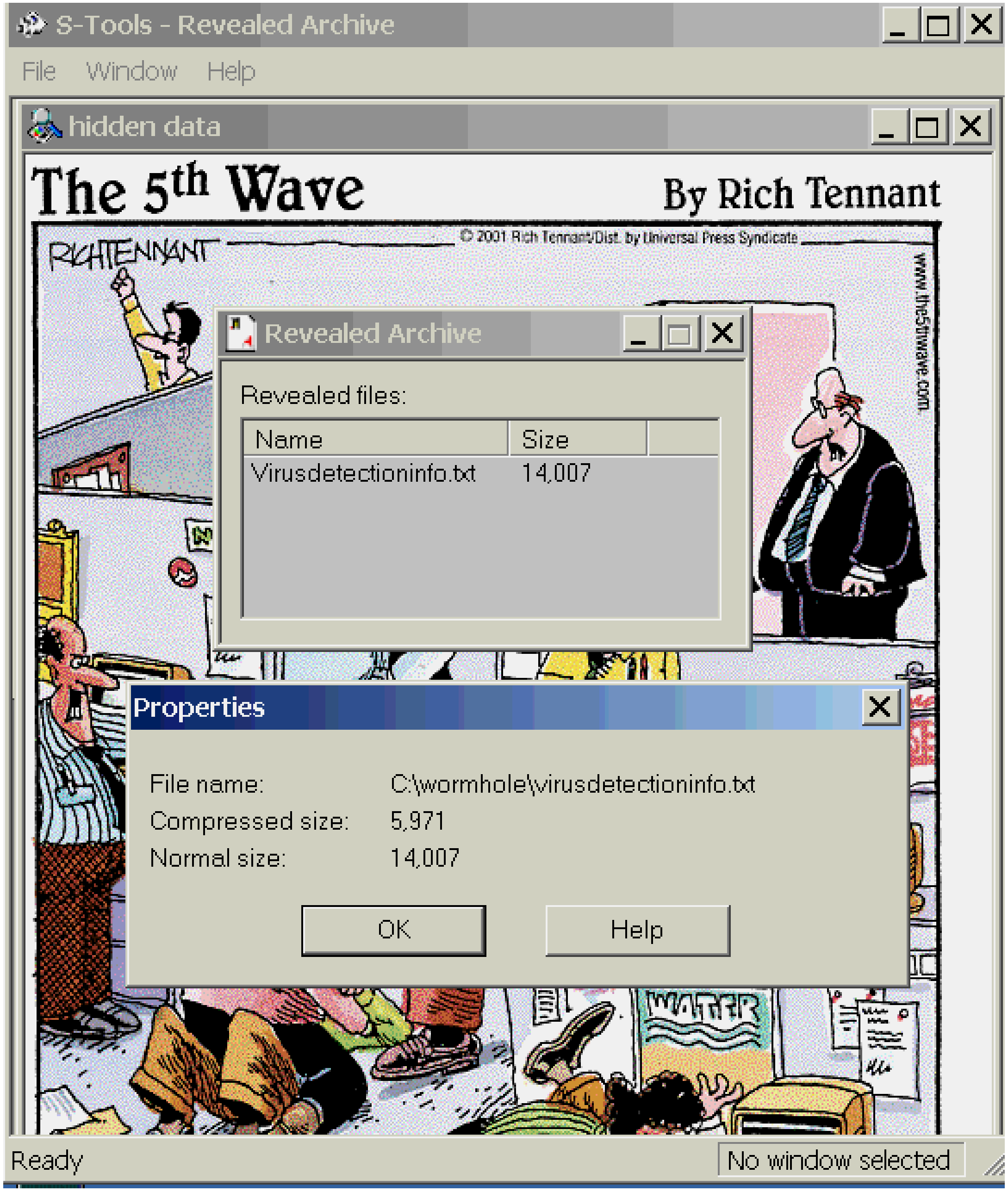

S-Tools - hidden data
File
Window
Help
5th wave.gif
hidden data
The 5th Wave
By Rich Tennant
RICHTENNANT
© 2001 Rich Tennant/Dist. by Universal Press Syndicate
www.the5thwave.com
3/18
"Sure, at first it sounded great — an intuitive network adapter that helps people write memos by finishing their thoughts for them."
Ready
can hold up to 138,547 bytes

FIGURE 2
The original image file (left) and image file with embedded text (right), side by side



Revealed files:

Name	Size
Virusdetectioninfo.txt	14,007

File name: C:\wormhole\virusdetectioninfo.txt

Compressed size: 5,971

Normal size: 14,007

Ready

No window selected

FIGURE 3. Extracting hidden information from the image file

Once the image file has been received, the user merely drags the file to S-Tools and right-clicks over the image, specifying the Reveal option. A dialog box will pop up requesting the passphrase. Figure 3 shows the information about the hidden archive file, and allows the user to open the file.

