University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Information Hiding and Watermarking
إخفاء المعلومات والعلامة المائية

Lect. Prof. Dr. Nidaa Flaih Hassan
أ.د. نداء فليح حسن

cs.uotechnology.edu.iq

Information Hiding & Watermarking – 1ST Course	
Part I : Information Hiding and Steganography	
Chapter_1	**Introduction to Information Hiding** – Introduction – Main Sub-disciplines of Information Hiding – A Brief History of Information Hiding – Linguistic Steganography in Ancient time – Copyright Enforcement – Wisdom from Cryptography – Some Applications of Information Hiding
Chapter_ 2	**Secret Writing and Steganography** – Principles of Digital Steganography – Frameworks for Secret Communication • Pure Steganography • Secret Key Steganography • Public Key Steganography – Characterization of Steganography Systems
Chapter_3	**Information Hiding in Noisy Data & Survey of Steganographic Techniques**
3.1	**Substitution Systems** – Least Significant Bit Substitution – Pseudorandom Permutations – Image Downgrading and Covert Channels – Palette-Based Images – Information Hiding in Binary Images – Unused or Reserved Space in Computer Systems – Transform Domain Techniques
3.2	**Transform Domain Techniques**
3.3	**Spread Spectrum and Information Hiding** **Statistical Steganography** **Distortion Techniques** **Cover Generation Techniques** **Active and Malicious Attackers**
Chapter 4	**Information Hiding in Written Text**

Part II : Watermarking and Copyright Protection	
Chapter 5	**Watermarking and Copyright Protection**
Chapter 6	**Basic Watermarking Principles and Requirements and Algorithmic Design Issues**
6.1	**Basic Watermarking Principles** – **Watermarking and Copyright Protection** – Watermarking Terminology – Basic Watermarking Principles – Watermarking Applications ➢ Watermarking for Copyright Protection ➢ Fingerprinting for Traitor Tracking ➢ Watermarking for Copy Protection ➢ Watermarking for Image Authentication
6.2	**Requirements and Algorithmic Design Issues**
Chapter_ 7	**Evaluation and Benchmarking of Watermarking Systems** – Introduction – Perceptibility of the Watermarks

References

[1] Stefan Katzenbeisser, and Fabien A. P. Petitcolas, " Information Hiding Techniques for Steganography and Digital Watermarking" , Artech House computing library, 2000.

[2] Frank Y. Shih "Digital Watermarking and Steganography: Fundamentals and Techniques", Taylor & Francis Group, LLC CRC Press is an imprint of Taylor & Francis Group, 2017

Introduction to Information Hiding

1. Introduction

The rapidly decreasing cost of processing, storage, and bandwidth has already made digital media increasingly popular over traditional analog media. Audio, video and other works become available in digital form, and the ease with which perfect copies can be made may lead to large-scale unauthorized copying which might weaken the music, film, book, and software publishing industries.

These concerns over protecting copyright have triggered significant research to find ways to hide copyright messages and serial numbers into digital media; the idea is that the latter can help to identify copyright violators (منتهكي), and the former to prosecute (محاكمه) them. At the same time, moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly harmless cover messages.

2. Information Hiding

Information hiding (i.e., data embedding) is a communication issue with two important parts: signal sources and communication channels.

The concept of information hiding is to conceal an important secret message in public information. Figure (1.1) shows the classification of information hiding technologies

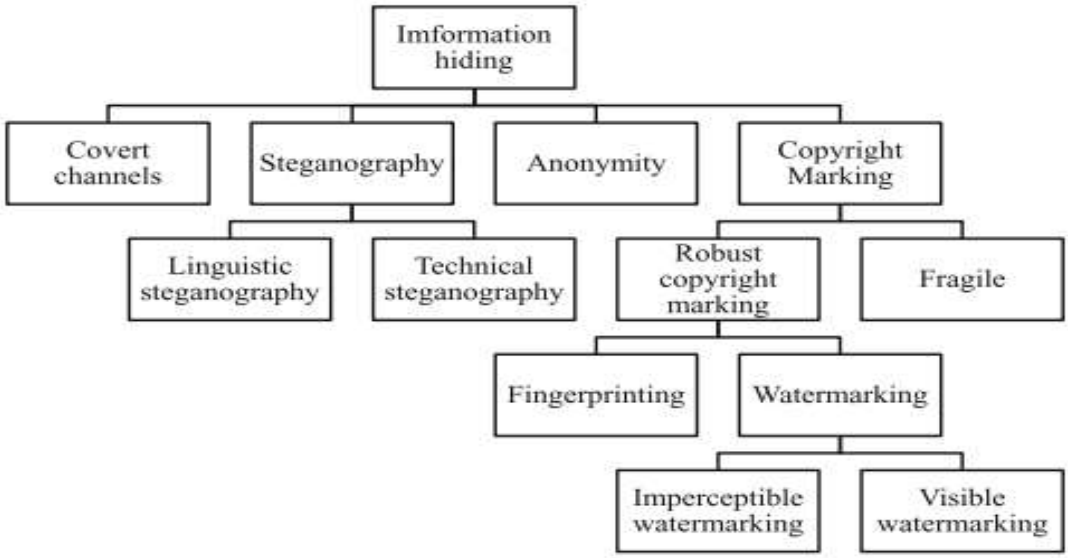


Figure (1.1): Classification of information hiding technologies. The four types of information hiding technologies are covert channels, steganography, anonymity, and copyright marking

3. Main Sub-disciplines of Information Hiding

The different disciplines of information hiding are summarized in the following section:

- **Covert Channels** (**القنوات السرية**) have been defined by Lampson, as communication paths that were neither designed nor intended to transfer information at all. These channels are typically used by untrustworthy (**غير جدير بالثقة**) programs to leak information to their owner while performing a service for another program. Thus a covert channel is any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy. In short, covert channels transfer information using non-standard methods against the system design

- **Anonymity** (مجهول الهويه) is finding ways to hide the meta content **(المحتوى** التعريفي) of messages, that is, the sender and the recipients of a message. The Anonymity is when nobody knows who you are but potentially they know what you are doing. When one keeps their actions and activities separate from their identity. An author who is not releasing his name is an example of maintaining someone maintaining anonymity
- **An** important sub-discipline of information hiding is **Steganography**. Steganography is the science of hiding a secret message in cover media, without any perceptual distortion of the cover media. Using steganography, information can be hidden in the carrier items such as images, videos, sound files, and text files, while performing data transmission".
- **Watermarking**, A digital watermark is a kind of marker covertly embedded in signal such as audio, video or image data. It is typically used to identify ownership of the copyright of such signal. Watermarking has the additional requirement of **robustness** against possible attacks. Robustness has strong implications for the overall design of a watermarking. Watermarks do not always need to be hidden, as some systems use: V***isible digital watermarks,*** but most of the literature has focused on ***Imperceptible*** **(invisible, transparent, or inaudible, depending on the context)** digital watermarks whichhave wider applications. Visible digital watermarks are strongly linked to the original paper watermarks which appeared at the end of the 13th century. Modern visible watermarks may be visual patterns (e.g., a company logo (شعار) or copyright sign) overlaid on digital images and are widely used by many photographers who do not trust invisible watermarking techniques.

From this brief overview, its noticed another fundamental difference between steganography and watermarking, and they are :

- The information hidden by a watermarking system is always associated with the digital object to be protected or to its owner while steganography systems just hide any information.
- The ***"robustness"*** criteria are also different since steganography is mainly concerned with the detection of the hidden message while watermarking concerns potential removal by a pirate قرصان
- Finally, ***stenographic communications*** are usually point-to-point (between sender and receiver) while watermarking techniques are usually one-to-many.

Steganography and Watermarking techniques are widely used to hide the original message for secure communication. Cryptography means the sender converts plaintext to cipher text by using the encryption key and the receiver side requires the receiver to decrypt cipher text to plain text by using a technique known as cryptanalysis. The purpose of Steganography, cryptography, and watermarking is shown in **Table (1.1). The c**omparison between Steganography, Cryptography, and Watermarking **shows in Table (1.2).**

Table (1.1): The purpose of Steganography, Cryptography, and Watermarking

Technique	Purpose
Steganography	Hiding existence of digital content from outsiders
Cryptography	Rendering the digital content inaccessible to outsiders
Watermarking	Protection of digital content of carrier

Table (1.2) : Comparison between Steganography, Cryptography, and Watermarking

Factors	Steganography	Cryptography	Watermarking
Definition	Steganography is the art of hiding information behind the media file	Cryptography is the art of achieving security by encoding messages to make them non-readable.	Watermarking is The art and science of hiding information
Techniques	LSB, Spatial Domain, Jsteg, Outguess	Transposition, substitution, Stream ciphers, Block ciphers.	Spatial domain, Fragile watermarking, DCT.
Security	No one can percept the hidden message	No one can read the message without knowing the key	Verify the authenticity or integrity of the digital files or to show the identity of its owners
Carrier	Any digital media	Usually text	Usually Image
Secret key	Not necessary	Necessary	Not necessary
Robustness	Yes	Yes	Yes
Type of attack	Steganalysis	Cryptanalysis	Synchronization attacks, stochastic attacks
Output	Stego Media	Cipher Media	Watermarked Media
Fails	When it is detected	De-ciphered	When it is detected
Key length	Small	Very large	Small
Naked eye Identification	No, cannot be possible because the message is kept hidden within the media file.	Yes, can be possible because the original message is converted into cipher text.	Yes, as an actualmessage is Hidden bysome watermark.

4. A Brief History of Information Hiding

In the following section, the history of information hiding is covered; rather just gives the important landmarks.

4. 1. Technical Steganography*:* The most famous examples of steganography go back to antiquity (العصور القديمه) .

- Histiaeus (Ancient Greek) shaved the head of his most trusted slave and tattooed it with a messagewhich disappeared after the hair had regrown.
- Herodotus warned Sparta by removing the wax from a writing tablet, wrote his message on the wood underneath, and then covered the message **with wax**. The tablet looked exactly like a blank one.
- A large number of techniques were invented including :
  - Notes carried by pigeons (الحمام).
  - **Hiding text by changing the heights of letter strokes or by making very small holes above or below letters in a cover text. This technique** was still in use during the 17th century but was improved by using invisible ink to print very small dots instead of making holes.
- Microscopic images were hidden in ears, nostrils, and under fingernails.
- Another example comes from architecture: since its early days, artists have understood that works of sculpture or painting appear different from certain angles, and established rules for perspective (**المنظور**) and anamorphosis (**التشوه**).

