Steganography techniques

Steganography is the act of covert communications, which means that only the sender, Alice, and receiver, Bob, are aware of the secret communication. To accomplish this, the secret message is hidden within benign-looking communications known as cover texts or cover Works. To an adversary, Eve, it is clear that Alice and Bob are communicating, but the combined covertext and hidden message, referred to as a stegotext or stego Work, appears to be innocuous (i.e.,Eve is unaware that the innocuous content hides a message).

The main requirement of steganography is *undetectability*, which, loosely defined, means that no algorithm exists that can determine whether a Work contains a hidden message. Steganalysis is the process of detection of steganographic communications. And since steganography and steganalysis are closely intertwined.

Steganography and watermarking are both forms of data hiding and share some common foundations. Nevertheless, it is worth reiterating the goals of these two data-hiding applications in order to highlight the key differences.

We define steganography as the practice of undetectably altering a Work to embed a message.

We define watermarking as the practice of imperceptibly altering a Work to embed a message about that Work.

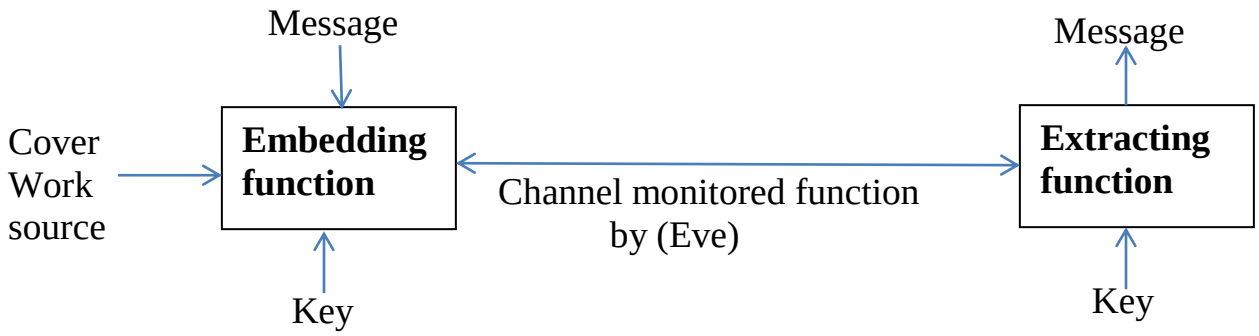
When designing a steganographic scheme, we need to consider issues such as the properties of the communication channel, the source of cover Works, and the embedding/extraction function.

The central concept in steganography is statistical undetectability. Without a precise definition, the field of steganography would lack the criterion to evaluate how secure steganographic schemes really are.

STEGANOGRAPHIC COMMUNICATION

Two prisoners, Alice and Bob, are under the surveillance of a warden, Eve. The warden will permit Alice and Bob to communicate, but all communications must go through the warden. If the warden thinks that Alice's message to Bob is innocuous, she may simply forward it to Bob. Alternatively, she may intentionally distort the content (e.g., apply lossy compression) in the hope that such a distortion will remove any secret message that just might be present. If the warden thinks Alice's message to Bob hides a covert communication, then she may block the communication entirely.

This framework, which models the applications depicted in Figure (1). A number of different assumptions can be made regarding the channel, the source of cover Works, and the embedding and extraction functions.



. **Figure (1) Steganographic embedding scheme**

The Channel

In steganography, the physical channel used for communication is generally assumed noise free, as this can be ensured using error correction and standard Internet protocols. Instead, the channel's properties are defined by the warden. The warden is considered a part of the channel because she may, or may not, interfere with the communication. As such, there are three types of warden:
passive, active, and malicious.

The warden is called **passive** if she is restricted from modifying the content sent by Alice prior to receipt by Bob (i.e., the warden can only prevent or permit delivery of Alice's message). In this scenario, the warden tests each communication from Alice for the presence of a covert message. If the warden's test is negative, the communication is relayed to Bob. Otherwise it is blocked.

This is the most commonly assumed scenario and why most steganographic algorithms are not designed to be robust.

The warden is called **active** if she intentionally modifies the content sent by Alice prior to receipt by Bob. In this scenario, the warden may not be entirely confident of her steganalysis program. Thus, even though her tests are negative, the warden may alter the content, hoping that the modification will destroy any steganographic message that *might* be present. If the steganographic algorithm assumes a passive warden, then there is a good chance that alterations to the content will severely degrade or remove the hidden message. The types of modification an active warden might apply include lossy recompression of images and audio clips, low-pass filtering, and other procedures that slightly degrade the content.