Conclusion

Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. But it is also quite real; this is not just something that's used in the lab or an arcane subject of study in academia. Stego may, in fact, be all too real — there have been several reports that the terrorist organization behind the September 11 attacks in New York City, Washington, D.C., and outside of Pittsburgh used steganography as one of their means of communication.

Two of the more recent books on the subject are Information Hiding: Steganography and Watermarking - Attacks and Countermeasures by N.F. Johnson, Z. Duric, and S. Jajodia (Kluwer Academic Publishers, 2000) and Information Hiding Techniques for Steganography and Digital Watermarking, edited by S. Katzenbeisser and F.A.P. Petitcolas (Artech House Books, 2000). One of the most informative Web sites on the subject is Neil Johnson's Steganography & Digital Watermarking Page at http://www.jjtc.com/Steganography/.

Sidebar
Other Forms of Steganography

While much of the steganography employed today is quite high-tech, steganography itself can make use of many low-tech methods. The goal of stego is merely to hide the presence of a message; remember how well the critical missive was hidden in plain sight in Poe's "The Purloined Letter"?

One common, almost obvious, form of steganography is called a null cipher. In this type of stego, the hidden message is formed by taking the first (or other fixed) letter of each word in the cover message. Consider this cablegram that might have been sent by a journalist/spy from the U.S. to Europe during World War I:

```
PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE
SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN
OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT
IMMENSELY.
```

The first letters of each word form the character string: PERSHINGSAILSFROMNYJUNEI. A little imagination and some spaces yields the real message: PERSHING SAILS FROM NY JUNE I.

Another form of steganography uses a template (e.g., a piece of paper with holes cut in it) or a set of preselected locations on the page to hide a message. In this case, obviously, the sender and receiver must use the same template or rules. Consider this note:

```
THE MOST COMMON WORK ANIMAL IS THE HORSE. THEY CAN BE USED
TO FERRY EQUIPMENT TO AND FROM WORKERS OR TO PULL A PLOW.
BE CAREFUL, THOUGH, BECAUSE SOME HAVE SANK UP TO THEIR
KNEES IN MUD OR SAND, SUCH AS AN INCIDENT AT THE BURLINGTON
FACTORY LAST YEAR. BUT HORSES REMAIN A SIGNIFICANT FIND. ON
A FARM, AN ALTERNATE WORK ANIMAL MIGHT BE A BURRO BUT THEY
ARE NOT AS COMFORTABLE AS A TRANSPORT ANIMAL.
```

Applying a template or rule as to which words to read to this message might yield the following:

```
                                   HORSE
   FERRY
                                      SANK
      IN                                         BURLINGTON
                                                   FIND
```

ALTERNATE TRANSPORT

There are other alternatives to the template method such as:

- Pinpricks in maps to use as an overlay for relevant letters in messages
- Deliberate misspelling to mark words in the message
- Use of small changes in spacing to indicate significant letters or words in a hidden message
- Use of a slightly different font in a typeset message to indicate the hidden letters (e.g., the difference between Courier and Courier New is barely noticeable unless you are looking for it)

Steganography doesn't just apply to written forms of communication. Radio and TV messages, from World War II to today, can be used to hide coded or hidden messages. Some government sources suspect that Osama bin Laden's pre-recorded videos that are re-played on TV stations around the world contain hidden messages.

Some argue that the U.S. Marine Corps Navaho code talkers of WWII represent a form of steganography. The messages themselves weren't encrypted; the plaintext was right there in the open, just in a language that was unknown by the Japanese. Disappearing ink and microdots are other ways in which messages can be hidden from the casual observer.

One of the oldest stego schemes was to shave the head of a messenger and tattoo a message on the messenger's head. After the hair grows back, the messenger can be sent to the intended recipient, where the messenger's head can be shaved and the message recovered. This method is decidingly clever, patient, and very low-tech, and goes right to the heart of steganography's literal meaning of "covered writing."