4. 2. Linguistic Steganography in Ancient time

Linguistic steganography is **a branch of Information Hiding (IH) using written natural language to conceal secret messages**. It plays an important role in Information Security (IS) area. Previous work on linguistic steganography was mainly focused on steganography and there was little research on attacks against it.

The most famous example :

1. A widely used method of linguistic steganography is the **acrostic (قصيده).** Expanding on the simple idea of the acrostic, conceal messages mainly into text. Each letter of the plain text is replaced by the word or phrase that appears

in the corresponding table entry and the stego-text ends up looking like a prayer, a simple correspondence letter, or a magic spell such as (Figure 1.3). one can hide secretly a message into a geometric drawing using points, lines, or triangles: "the point, the ends of the lines and the angles of the figures do each of them by their different situation express several letters" .

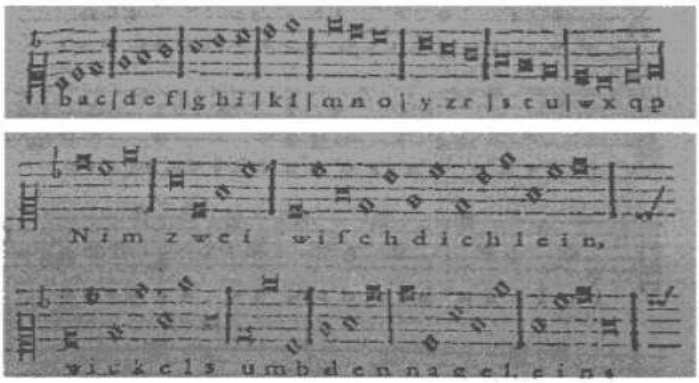


Figure (1.3): Example of Hiding information in music scores

2. An **improvement** is made when the message is hidden at **random locations** in the cover text. This idea is the core of many current stenographic systems.
3. An early example each letter is encoded in a five-bit binary code and embedded in the cover text by printing the letters in either normal or italic fonts.
4. Further examples come from the world of mathematical tables.

4. 3. Copyright Enforcement (إنفاذ حقوق الطبع والنشر)

The enforcement of copyright is the responsibility of the copyright holder. Copyright is essentially a private right and it is the rightsholder themselves who must take legal action against someone who infringes their rights.

A last example of an old solution that is being reused against forgery and for copy protection is the catalog of signed images by Lorraine. Lorraine kept a book

that he called the Liber Veritatis. The Liber Veritatis was a collection of drawings in the form of a sketchbook. The book was specially made for him, with a scheme of alternating pages, four blue pages followed by four white, which repeated in this manner and contained around 195 drawings.

Similar techniques are being used today. Image Lock, for instance, keeps a central database of image digests and periodically searches the Web for images having the same digest.

4. 4. Wisdom from Cryptography

Although steganography is different from cryptography, we can borrow many of the techniques and much practical wisdom from the latter, a more thoroughly researched discipline. In 1883, Auguste Kerckhoffs enunciated the first principles of cryptographic engineering, in which he advises that we **assume the method used to encipher data is known to the opponent, so security must lie only in the choice of key**. Applying this wisdom, we obtain a tentative definition of a secure stego- system: **one where an opponent who understands the system, but does not know the key, can obtain no evidence (or even grounds for suspicion)that communication has taken place**.

5. Some Applications of Information Hiding:

The following are five applications which are used IH techniques :

1. **Military communications** use techniques such as spread spectrum modulation or meteor scatter transmission to make signals hard for the enemy to detect or jam. As a side effect, law enforcement and counterintelligence agencies are interested in understanding these technologies and their weaknesses, so as to detect and trace hidden messages.

2. Information hiding techniques can also be used in situations where **plausible**

deniability (إنكار معقول) is required. "The obvious motivation for plausible deniability is when the two communicating parties are engaged in an activity which is somehow illicit, and they wish to avoid being caught" but more legitimate motives include fair voting, personal privacy, or limitation of liability..

3. **Anonymous communications (**الاتصالات المجهولة **)** , including anonymous remailers and Web proxies, are required by legitimate users to vote privately in online elections, make political claims, preserve online free speech, or to use digital cash. But the same techniques can be abused for defamation, blackmail, or unsolicited commercial mailing.

4. The **healthcare industry and especially medical imaging systems** may benefit from information hiding techniques. They use standards such as DICOM (Digital Imaging and Communications in Medicine) which separates image data from the caption, such as the name of the patient, the date, and the physician. Sometimes the link between the image and the patient is lost, thus, embedding the name of the patient in the image could be a useful safety measure.

5. In many cases they can use information hiding techniques already developed **for copyright marking directly**; they include the following applications: :

- **Automatic monitoring of copyrighted material on the Web:** A robot searches the Web for marked material and hence identifies potential illegal usage. An alternative technique downloads images from the Internet computes a digest of them and compares this digest with digests registered in its database.
- **Automatic audit of radio transmissions:** A computer can listen to a radio

station and look for marks, which indicate that a particular piece of music, or advertisement, has been broadcast.

- **Data augmentation:** Information is added for the benefit of the public. This can be details about the work, annotations, other channels, or purchasing information (nearest shop, price, producer, etc.) so that someone listening to the radio in a car could simply press a button to order the compact disc. This can also be hidden information used to index pictures or music tracks to provide more efficient retrieval from databases.
- **Tamper proofing:** The information hidden in a digital object may be a signed "Summary" of it, which can be used to prevent or detect unauthorized modifications.

Secret Writing and Steganography

1. Principles of Digital Steganography

Digital Steganography aims at hiding digital information in covert channels so that one can conceal the information and prevent the detection of the hidden message. The term **steganography** is derived from the Greek words words staganos and graphein, meaning "covered, concealed, or protected" and "writing," respectively."

Figure (2.1) depicts a classic steganographic model presented by Simmons

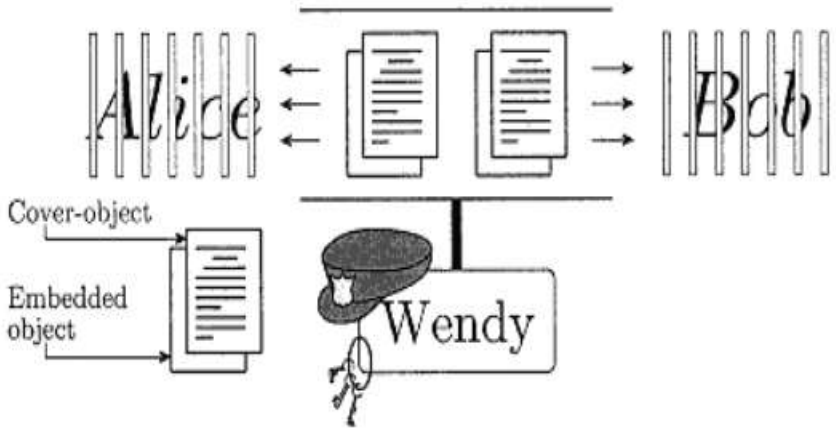


Figure (2.1): The Classic Steganographic Model

The "classic" model for invisible communication was first proposed by Simmons as the **"prisoners' problem".**

Alice and **Bob** are planning to escape from jail. All communications between them are monitored by the warden **Wendy**, so they must hide the messages in other harmless-looking media (cover objects) in order to obtain each other's stegoobjects. The stego-objects are then sent through public channels. Wendy is

free to inspect all messages between Alice and Bob in one of two ways: **passively** or **actively**.

- The passive approach involves inspecting the message in order to determine whether it contains a hidden message and then to take proper action.
- The active approach involves always altering Alice's and Bob's messages even if Wendy may not perceive any traces of hidden meaning. Examples of the active method would be image-processing operations such as **lossy compression**, **quality-factor alteration**, **format conversion**, **palette modification**, and **low-pass filtering**.

Steganalysis is the art of discovering the existence of hidden information; Steganalysis systems are used to detect whether an image contains a hidden message. By analyzing the various features of stego-images (those containing hidden messages) and cover images (those containing no hidden messages), a Steganalysis system is able to detect stego-images.

2. Frameworks for Secret Communication

Most applications of steganography follow one general principle, illustrated in Figure 2.2.

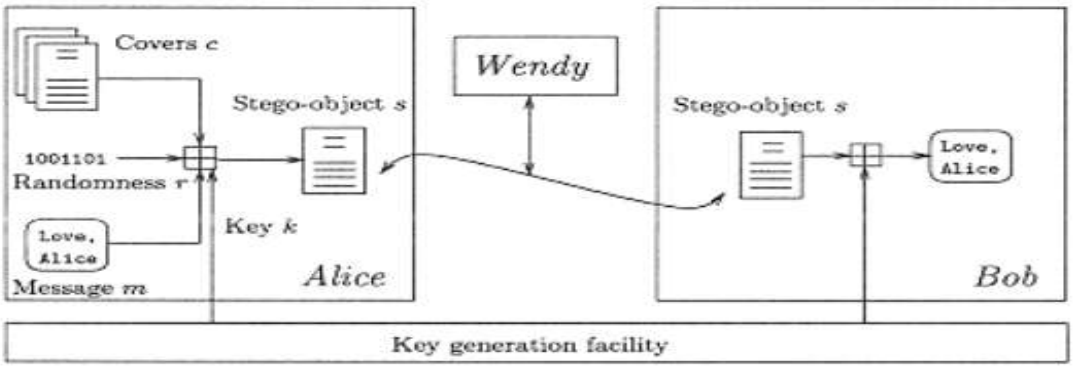


Figure (2.2): Schematic description of steganography: Alice randomly chooses a cover c using her private random source r and embeds the message m in c using a key k, creating the stego-object s which she passes on to Bob. Bob reconstructs m with the key k he shares with Alice.

Alice, who wants to share a secret message m with Bob, **randomly chooses (using the private random source r) a harmless message c**, called **cover-object**, which can be transmitted to Bob without raising suspicion and embeds the secret message into **c**, probably by using a key **k**, called **stego-key**. Alice, therefore , changes the **cover c** to a **stego-object s**.

This must be done in a very careful way, so that **a third party, knowing only the apparently harmless message s, cannot detect the existence of the secret**.