The warden is called **malicious** if her actions are based on the specifics of the steganographic scheme and are aimed at catching the prisoners communicating

secretly. This may include the warden trying to impersonate Alice or Bob or otherwise tricking them. A malicious warden is usually considered in public-key

steganography. In this scenario, the *stego* key is known and anyone can extract the secret message. However, the message is encrypted using a public-key cryptosystem. Only those who possess Bob's private key can decipher Alice's message. Even though the stego key is known, it is difficult to distinguish between an encrypted message and a random bit sequence extracted from a cover Work. Nevertheless, since the Warden also knows the stego key, she has more options to attack the stego system.

The Building Blocks

The main building blocks of any steganographic algorithm are:

1. The choice of the cover Work.

2. The embedding and extracting algorithms, which might include

a. Symbol assignment function.

b. The embedding modification.

c. The selection rule.

3. Stego key management.

We now discuss each of these design elements in more detail.

Unlike a watermark, a steganographic message says nothing about the cover Work in which it is hidden. Consequently, the steganographer is free to choose a particular cover Work from his or her source of covers. The main restriction is the source of cover Works, which is determined by the resources available to Alice and Bob, by the warden herself, and the context in which the communication takes place. For example, an oppressive regime can specify allowable forms of messages and Alice must comply with them to avoid being caught. Or, if Alice and Bob communicate by posting images to a discussion newsgroup, they must choose the covers among those that are typically posted. But even with these restrictions, there are still numerous cover Works in which to hide the covert message. Alice is therefore at liberty to choose the cover Work which, after embedding, has the least likelihood of being detected.

For example, it is intuitively clear that noisy or highly textured images will better mask any embedding changes than high-quality images with little content (e.g., blue sky images). Alternatively, Alice can think ahead and attempt to guess what tests the warden is going to use and embed the same message into many different covers, run known steganalysis attacks on each stego Work, and then simply send the cover that passes the tests.

Fundamentally, an embedding function can be based on three different principles, namely:

1. The cover Works are preexisting and the embedder does *not* modify the cover Works. This is referred to as steganography by cover lookup.

2. The cover Works are generated based on the hidden message and the embedder does *not* modify the cover Works. This is referred to as cover synthesis.

3. The cover Works are preexisting and the embedder modifies the cover Works. This is referred to as steganography by cover modification.

Steganography by cover modification describes methods where Alice alters an existing cover Work to create a stego Work that conveys the desired message.

This approach is both the most common and the most advanced. we will only focus on this class of steganographic algorithms.

The type of changes introduced by the embedder, together with the location of these changes within the cover Work, have a major influence on how inconspicuous the embedded message will be. Intuitively, changes of large magnitude will be more obvious than changes of smaller magnitude. Consequently,

most steganographic schemes try to modify the cover Work as little as possible.

The location of the changes is controlled by the *selection rule*. There are three types of selection rules: sequential, (pseudo) random, and adaptive.

A sequential selection rule embeds the message bits in individual elements of the cover Work in a sequential manner, for example, starting in the upper left corner of an image and proceeding in a row-wise manner to the lower right corner. Although the sequential selection rule is the easiest one to implement, it provides poor security, since steganalysis algorithms can inspect the statistical properties of pixels in the same order, looking for a sudden change in behavior.

A pseudo-random selection rule embeds the message bits in a pseudo randomly

selected subset of the cover Work. The sender might first use a secret stego key, *Ks*, to initialize a pseudo-random number generator (PRNG) that in turn generates a pseudo-random walk through the cover Work. The message bits are then embedded into the elements constituting this walk. Pseudo-random selection rules typically offer better security than sequential rules.

An adaptive selection rule embeds the message bits at locations that are determined based on the content of the cover Work. The motivation for this is that statistical detectability is likely to depend on the content of the cover Work as well The process of embedding is controlled by a secret key shared between Alice and Bob. The key can be used for several different purposes. As previously mentioned, the key may seed a pseudo-random number generator to generate a random walk through the cover Work. It can also be used to generate other pseudo-random entities needed for embedding.

The primary goal of steganography is to design embedding functions that are statistically undetectable and capable of communicating practical (i.e., large) payloads.

NOTATION AND TERMINOLOGY

We now mathematically define a steganographic scheme. Let *Ks* denote a stego key drawn from a set, K, of all secret stego keys, M the set of all embeddable messages, and C the set of all cover Works. A steganographic scheme is formed by two mappings, the embedding mapping, *Emb*, and the extraction mapping, *Ext*:

Emb: C ×K×M→C

Ext: C →M,

such that *Ext*(*Emb*(**c**, *Ks* ,**m**)) = **m** for all **c** ∈ C, *Ks* ∈ K, and **m**∈M. The Work **s** = *Emb*(**c**, *Ks* ,**m**) is called the stego Work.

The embedding algorithm *Emb* takes the cover Work, the secret key, and the message as its input and produces the modified stego Work.

Cover Work may be (text, image, audio, video).