- **In a "perfect" system, a normal cover should not be distinguishable from a stego-object, neither by a human nor by a computer looking for a statistical pattern. Theoretically, covers could be any computer-readable data such as image files, digital sound, or written text.**

Alice then transmits s over an insecure channel to Bob and hopes that Wendy will not notice the embedded message. Bob can reconstruct **m** since he knows the embedding method used by Alice and **has access to the key k** used in the embedding process. **This extraction process should be possible without the original cover c.**

A third person watching the communication should not be able to decide whether the sender is active in the sense that he sends covers containing secret messages rather than covers without additional information. More formally, if an observer has access to a set {c1, . . . , cn} of cover-objects transmitted between both communication parties, **he should be unable to decide which cover-objects ci contain secret information.** Thus, the security of invisible communication lies mainly in the inability to distinguish cover- objects from stego-objects.

- **In practice, however, not all data can be used as cover for secret**

communication, since the modifications employed in the embedding process should not be visible to anyone not involved in the communication process. **This fact requires the cover to contain sufficient redundant data, which canbe replaced by secret information.**

- As an example, due to measuring errors, any data which are the result of some physical scanning process will containa stochastic (العشوائيه) component called noise. Such **random artifacts can be used for the submission of secret information**. **It turns out that noisy data has more advantageous properties in most steganographic applications.**

- **A cover should never be used twice** since an attacker who has access to two "versions" of one cover can easily detect and possibly reconstruct the message. To avoid accidental reuse, both sender and receiver should destroy all covers they have already used for information transfer.

In the literature there are three types of steganographic protocols: **Pure Steganography, Secret Key Steganography,** and **Public Key Steganography**; the latter is based on principles of public key cryptography. In the following subsections, all three types will be discussed.

1. Pure Steganography

Pure Steganography is a Steganography system that doesn't require prior exchange of some secret information before sending a message; therefore, no information is required to start the communication process: the security of the system thus depends entirely on its secrecy. Pure Steganography can be defined as the quadruple (C, M, D, and E) where:

C: the set of possible covers.

M: the set of secret messages with $|C| \geq |M|$.

E: $C \times M \rightarrow C$ the embedding function

D: C→M of the extraction function with the property that D (E(c,m))=m for all m Є M and c Є C

It is necessary that |C| ≥ |M|. Both senderand receiver must have access to the embedding and extraction algorithm, but the algorithms should not be public.

Definition (Pure steganography) *The quadruple* $\mathfrak{S} = <C, M, D, E>$, *where C is the set of possible covers, M the set of secret messages with* $|C| \geq |M|$, $E : C \times M \rightarrow C$ *the embedding function and* $D : C \rightarrow M$, *the extraction function, with the property that* $D(E(c,m)) = m$ *for all* $m \in M$ *and* $c \in C$ is called a pure steganographic system.

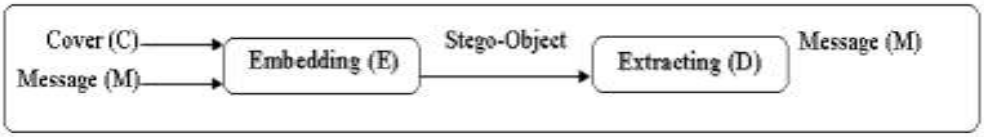


Figure (2.3): Pure Steganography

In most practical steganographic systems set , the following requriment must considered:

- C is chosen to consist of meaningful, and harmless messages (like the set of all meaningful digital images)
- Two communication partners would be able to exchange without raising suspicion.
- The embedding process is defined in a way that a cover and the corresponding stego-object are perceptually similar. Formally, perceptual similarity can be defined via a similarity function:

Definition (Similarity function) *Let C be a nonempty set. A function* $sim : C^2 \rightarrow (-\infty, 1]$ *is called similarity function on C, if for* $x, y \in C$

$$sim(x, y) = 1 \quad \Leftrightarrow \quad x = y$$

For $x \neq y$, $sim(x, y) < 1$

In the case of digital images or digital sound, the correlation between two signals

can be used as a similarity function. Therefore, most practical steganographic systems try to fulfill the condition sim(c, E(c, m)) ≈ 1 for all m ∈ M and c ∈ C. The following issues must be studied when designed Pure Steganography:

- **Covers which have not been used before should be private to the sender** (i.e., an attacker should not have access to the covers used for secret communication). For instance, the sender could create covers through the use of recording or scanning techniques.
- **For every communication process, a cover is randomly chosen**. Rather than selecting one cover at random the sender could also look through the database of usable covers and select one that the embedding process will change the least. Such a selection process can be done via the similarity function sim. In the encoding phase, the sender chooses a cover c with the property

$$c = \max_{x \in C} sim(x, E(x, m))$$

- Some researchers propose **public cover databases**. Since an attacker who has access to the original version of a cover can easily detect the secret, the sender chooses one element c out of the database and performs some modifications to get a cover c′. He then uses this new cover for secret communication. This method, however, **is not free of dangers**. If an attacker has knowledge of the modification techniques used, he can create the "plain" cover (i.e., the cover without the secret information) himself and break the communication. Even if he does not know the techniques which have been applied, he could create a similar cover by comparing c to the stego-object.
- Some steganographic methods combine traditional cryptography with steganography: **the sender encrypts the secret message before the embedding process**. Such a combination increases the security of the overall

communication process, as it is more difficult for an attacker to detect embedded ciphertext (which itself has a rather random appearance) in a cover. Strong steganographic systems, however, do not need prior enciphering

2. Secret Key Steganography

With pure steganography, no information (apart from functions E and D) is required to start the communication process; the security of the system thus depends entirely on its secrecy. **Pure Steganography not very secure in practice because this violates Kerckhoffs' principle, Why?**

Thus, must assume that Wendy knows the algorithm Alice and Bob use for information transfer. In theory, she can extract information from every cover sent between Alice and Bob. **The security of a steganographic system should thus rely on some secret information traded by Alice and Bob, the stego-key.** Without knowledge of this key, nobody should be able to extract secret information from the cover.

A secret key steganography system is similar to a symmetric cipher: thesender chooses **a cover c** and embeds the secret message into **c** using a **secret key k**. If the key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message. Anyone who does not know the secret key should not be able to obtain evidence of the encoded information. Again, the cover c and the stego-object can be perceptually similar.

The secret key Steganography can be defined as the quintuple (C, M, K, DK, EK) where: C: the set of possible covers. M: the set of secret message. K: the set of secret keys. Ek: C×M×K→C With the property that DK (EK(c,m,k),k)=m for all m Є M, c Є C and k Є K

Definition **(Secret key steganography)** *The quintuple* $\mathfrak{S} = \langle C, M, K, D_K, E_K \rangle$, *where C is the set of possible covers, M the set of secret messages with* $|C| \geq |M|$, *K the set of secret keys,* $E_K : C \times M \times K \rightarrow C$ *and* $D_K : C \times K \rightarrow M$ *with the property that* $D_K(E_K(c, m, k), k) = m$ *for all* $m \in M$, $c \in C$ *and* $k \in K$, *is called a secret key steganographic system.*

- Secret key steganography requires the **exchange of some keys**, although the transmission of additional secret information subverts تخريب the original intention of invisible communication. So as in cryptography, we assume that all communication parties can trade secret keys through a secure channel.
- Some algorithms additionally require the **knowledge of the original cover** (or some other information not derivable from the stego-object) in the decoding phase. Such systems are of limited interest because their use requires the transmission of the original cover, a problem strongly related to key- exchange in traditional cryptography. These algorithms can be seen as a special case of secret key steganographic systems in which K = C or K = C×K′ where K′ denotes an additional set of secret keys.

Figure (2.4): Secret Key Steganography.

3. Public Key Steganography

As in public key cryptography, public key steganography does not rely on the exchange of a secret key. **Public key steganography systems require the use of two keys, one private and one public key**; the public key is stored in a public database. Whereas the public key is used in the embedding process, the secret key is used to reconstruct the secret message.

In this protocol, Alice first generates a random public/private key pair foruse with any public-key cryptosystem. Then she embeds the public key in a channel known to and viewable by Bob (and hence also Wendy). Neither Wendy nor Bob can determine whether the channel contains more than random bits. However, Bob suspects that the stego-object sent by Alice contains Alice's public key and tries to extract it. He uses the received public key to embed a randomly chosen key k along with a short message of acknowledgment, both encrypted with Alice's public key, in a cover and sends it to Alice. Again, Wendy can try to extract the secret information sent by Bob, but will likely notice only random-looking ciphertext. Alice suspects the arrival of a message from Bob, extracts the secret information, and decrypts it with her private key. Now Alice and Bob share a stego-key k. This protocol is illustrated in Figure 4.5

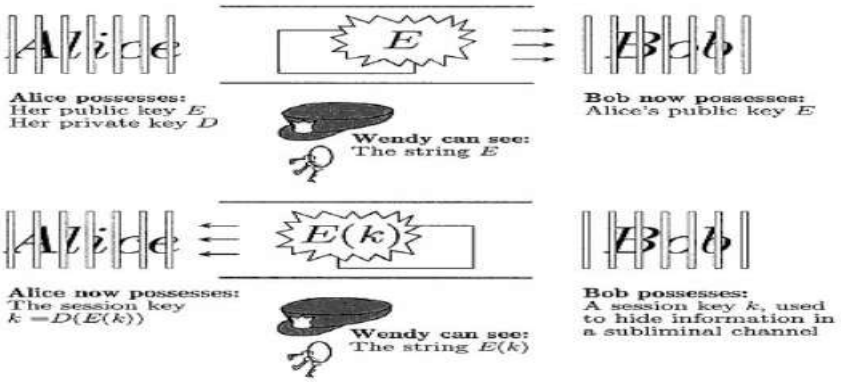


Figure (2.5): Illustration of a steganographic key-exchange protocol

3. Characterization of Steganography Systems

In Steganographic techniques, various features characterize the strength and weaknesses of the methods. The relative importance of each feature depends on the application. The following are some characteristics of Steganography Systems:

1. **Capacity** The idea of capacity in data hiding indicates the total number of bits hidden and successfully recovered by the Stego system.

2. **Robustness**: this refers to the ability of the embedded data to remain intact if the stego-system transforms, such as linear and non-linear filtering; addition of random noise; scaling, rotation, and loose compression
3. **Undetectable**: The embedded algorithm is undetectable if the image with the embedded message is consistent with a model of the source from which images are drawn. For example, if a Steganography method uses the noise component of digital images to embed a secret message, it should do so while not making statistical changes to the noise in the carrier. Undetectability is directly affected by the size of the secret message and the format of the content of the cover image.
4. **Invisibility (Perceptual Transparency)**: This concept is based on the properties of the human visual system or the human audio system. The embedded information is imperceptible if an average human subject is unable to distinguish between carriers that do contain hidden information and those that do not. It is important that the embedding occurs without significant degradation or loss of perceptual quality of the cover.
5. **Security**: It is said that the embedded algorithm is secure if the embedded information is not subject to removal after being discovered by the attacker and it depends on the total information about the embedded algorithm and secret key.

Chapter 3: Information Hiding in Noisy Data & Survey of Steganographic Techniques

1. Substitution Systems
    - Least Significant Bit Substitution
    - Pseudorandom Permutations
    - Image Downgrading and Covert Channels
    - Palette-Based Images
    - Information Hiding in Binary Images
    - Unused or Reserved Space in Computer Systems
2. Transform Domain Techniques
3. Spread Spectrum Techniques
4. Statistical Methods
5. Distortion techniques
6. Cover generation methods

– **Active and Malicious Attackers**

Information Hiding in Noisy Data & Survey of Steganographic Techniques

1. Introduction

Steganography utilizes the existence of **redundant information** in a communication process. **Images or digital sounds naturally contain such redundancies in the form of a noise component**.

In this section, we will assume without loss of generality that the cover **c** can be represented by a sequence of **binary digits**. In the case of a digital sound, this sequence is just the sequence of samples over time; in the case of a digital image, a sequence can be obtained by vectorizing the image (i.e., by lining up the grayscale or color values in a left-to-right and top-to-bottom order).

Many different steganographic methods have been proposed; most of them can be seen as substitution systems. Such methods try to substitute redundant parts of a signal with a secret message , but their main disadvantage is the relative weakness against cover modifications

There are several approaches in classifying steganographic systems. One could categorize them according to the **type of covers used for secret communication**. A classification according to the **cover modifications applied in the embedding process is another possibility**. We want to follow the second approach and group steganographic methods in six categories, although in some cases an exact classification is not possible:

1. **Substitution Systems:** substitute redundant parts of a cover with a secret message;
2. **Transform Domain Techniques:** embed secret information in a transform space of the signal (e.g., in the frequency domain);
3. **Spread Spectrum Techniques**: adopt ideas from spread spectrum

communication

4. **Statistical Methods** : encode information by changing several statistical properties of a cover and use hypothesis testing in the extraction process;
5. **Distortion techniques:** store information by signal distortion and measure the deviation from the original cover in the decoding step.
6. **Cover generation methods** encode information in the way a cover for secret communication is created.

In the following sections these six categories will be discussed

Substitution Systems

1. Substitution Systems

A number of methods exist for hiding information in various media. These methods range from LSB coding. 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 a passive attacker

➢ Least Significant Bit Substitution

Least significant bit (LSB) insertion is a common and simple method to embed data in a cover file. In this approach the LSB of a byte is restored with an M's bit (message bits) . LSB is common in steganography and are relatively easy to apply in image and audio. A surprising amount of information can be hidden with little, if any, perceptible impact to the carriers files . This technique operate well for image steganography.

Sample tools used in this group include StegoDos , S-Tools , Mandelsteg , EzStego , Hide and Seek , Hide4PGP , White Noise Storm , and Steganos. The image formats typically used in such steganography methods are lossless and the data can be directly manipulated and recovered. Some of these programs apply compression and encryption in addition to steganography services. These services provide better security of the hidden data.

Let l(c) be the number of elements in the sequence, **m** the secret message, and **l(m)** be its length in bits.

The general principle underlying most steganographic methods is to place the secret message in the **noise component** of a signal. If it is possible to code the information in such a way that it is indistinguishable from true random noise, an

attacker has no chance of detecting the secret communication.

The simplest way of hiding information in a sequence of binary numbers, and it is must consider the following issues :

- Replacing the least significant bit (LSB) of every element with one bit of the secret message **m**.
- In floating point arithmetic, the least significant bit of the mantissa العشري can be used instead.
- The size of the hidden message is much less than the number of bits available to hide the information **(l(m) << l(c))** the rest of the LSB can be left unchanged.
- The flipping the LSB of a byte (or a word) only means the addition or subtraction of a small quantity, the sender assumes that the difference will lie within the noise range and that it will therefore not be generally noticed.

The embedding process consists of choosing a subset *{j1, . . . , jl(m)}* of cover-elements and performing the substitution operation **cji** ≒ **mi** on them, which exchanges the LSB of **cji** by **mi (mi can either be 1 or 0).** One could also imagine a substitution operation which changes more than one bit of the cover, for instance by storing two message bits in the two least significant bits of one cover- element. In the extraction process, the LSB of the selected cover-elements are extracted and lined up to reconstruct the secret message. This basic scheme is presented in Algorithms 3.1 and 3.2. **One problem remains to be solved: in which way should the cji be chosen?**

Algorithm 3.1 Embedding process: least significant bit substitution

```
for i = 1,...,l(c) do
  s_i ← c_i
end for
for i = 1 ...,l(m) do
  compute index j_i where to store ith message bit
  s_ji ← c_ji ⇋ m_i
end for
```

Algorithm 3.2 Extraction process: least significant bit substitution

```
for i = 1,...,l(M) do
  compute index j_i where the ith message bit is stored
  m_i ← LSB(c_ij)
end for
```

In order to be able to decode the secret message, the receiver must have access to the sequence of element indices used in the embedding process. In the simplest case, the sender uses all cover-elements for information transfer, starting at the first element. Since the secret message will normally have less bits than l(c), the embedding process will be finished long before the end of the cover

The disadvantages of LSB technique :

1. Does not provide a high level of security.
2. An attacker can simply try to "decode" the cover, just as if he were the receiver.
3. In addition, the algorithm changes the statistical properties of the cover significantly, even if the message consists of truly random bits.

This technique can be improved by :

1. **I**nstead of using every cover element for information transfer, it is **possible to select only some elements in a rather random manner according to a secret key** and leave the others unchanged.
2. The above selection can be done using a pseudorandom number generator;

report a system in which the output of the random number generator is used to spread the sequence of message bits over the cover by determining the number of cover elements that are left unchanged between two elements used for information transfer.

Thus , a more sophisticated approach is the use of a pseudorandom number generator to spread the secret message over the cover in a rather random manner; a popular approach is the random interval method . If both communication partners share a stego-key k usable as a seed for a random number generator, they can create a random sequence k1, ..., kl(m) and use the elements with indices for information transfer.

$$j_1 = k_1$$
$$j_i = j_{i-1} + k_i, \quad i \geq 2$$

Thus, the distance between two embedded bits is determined **pseudorandomly.** Since the receiver has access to the seed **k** and knowledge of the pseudorandom number generator, he can reconstruct **ki** and therefore the entire sequence of element indices ji . This technique—which is especially efficient in the case of stream covers

Algorithm 3.3 Embedding process: random interval method

```
for i = 1...,l(c) do
  s_i ← c_i
end for
generate random sequence k_i using seed k
n ← k_1
for i = 1,...,l(m) do
  s_n ← c_n ⇋ m_i
  n ← n + k_i
end for
```

Algorithm 3.4 Extraction process: random interval method

```
generate random sequence k_i using seed k
n ← k_1
for i = 1,...,l(m) do
  m_i ← LSB(c_n)
  n ← n + k_i
end for
```

- **Pseudorandom Permutations** التبادل الشبه عشوائي

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

In a first attempt Alice could create (using a pseudorandom number generator) a sequence j1, . . . ,jl(m) of element indices and store the kth message bit in the element with index jk.

Note that one index could appear more than once in the sequence, since we have not restricted the output of the pseudorandom number generator in any way. We call such a case "collision".

If a collision occurs, Alice will possibly try to insert more than one message bit into one cover-element, thereby corrupting some of them.

If the message is quite short compared with the number of cover- elements, she hopes that the probability of collisions is negligible and that corrupted bits could be reconstructed using an error-correcting code. This is how- ever, only the case for quite short secret messages. The probability p of at least one collision can be estimated1 by (provided that l(m) << l(c)):

$$p \approx 1 - \exp\left(-\frac{\ell(m)[\ell(m)-1]}{2\ell(c)}\right)$$

For constant l(c), p converges rapidly to 1 as l(m) increases. If, for example, a digital image with 600 × 600 pixels is used as cover and about 200 pixels are selected in the embedding process, p is approximately 5%. On the other hand, if 600 pixels are used for information transfer, p increases to about 40%. We can conclude that only for very short messages the probability of collisions is negligible; if the message size increases, collisions must definitely be taken into account.

- **Image Downgrading and Covert Channels**

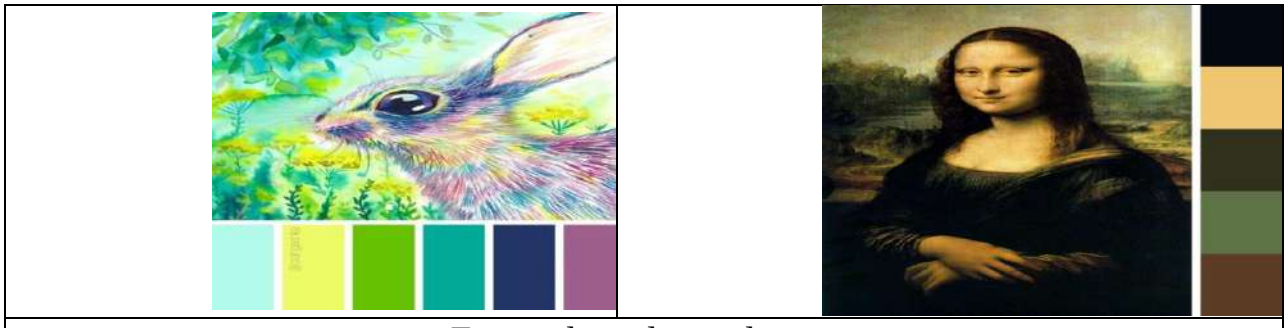
Image downgrading, could be used to exchange images covertly, it's a special case of a substitution system in which images act both as secret messages and covers.

Given a cover-image and a secret image of equal dimensions, the sender exchanges the four least significant bits of the cover's gray scale (or color) 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 secret image. While the degradation of the cover is not visually noticeable in many cases, 4 bits are sufficient to transmit a rough approximation of the secret image.

➢ Palette-Based Images

A color palette, in the digital world, refers to the full range of colors that can be displayed on a device screen or other interface or in some cases, a collection of colors and tools for use in paint and illustration programs.