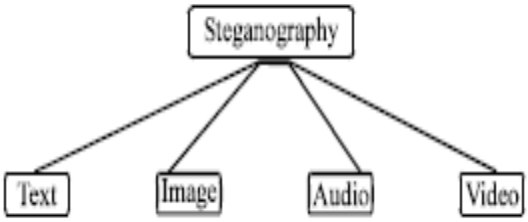


IMAGE STEGANOGRAPHY

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain. Image - also known as spatial - domain techniques embed messages in the intensity of the pixels directly, while for transform - also known as frequency - domain, images are first transformed and then the message is embedded in the image.

Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterized as simple systems. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format.

Steganography in the transform domain involves the manipulation of algorithms and image transforms. These methods hide messages in more significant areas of the cover image, making it more robust. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression. In the next sections steganographic algorithms will be explained in categories according to image file formats and the domain in which they are performed. Figure(2) indicate the possibility of using images as a cover carrier.

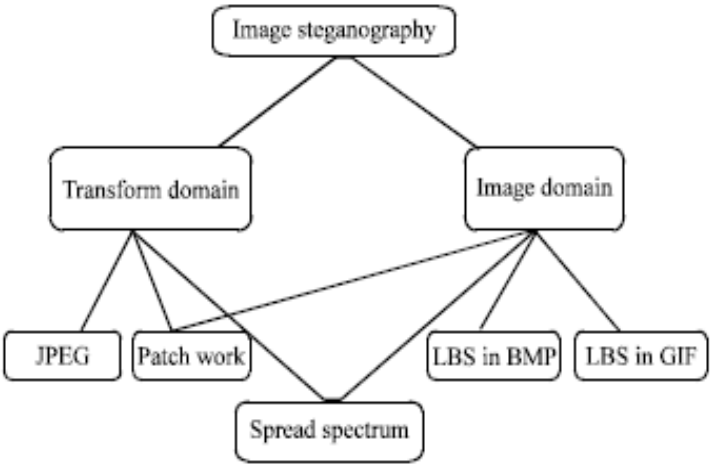


Figure (2) Image based steganography

Substitution Systems

Basic substitution systems try to encode secret information by substituting insignificant parts of the cover by secret message bits. The receiver can extract the information if he has knowledge of the positions where secret information has been embedded. Since only minor modifications are made in the embedding process, the sender assumes that they will not be noticed by an attacker. It consists of several techniques that will be discussed in more detail, in the following subsection:

Least Significant Bit Substitution (LSB)

The embedding process consists of choosing a subset {j1… jl(m)} of cover elements and performing the substitution operation cji _ mi on them, which exchange the LSB of cji by mi (mi can be either 1 or 0). In the extraction process, the LSB of the selected cover-element is extracted and lined up to reconstruct the secret message.

In the case of a 24-bit bitmap each pixel is represented by 4 bytes. Of those, 3 bytes, or 24 bits, are used to store the red, green and blue values for the pixel. The fourth byte is reserved and should be zero. To store each character in the low order bit plane of the raster data, it is necessary to obtain an 8 bit representation of the character. For example, the character A is represented by the number 65. The equivalent binary representation is 0100 0001. Each of the 8 bits used to represent the letter A is then stored in the low order bit of one byte of raster data. Thus, to store a single letter, 8 bytes of raster data are consumed. This leads to a limit of embeddable information of size length Of Raster Data/8. Consider hiding the letter A in the first 8 bits of raster data of an image. The first 8 bytes could possibly be (from left to right, top to bottom):

‘1001 1001’ ‘1110 0011’ ‘0110 1001’ ‘0001 1100’
‘0001 1100’ ‘0110 0100’ ‘1011 0000’ ‘1010 1001’

And the character A is:

‘0100 0001’

Therefore, we need to set bits 7, 5, 4, 3, 2 and 1 to zero,

Although the resulting bit has not changed, we have ensured that the least significant bit has been set to ‘1’. Because the byte values for the red, green and blue pixels will only change by at most 1, the change in the resulting image will be imperceptible to the human eye. The resulting image will not, however, be well protected against statistical attack.

Pseudo Random Permutation

If all cover bits are accessed in the embedding process, the cover is a random access cover and the secret message bits can be distributed randomly over the whole cover. This technique further increases the complexity for the attacker, since it is not guaranteed that the subsequent message bits are embedded in the same order.

Image Downgrading and Cover Channels

Image downgrading is a special case of a substitution system in which image acts both as a secret message and a cover. Given cover-image and secret image of equal dimensions, the sender exchanges the four least significant bits of the cover gray-scale (or colour) values with the four most significant bits of the secret image. The receiver extracts the four least significant bits out of the stego-image, thereby gaining access to the most significant bits of the stego-image. Whereas, the degradation of the cover is not visually noticeable in many cases, four bits are sufficient to transmit a rough approximation of the secret image.