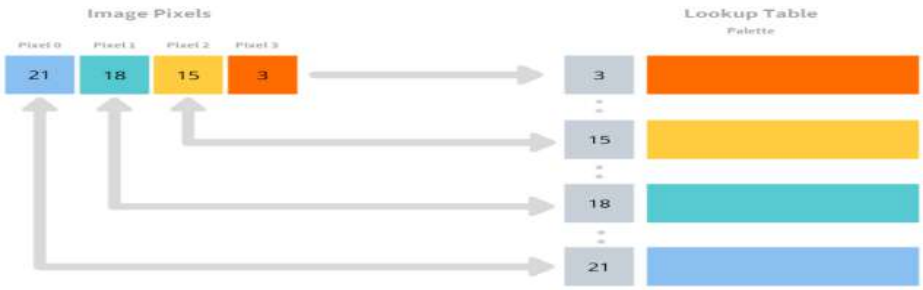
The new technique embeds one message bit into one pixel (its pointer to the palette). The pixels for message embedding are chosen randomly using a pseudo-random number generator seeded with a secret key. For each pixel at which one message bit is to be embedded, the palette is searched for closest colors.



Example color palettes

In a palette-based image only a subset of colors from a specific color space can be used to colorize the image.

Every palette-based image format consists of two parts: a palette specifying N colors as a list of indexed pairs (i, ci), assigning a color vector ci to every index i, and the actual image data which assign a palette index to every pixel rather than the color value itself. If only a small number of color values are used throughout the image, this approach greatly reduces the file size. Two of the most popular formats are the graphics interchange format (GIF) and the BMP bitmap format. However, due to the availability of sophisticated compression techniques, their use declines.



However, all methods which use the order of a palette to store information, are not robust, since an attacker can simply sort the entries in a different way and destroy the secret message (he thereby does not even modify the picture visibly).

Alternatively, information can be encoded in the image data. Since neighboring palette color values need not be perceptually similar, the approach of simply changing the LSB of some image data fails. Some steganographic applications (e.g., the program EzStego) therefore sort the palette so that neighboring colors are perceptually similar before they start the embedding process.

- **Information Hiding in Binary Images**

Binary images—like digitized fax data—contain redundancies in the way black and white pixels are distributed. Although the implementation of a simple substitution scheme is possible (e.g., certain pixels could be set to black or white depending on a specific message bit), these systems are highly susceptible to transmission errors and are therefore not robust.

One information hiding scheme which uses the number of black pixels in a specific image region to encode secret information was presented by Zhao and Koch [25], the process of hiding as follows :

A binary image is divided into rectangular image blocks Bi; let P0(Bi) be the percentage of black pixels in the image block Bi and P1(Bi) the percentage of white pixels, respectively.

Basically, one block embeds a 1, if P1 (Bi) > 50% and a 0, if P0 (Bi) > 50%. In the

embedding process the color of some pixels is changed so that the desired relation holds.

In order to make the entire system robust to transmission errors and other image modifications, we have to adapt the embedding process. If it is possible that some pixels change color during the transmission process, it could be the case that for instance P1(Bi) drops from 50.6% to 49.5%, thereby destroying the embedded information. Therefore two threshold values R1 > 50% and R0 < 50% and a robustness parameter λ , which specifies the percentage of pixels which can change color during transmission, are introduced.

The sender assures during the embedding process that either

$P1(Bi) \in [R1, R1 + \lambda]$

or $P0(Bi) \in [R0 - \lambda, R0]$ instead of P1(Bi) > 50% and P0(Bi) < 50%.

If too many pixels must be changed in order to achieve that goal, the block is marked as "invalid": P1(Bi) is modified to fulfill one of the two conditions

$$P_1(B_i) < R_0(B_i) - 3\lambda$$
$$P_1(B_i) > R_1(B_i) + 3\lambda$$

and another block is pseudorandomly chosen for bit i. In the decoding process, invalid blocks are skipped. Otherwise, the information is decoded according to P1(Bi).

The embedding and extraction algorithms are outlined in Algorithms 1 and 2.

```
For i = 1,...,l(M) do
  do forever
    Pseudorandomly select a new image block B_j
    /* Test, if block B_i is valid */
    If P_1(B_j) > R_1 + 3λ or P_1(B_j) < R_0 - 3λ then continue
    If (c_i = 1 and P_1(B_j) < R_0) or (c_i = 0 and P_1(B_j) > R_1) then
       Mark block B_j as unusable, i.e. modify block so that
        Either P_1(B_j) < R_0 - 3λ or P_1(B_j) > R_1 + 3λ
       Continue
    end if
    break
  End do
  /* embed secret message bit in B_j */
  If c_i = 1 then
     Modify B_j so that P_1(B_j) ≥ R_1 and P_1(B_j) ≤ R_1 + λ
  Else
     Modify B_j so that P_0(B_j) ≤ R_0 and P_0(B_j) ≥ R_0 - λ
  End if
end for
```

Algorithm 1 Zhao and Koch's algorithm for data embedding in binary images

```
For i = 1,...,l(M) do
  do forever
    Pseudorandomly select image block B_j
    If P_1(B_j) > R_1 + 3λ or P_1(B_j) < R_0 - 3λ then continue
    break
  End do
  If P_1(B_j) > 50% then
    m_i ← 1
  Else
    m_i ← 0
  End if
end for
```

Algorithm 2 Extraction process (Zhao and Koch)

➢ ***Unused or Reserved Space in Computer Systems***

Taking advantage of unused or reserved space to hold covert information provides a means of hiding information without perceptually degrading the carrier. For example:

- The way operating systems store files typically results in unused space that appears to be allocated to a file. This "extra" space can be used to hide information without showing up in the directory
- Another method of hiding information in file systems is to create a hidden partition. These partitions are not seen if the system is started normally.
- Protocols in the OSI network model have characteristics that can be used to hide information. TCP/IP packets used to transport information across the Internet have unused space in the packet headers.

Transform Domain Techniques

2. **Transform Domain Techniques :** Transform domain methods hide messages in significant areas of the cover image which makes them more robust to attacks, such as compression, cropping, and some image processing, than the LSB approach. However, while they are more robust to various kinds of signal processing, they remain imperceptible to the human sensory system. Many transform domain variations exist.
   - The advantage of performing data hiding in the frequency domain instead of in the spatial domain is that the hiding is done on the transformed coefficients and not directly onto the pixels of the cover image. Since the hiding is done on the coefficients of the transformed image the security of the secret image is higher.
   - One method is to use the Discrete Cosine Transformation (DCT) as a vehicle to embed information in images; another would be the use of Wavelet Transforms. Transformations can be applied over the entire image, to blocks throughout the image, or other variations. However, a trade-off exists between the amount of information added to the image and the robustness obtained.

     Before we describe transform domain steganographic methods, we will briefly review the cosine transforms which can be used to map a signal into the frequency domain.
   - The Discrete Cosine Transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain. The cosine transform, uses only cosine functions and not sine functions. Assuming an N×N image, the discrete cosine transform equation

is given by:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} I(r,c)\cos\left[\frac{(2r+1)u\pi}{2N}\right]\cos\left[\frac{(2c+1)v\pi}{2N}\right]$$

$$\alpha(u),\alpha(v) = \begin{cases}\sqrt{\frac{1}{N}} & \text{for } u,v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u,v = 1,2,\ldots,N-1\end{cases}$$

Because this transform uses only the cosine function, it can be calculated using on real arithmetic (not complex).

The inverse cosine transform is given by:

$$C^{-1}[C(u,v)] = I(r,c) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2r+1)u\pi}{2N}\right]\cos\left[\frac{(2c+1)v\pi}{2N}\right]$$

The popular way of doing steganography in the DCT domain is to modulate the relative size of two or more DCT coefficients within one image block. The basic algorithm is described as :

Encoding process,

- The sender splits the cover-image in 8×8 pixel blocks; each block encodes exactly one secret message bit. The embedding process starts with selecting a pseudorandom block bi which will be used to code the ith message bit.

  Let Bi = D{bi} be the DCT-transformed image block.

- Before the communication starts, both sender and receiver have to **agree on the location of two DCT coefficients**, which will be used in the embedding process;
- let us denote these two indices by (u1, v1) and (u2, v2). The two coefficients should correspond to cosine functions **with middle frequencies**; this ensures that :
-

- The information is stored in significant parts of the signal (hence the embedded information will not be completely damaged by JPEG compression).
- Furthermore, we can assume that the embedding process will not degenerate the cover heavily, because it is widely believed that DCT coefficients of middle frequencies have similar magnitudes.

– Since the constructed system should be robust against JPEG compression, we choose the DCT coefficients in such a way that the quantization values associated with them in the JPEG compression algorithm are equal. According to Table 3.1 the coefficients (4,1) and (3,2) or (1,2) and (3,0) are good candidates.

Table (1) Quantization values used in the JPEG compression scheme

(u,v)	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	14	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

One block encodes a "1," if Bi(u1, v1) > Bi(u2, v2), otherwise a "0." In the encoding step, the two coefficients are swapped if their relative size does not match with the bit to be encoded. Since the JPEG compression can (in the quantiza tion step) affect the relative sizes of the coefficients, the algorithm ensures that |Bi(u1, v1) - Bi(u2, v2)| > x for some x > 0, by adding random values to both coefficients.

The higher x is, the more robust the algorithm will be against JPEG compression, however, at the expense of image quality.

The sender then performs an inverse DCT to map the coefficients back into the space domain. To decode the picture, all available blocks are DCT-transformed. By comparing the two coefficients of every block, the information can be restored. Embedding and extraction algorithms are outlined in Algorithms 1 and 2

Algorithm 1: DCT-Steg encoding process

For $i = 1,...,l(M)$ **do**
choose one cover-block b_i
$B_i = D\{b_i\}$
If $m_i = 0$ **then**
If $B_i(u_1, v_1) > B_i(u_2, v_2)$ **then**
swap $B_i(u_1, v_1)$ and $B_i(u_2, v_2)$
End if
Else
If $B_i(u_1, v_1) < B_i(u_2, v_2)$ **then**
swap $B_i(u_1, v_1)$ and $B_i(u_2, v_2)$
End if
End if
adjust both values so that $|B_i(u_1, v_1) - B_i(u_2, v_2)| > x$
$b'_i = D^{-1}\{B_i\}$
End for

If the constant x and the location of the used DCT coefficients are chosen properly, the embedding process will not degenerate the cover visibly. We can expect this method to be robust against JPEG compression, since in the quantization process both coefficients are divided by the same quantization values. Their relative size will therefore only be affected in the rounding step.

Perhaps the most important drawback of the system presented above is the fact that Algorithm (1) does not discard image blocks where the desired relation of the DCT coefficients cannot be enforced without severely damaging the image data contained

Algorithm 2 DCT-Steg decoding process

For $i = 1,...,l(M)$ **do**

mm mget cover-block b_i associated with bit i

$B_i = D\{b_i\}$

if $B_i(u_1, v_1) \leq B_i(u_2, v_2)$ **then**

$m_i = 0$

else

$m_i = 1$

End if

End for

للاطلاع ... ليس مطلوب لكن لتوضيح الموضوع السابق

12	10	8	10	12	10	8	11
11	12	10	8	10	12	10	8
8	11	12	10	8	10	12	10
10	8	11	12	10	8	10	12
12	10	8	11	12	10	8	10
10	12	10	8	11	12	10	8
8	10	12	10	8	11	12	10
10	8	10	12	10	8	11	12

(a) Original data

81	0	0	0	0	0	0	0
0	1.57	0.61	1.90	0.38	1.81	0.20	0.32
0	0.61	0.71	0.35	0	0.07	0	0.02
0	1.90	0.35	4.76	0.77	3.39	0.25	0.54
0	0.38	0	0.77	8.00	0.51	0	0.07
0	1.81	0.07	3.39	0.51	1.57	0.56	0.25
0	0.20	0	0.25	0	0.56	0.71	0.29
0	0.32	0.02	0.54	0.07	0.25	0.29	0.90

(b) DCT coefficients

81	0	0	0	0	0	0	0
0	2	1	2	0	2	0	0
0	1	1	0	0	0	0	0
0	2	0	5	1	3	0	1
0	0	0	1	8	1	0	0
0	2	0	3	1	2	1	0
0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1

(c) Quantized

12.29	10.26	7.92	9.93	11.51	9.94	8.18	10.97
10.90	12.06	10.07	7.68	10.30	11.64	10.17	8.18
7.83	11.39	12.19	9.62	8.28	10.10	11.64	9.94
10.15	7.74	11.16	11.96	9.90	8.28	10.30	11.51
12.21	10.08	8.15	11.38	11.96	9.62	7.68	9.93
10.09	12.10	9.30	8.15	11.16	12.19	10.07	7.92
7.87	9.50	12.10	10.08	7.74	11.39	12.06	10.26
9.66	7.87	10.09	12.21	10.15	7.83	10.90	12.29

(d) Reconstructed data (good)

The popular way of doing steganography in the DCT domain is to modulate the relative size of two or more DCT coefficients within one image block. The basic algorithm is described as :

- Splitting the image into 8×8 blocks and calculating the DCT of the block.
- Then two middle-frequency (so that they are not to altered by the quantization/compression which will take place in JPEG) are chosen and agreed upon by both send and receive parties.
- A block encodes a 1 if DCT (a,b) > DCT(c,d) otherwise it encodes a 0.
- In the encoding step the coefficients are swapped if their relative size does not match with the bit to be encoded.
- Since the JPEG compression can affect the relative size of the coefficients the algorithm ensures that abs(DCT(a,b) - DCT(c,d)) > x where x is a value which represents the tradeoff between image quality and robustness

Spread Spectrum (SS) Communication Technologies

3. <u>Spread Spectrum (SS) Communication Technologies</u>: تقنيات الاتصالات (SS) انتشار الطيف

Spread spectrum techniques as "means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery".

- Although the power of the signal to be transmitted can be large, the signal-to-noise ratio in every frequency band will be small. Even if parts of the signal could be removed in several frequency bands, enough information should be present in the other bands to recover the signal. Thus, SS makes it difficult to detect and/or remove a signal.
- This situation is very similar to a steganography system which tries to spread a secret message over a cover in order to make it impossible to perceive. Since spreaded signals tend to be difficult to remove, embedding methods based on SS should provide a considerable level of robustness.

In information hiding, two special variants of SS are generally used:

- Direct-sequence : The secret signal is spread by a constant called chip rate, modulated with a pseudorandom signal and added to the cover
- Frequency-hopping schemes: the frequency of the carrier signal is altered in a way that it hops rapidly from one frequency to the another.

The main advantage of using spread spectrum techniques in steganography is the relative robustness to image modifications. Since the encoded information is spread over a wide frequency band it is quite difficult to remove it completely without entirely destroying the cover

4. **Statistical Steganography :** utilize the existence of "1-bit" steganographic schemes, which embed one bit of information in a digital carrier. This is done by modifying the cover in such a way that some statistical characteristics change significantly if a "1" is transmitted. Otherwise the cover is left unchanged. So the receiver must be able to distinguish unmodified covers from modified ones.

   In order to construct a l(m)-bit stego-system from multiple "1-bit" stegosystems, a cover is divided into l(m) disjoint blocks B1, . . . , Bl(m). A secret bit, mi , is inserted into the ith block by placing a "1" into Bi if mi = 1. Otherwise, the block is not changed in the embedding process. The detection of a specific bit is done via a test function which distinguishes modified blocks from unmodified blocks

$$f(B_i) = \begin{cases} 1 & \text{block } B_i \text{ was modified in the embedding process} \\ 0 & \text{otherwise} \end{cases}$$

   The function *f* can be interpreted as a hypothesis-testing function; we test the nullhypothesis "block Bi was not modified" against the alternative hypothesis "block Bi was modified." Therefore, we call the whole class of such steganography systems statistical steganography. The receiver successively applies *f* to all cover-blocks Bi in order to restore every bit of the secret message.

5. **Distortion techniques** : store information by signal distortion and measure the deviation from the original cover in the decoding step. **In contrast to substitution systems, distortion techniques require the knowledge of the original cover in the decoding process**. Alice applies a sequence of modifications to a cover in order to get a stego-object; she chooses this

sequence of modifications in such a way that it corresponds to a specific secret message she wants to transmit. Bob measures the differences to the original cover in order to reconstruct the sequence of modifications applied by Alice, which corresponds to the secret message

In many applications, such systems are not useful:

- since the receiver must have access to the original covers.
- If Wendy also has access to them, she can easily detect the cover modifications and has evidence for a secret communication. If the embedding and extraction functions are public and do not depend on a stego-key, it is also possible for Wendy to reconstruct secret messages entirely.

Thus , its better to assume that original covers can be distributed through **a secure channel.**

6. **Cover Generation Techniques**: In contrast to all embedding methods presented above, where secret information is added to a specific cover by applying an embedding algorithm, some steganographic applications generate a digital object only for the purpose of being a cover for secret communication. That means A cover generation method actually creates a cover for the sole purpose of hiding information. **Spam Mimic is an excellent example of a cover generation method.**

Active and Malicious Attackers

During the design of a steganographic system, special attention has to be paid to the presence of attacke. The following are a summarization of these attacks :

- **Active attacks:** An Active attack can change a cover during the communication process. Active attacks involve some modification of the data

stream or the creation of false statements. Types of active attacks are as follows:

- Modification of messages
- Denial of Service
- Image processing techniques (like smoothing, filtering, and image transformations)
- Lossy compression

➢ **Passive attacks:** A Passive attack attempts to learn or make use of information from the system but does not affect system resources. Passive Attacks are like eavesdropping on or monitoring transmission. The goal of the opponent is to obtain information that is being transmitted. Types of Passive attacks are as follows:

- Eavesdropping.
- Spying.

There is another classification of attacks based on **information available to the attacker**, the following are these classifications:

1. **Stego-only attack**. Only the stego-object is available for analysis.
2. **Known cover attack**. The "original" cover-object and stego-object are both available.
3. **Known message attack**. At some point, the hidden message may become known to the attacker. Analyzing the stego-object for patterns that correspond to the hidden message may be beneficial for future attacks against that system. Even with the message, this may be very difficult and may even be considered equivalent to the stego-only attack.
4. **Chosen stego attack**. The steganography tool (algorithm) and stego-object are known.

5. **Chosen message attack**. The steganalyst generates a stego-object from some steganography tool or algorithm from a chosen message. The goal in this attack is to determine corresponding patterns in the stego-object that may point to the use of specific steganography tools or algorithms.
6. **Known stego attack**. The steganography algorithm (tool) is known and both the original and stego-objects are available.

Even given the "best" alternative for the attacker, the embedded message may still be difficult to extract. Sometimes the approach is not to attack the algorithm or images at all, but to attack the password used to encrypt or choose the bits to hide the message. **This "brute force" is successful against some tools,** but still requires significant processing time to achieve favorable results

Information Hiding in Written Text

1- Introduction

Unlike noisy data, the written text contains less redundant information which could be used for secret communication.

Text steganography is considered **exceedingly difficult** due to the inadequate random data in textual files compared with other digital media, such as audio, image, or video files.

Steganographic methods can try to encode the information in text as follows :

- directly in the text (and so exploit the natural redundancy of languages)
- or in the text format (e.g., by adjusting the interword or interline space).

Many ways have been proposed to store information directly in messages, but it may cause:

1- Infrequent typing or spelling errors could be introduced
2- Commas omitted Words replaced by synonyms (مرادفات)
3- Most of them are not serious options, as they degrade the text heavily.
4- Additionally, the embedding task requires the interaction of the user, it therefore cannot be automated.

Text steganography can be generally split into three classes, as depicted in Figure (4.1): **Format-based, Random and Statistical generation and linguistic**

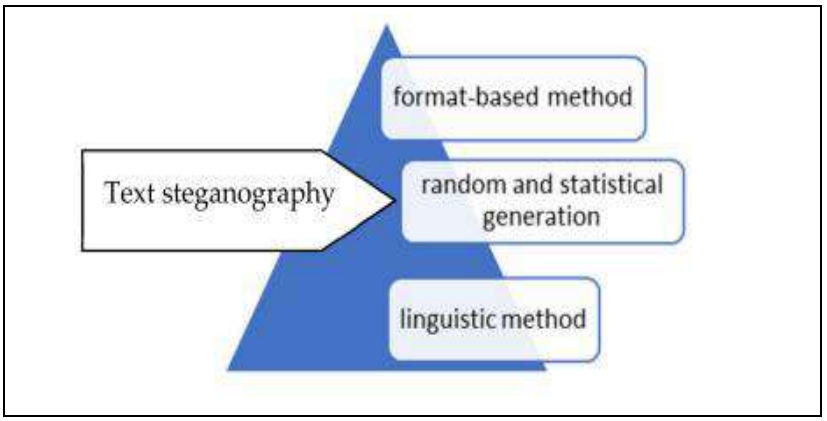
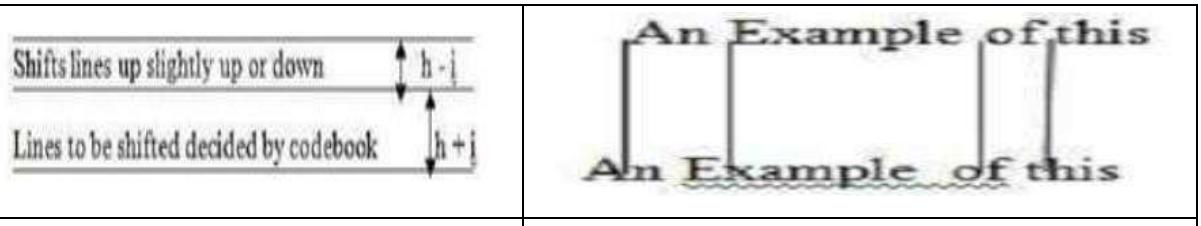


Figure (4.1): Categorization of Text Steganography

1- **Format-Based Method (Structural)** : used physical text formatting of text as a place in which to hide information. Generally, this method modifies existing text in order to hide the steganographic text. Insertion of spaces, misspellings distributed throughout the text, resizing the fonts are some of the many format-based methods being used in text steganography. However, those format-based methods managed to trick most of the human eyes but it cannot trick once computer systems have been used. The following some of Structural hiding example

- **Line Shift:** In this method, a secret message is hidden by vertically shifting the text lines to some degree. A line marked has two unmarked control lines one on either side of it for detecting the direction of movement of the marked line. To hide bit 0, a line is shifted up, and to hide bit 1, the line is shifted down. Determination of whether the line has been shifted up or down is done by measuring the distance of the centroid of the marked line and its control lines . If the text is retyped or if a Character Recognition Program (OCR) is used, the hidden information would get destroyed. Also, the distances can be observed by using special instruments of distance assessment.
    - **Word Shift:** In this method, the secret message is hidden by shifting the words horizontally, i.e. left or right to represent bit 0 or 1 respectively. Words shift are detected using the correlation method that treats a profile as a waveform and decides whether it originated from a waveform whose middle block has been shifted left or right. This method can be identified less, because change of distance between words to fill a line is quite common . But if someone knows the algorithm of distances, he can compare the stego text with the algorithm and obtain the hidden content by using the difference. Also, retyping or

using OCR programs destroys the hidden information.



Line Shift	**Word Shift**

Figure (4.2) : Examples of Format-Based Methods

These methods have certain flaws:

- If the stego file is opened with a word processor, misspellings and extra white spaces will get detected.
- Changed fonts sizes can arouse suspicion to a human reader.
- Additionally, if the original plaintext is available, comparing this plaintext with the suspected steganographic text would make manipulated parts of the text quite visible .

2- <u>Random and Statistical Generation</u>: The statistical characteristics of a language are obtained and then employed to generate cover text, to avoid comparison with known plaintext, the stenographer's often resorted to generating their cover texts. One method is concealing information in a random-looking sequence of characters. In another method, the statistical properties of word length and letter frequencies are used to create words that will appear to have the same statistical properties as actual words in the given language. As an examples :

- **Word Mapping** : This technique encrypts a secret message using genetic operator crossover and then embeds the resulting cipher text, taking two bits at a time, in a cover file by inserting blank spaces between words of even or odd length using a certain mapping technique. The embedding positions are saved in another file and transmitted to the

receiver along with the stego object.

- **MS Word Document** In this technique, text segments in a document are degenerated, mimicking to be the work of an author with inferior writing skills, with the secret message being embedded in the choice of degenerations which are then revised with changes being tracked. Data embedding is disguised such that the stego document appears to be the product of collaborative writing

3- <u>Linguistic Steganography</u>: Linguistic steganography specifically considers the **linguistic properties** of generated and modified text, and in many cases, uses linguistic structure as the space in which messages are hidden.

Linguistic method is a combination of syntax and semantics methods.

Syntactic steganalysis is to ensure that structures are syntactically correct.

In Semantic Method you can assign the value to synonyms and data can be encoded into actual words of text.

1- Syntactic Method: This technique uses punctuation marks such as full stop (.), comma (,), etc. to hide bits 0 and 1. But problem with this method is that it requires identification of correct places to insert punctuation marks . Therefore, care should be taken in using this method as readers can notice improper use of the punctuations . For instance, a method presented in [1], which utilizes the similarity of **La** word in the Arabic/Persian text, in their approach, the primary form of "La" ("لا ("is employed for hiding a bit "0," and specific form of the word "La" ("لا ("is employed for concealing a bit "1" . In practice, the syntacticbased techniques have low embedding capacity, high invisibility and high robustness against structural attacks. They are also vulnerable to visual attacks

2- Semantic method: This method uses the synonym of certain words

thereby hiding information in the text. The synonym substitution may represent a single or multiple bit combination for the secret information. However, this method may alter the meaning of the text .

Uses **context-free grammar (CFG)** to create cover texts and chooses the productions according to the secret message to be transmitted. Thus, the secret information is not embedded in the cover, the cover itself (actually the way it has been produced by a CFG) *is* the secret message.

If the grammar is unambiguous the receiver can extract the information by applying standard parsing techniques, methods which have been extensively studied in the construction of compilers. CFG creates a tree structure that can be used for concealing the bits where the left branch represents '0' and the right branch corresponds to '1'. This method has some drawbacks.

- First, a small grammar will lead to a lot of text repetition.
- Secondly, although the text is syntactically flawless, there is a lack of semantic structure. The result is a string of sentences that have no relation to one another.

uses a very sophisticated algorithm for computing line- and page breaks actually.

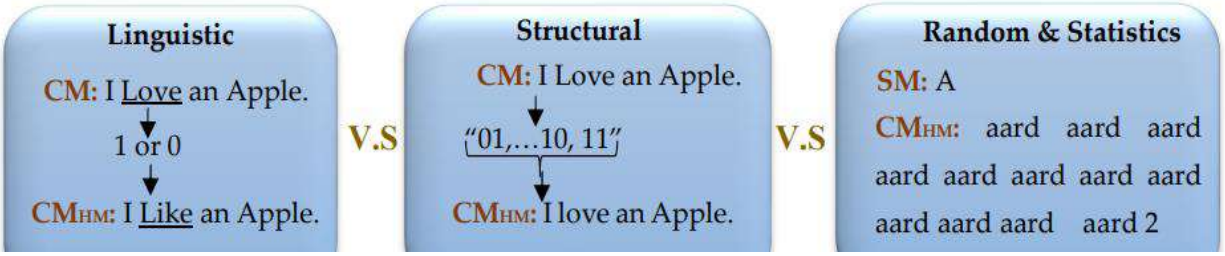


Figure (4.3). An empirical comparison between linguistic, Format (structural) , and random & statistics algorithms.

[1] Shirali-Shahreza, M. A New Persian/Arabic Text Steganography Using "La" Word. In Advances in Computer and Information Sciences and Engineering; Springer: Berlin/Heidelberg, Germany, 2008; pp. 339–342

Watermarking and Copyright Protection

1. Introduction

Digital Watermark (DW) is a pattern of bits that are permanently implanted مزروع in digital information (audio, pictures, video, and text) that can be recognized or retrieved later using computing operations to make claims about the data. The watermark is embedded in the host records in such a way that it is resistant to a broad range of attacks degrading the host document.

Watermarking is a technique with similarities to steganography. It has been around for centuries and is commonly used in money and stamps to assist in identifying counterfeiting التزييف. The idea behind watermarking is to create a transparent image on the paper to provide authenticity. Since mailing letters was far more expensive centuries back, it was common for people to use counterfeit stamps on their mail. For example, a translucent elephant watermark was used on stamps in India to prevent counterfeiting.

A popular application of watermarking is to give proof of ownership of digital data by embedding copyright statements. It is obvious that for this application the embedded information should be robust against manipulations that may attempt to remove it.

Both steganography and watermarking describe techniques that are used to imperceptibly convey نقل information by embedding it into the cover-data.

2. Introduction to Watermarking Techniques (*History)*

Paper watermarks appeared in the art of handmade papermaking nearly 700 years ago. The oldest watermarked paper found in archives dates back to 1292 and has its origin in the town of Fabriano in Italy which has played a major role in the evolution of the papermaking industry.

– At the end of the 13th century about 40 paper mills مصانع الورق were

sharing the paper market and producing paper with different format, quality, and price. They produced raw paper with very coarse surfaces not yet suitable for writing. This raw paper material was given to other artisans الحرفيين who smoothed the paper surface with the help of a hard stone, called calendar, to make it suitable for writing.

- The introduction of watermarks was the perfect method to eliminate any possibility of confusion (see Figure 5.1). Watermarks were mainly used to identify the mill producing the paper; a means of guaranteeing quality.

Figure (5.1) Monograms figuring TGE RG (the papermaker).

- After their invention, watermarks quickly spread in Italy and then over Europe and although initially used to indicate the paper brand or paper mill, they later served as indication for paper format, quality, and strength, and were also used as the basis for dating and authenticating paper.
- The analogy قياس between paper watermarks and digital watermarking is obvious: paper watermarks in bank notes or stamps inspired ألهمت the first use of the term "water mark" in the context of digital data. The first publications that focussed on watermarking of digital images were published by Tanaka et al. in 1990
- In 1995, the time was obviously right to pick up the topic, and it began to stimulate increasing research activities. Since 1995, digital watermarking has gained a lot of attention and has evolved very fast ,

and while there are a lot of topics open for further research.

3. Watermarking Terminology (*مصطلحات العلامات المائية*)

In this section, a familiar terminology is presented, and they are:

- **Visible watermarks**, as the name says, are visual patterns like logos which are inserted into or overlaid on images (or video), very similar to visible paper watermarks. Visible watermarks are mainly applied to images, for example, to visibly mark preview images available in image databases or on the Web in order to prevent commercial use of such images. So, watermarking is said to be visible when the content is visible to human eye whereas transparent watermarks are also known as **invisible watermarks,** in which the content is not visible to human eye. The applications of visible watermarks to video are of course also possible and under some circumstances one might even think of embedding an audible watermark into audio. Figure (5.2) shows example of visible and invisible watermarks

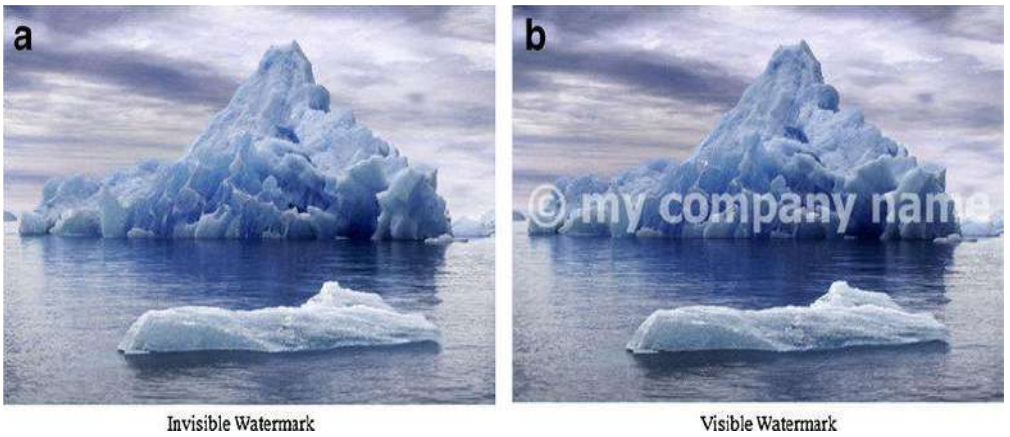


Figure (5.2) : Example of Visible and Invisible watermarks

- **Robust & Fragile Watermarking**: Watermarking as opposed to steganography has the additional notion of robustness against attacks. **Even if the existence of the hidden information is known it should be hard for an attacker to destroy the embedded watermark without knowledge of a key**.

- Thus, in **robust watermarking**, if the changes are made to the watermarked image or video which will not change the watermark, but in **fragile watermarking** technique watermarked content is changed which will change or destroy watermark. Fragile watermarks are watermarks that have only very limited robustness. They are applied to detect modifications of the watermarked data, rather than conveying un erasable information .
- **A practical effect of the robustness requirement is that watermarking methods can typically embed much less information into cover-data than steganographic methods.**
- Fingerprinting and labeling are terms that denote special applications of watermarking. They relate to watermarking applications where information such as the creator or recipient of digital data is embedded as watermarks. Fingerprinting means watermarking where the embedded information is either a unique code specifying the author or originator of the cover-data, or a unique code out of a series of codes specifying the recipient of the data.

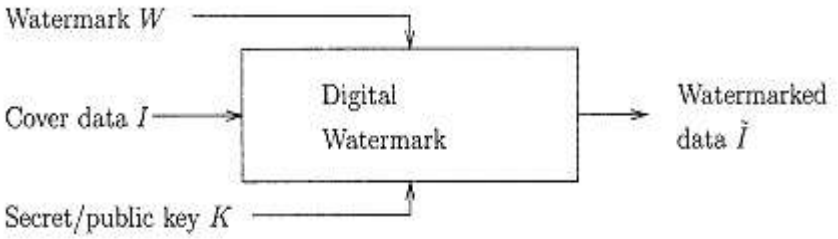


Figure (5.3) : Generic digital embedding watermarking scheme.

- **Bitstream watermarking:** is sometimes used for watermarking of compressed data, for example compressed video. Bitstream watermarking is another way of server-side watermarking which can be done in two ways: by pre-encoding the video or post-encoding it. The post-encoding advantage is that the video only needs to be encoded once to have the watermark, when for pre-encoding it has to be twice. This doubles the

encoding costs and time.

- **The term embedded signatures** has been used instead of "watermarking" in early publications, but is usually not used anymore, since it potentially leads to confusion with cryptographic signatures. Cryptographic signatures serve for authentication purposes. They are used to detect any alteration of the signed data and to authenticate the sender. Watermarks, however, are only used for authentication in special applications, and are usually designed to *resist* alterations and modifications.

4. Basic Watermarking Principles

All watermarking methods share the same generic building blocks: a *watermark embedding system* and a *watermark recovery system* (also called watermark extraction or watermark decoder). For real-world robust watermarking systems, a few very **general properties**, shared by all proposed systems, can be identified. They are:

- **Imperceptibility:** The modifications caused by watermark embedding should be below the perceptible threshold, which means that some sort of perceptibility criterion should be used not only to design the watermark, but also quantify the distortion. As a consequence of the required imperceptibility, the individual samples (or pixels, voxels, features, etc.) that are used for watermark embedding are only modified by a small amount.
- **Redundancy:** To ensure robustness despite the small allowed changes, the watermark information is usually redundantly distributed over many samples (or pixels, voxels, features, etc.) of the cover-data, thus providing a global robustness which means that the watermark can usually be recovered from a small fraction of the watermarked data. Obviously watermark recovery is more robust if more of the watermarked data is available in the recovery process.
- **Keys:** In general, watermarking systems use one or more cryptographically secure keys to ensure security against manipulation and removal of the watermark. As soon as a watermark can be read by someone, the same person may easily destroy it because not only the embedding strategy, but also the locations of the watermark are known in this case.

  These principles apply to watermarking schemes for all kinds of data that can be watermarked, like audio, images, video, formatted text, 3D models, model animation parameters, and others.

5. Watermarking Systems

Three types of watermarking systems can be identified. Their difference is in the nature and combination of inputs and outputs:

- **Private watermarking** (also called non-blind watermarking) systems require at least the original data.
  - **Type I**: systems extract the watermark W from the possibly distorted data $\tilde{I}'$ and use the original data as a hint to find where the watermark could be in $\tilde{I}'$.
  - **Type II**: systems also require a copy of the embedded watermark for extraction and just yield a "yes" or "no" answer to the question: does $\tilde{I}'$ contain the watermark W? ($\tilde{I}' \times I \times K \times W \rightarrow \{0, 1\}$). It is expected that this kind of scheme will be more robust than the others since it conveys very little information and requires access to secret material.
- **Semiprivate watermarking** (or semi-blind watermarking) does not use the original data for detection ($\tilde{I}' \times K \times W \rightarrow \{0, 1\}$) but answers the same question. Potential applications of private and semiprivate watermarking are for evidence in court to prove ownership, copy control in applications such as digital versatile disc (DVD) where the disc reader needs to know whether it is allowed to play the content or not, and fingerprinting where the goal is to identify the original recipient of pirated copies.
- **Public watermarking** (also referred to as *blind* or *oblivious* watermarking) remains the most challenging problem since it requires neither the secret original I nor the embedded watermark W. Indeed, such systems really extract n bits of information (the watermark) from the marked data: $\tilde{I}' \times K \rightarrow W$.

6. Watermarking Applications

In this section some applications of watermarking is presented. For obvious reasons there is no "universal" watermarking method. Although watermarking

methods have to be robust in general, different levels of required robustness can be identified depending on the specific application-driven requirements.

1) Watermarking for Copyright Protection

Copyright protection is probably the **most prominent application** of watermarking today. The objective is to embed information about the source, and thus typically the copyright owner, of the data in order to prevent other parties from claiming the copyright on the data. Thus, the watermarks are used to resolve rightful ownership, and this application requires a very high level of robustness. The driving force for this application is the Web which contains millions of freely available images that the rightful owners want to protect. Additional issues besides robustness have to be considered. For example, the watermark must be unambiguous and still resolve rightful ownership if other parties embed additional watermarks. Hence, additional design requirements besides mere robustness apply.

2) Fingerprinting for Traitor Tracking (*بصمات الأصابع لتعقب الخائن***)**

There are other applications where the objective is to convey information about the legal recipient rather than the source of digital data, mainly in order to identify single distributed copies of the data. This is useful to monitor or trace back illegally produced copies of the data that may circulate, and is very similar to serial numbers of software products. This type of application is usually called "fingerprinting" and involves the embedding of a different watermark into each distributed copy. Also, for some fingerprinting applications it is required to extract the watermark easily and with a low complexity, for example, for World Wide Web applications where special Web crawlers search for pirated watermarked images. Watermarks for fingerprinting applications also require a high robustness against standard data processing as well as malicious attacks.

3) Watermarking for Copy Protection

A desirable feature in multimedia distribution systems is the existence of a copy protection mechanism that disallows unauthorized copying of the media. Copy protection is very difficult to achieve in open systems; in closed or proprietary systems, however, it is feasible. In such systems it is possible to use watermarks indicating the copy status of the data. An example is the DVD system where the data contains copy information embedded as a watermark. A compliant DVD player is not allowed to playback or copy data that carry a "copy never" watermark. Data that carry a "copy once" watermarks may be copied, but no further consecutive copies are allowed to be made from the copy.

4) Watermarking for Image Authentication

In authentication applications, the objective is to detect modifications of the data. This can be achieved with so- called "fragile watermarks" that have a low robustness to certain modifications like compression, but are impaired by other modifications [14, 15]. Furthermore, the robustness requirements may change depending on the data type and application. Nevertheless, among all possible watermarking applications, authentication watermarks require the lowest level of robustness by definition. It should be noted that new approaches have emerged in which data attributes, such as block average or edge characteristics, are embedded and check if the received image still has the same attributes. It is clear that such schemes may require a higher robustness if identification of the modified areas is of interest.

7. Requirements and Algorithmic Design Issues

Depending on the watermarking application and purpose, different requirements arise resulting in various design issues. The following are the requirements of watermarking design have to be taken into consideration when designing watermarking techniques:

1) **Imperceptibility:** Watermark imperceptibility is a common requirement and independent of the application purpose , its one the most important requirements is the perceptual transparency of the watermark, independent of the application and purpose of the watermarking system. Artifacts introduced through a watermarking process are not only annoying مزعج and undesirable, but may also reduce or destroy the commercial value of the watermarked data. It is therefore important to design watermarking methods which exploit effects of the human visual or auditory system in order to maximize the energy of the watermark under the constraint of not exceeding the perceptible threshold. Two problems are related to this issue

- The first one is the reliable assessment of the introduced distortion
- The second problem occurs when processing is applied to the watermarked data. For example, in image watermarking the visibility of the watermark may increase if the image is scaled.

2) **Robustness:**

The ultimate watermarking method should resist any kind of distortion introduced by standard or malicious data processing. No such perfect method has been proposed so far, and it is not clear yet whether an absolutely secure watermarking method exists at all. Thus, practical systems must implement **a compromise between robustness and the competing requirements** like invisibility and information rate. Depending

on the application purpose of the watermarking methods, the desired robustness therefore influences the design process. For example:

- In image watermarking, if we need a method that is resilient to JPEG compression with high compression factors, it is probably more efficient to employ a method working in a transform domain than to use a method that works in the spatial domain.
- Similarly, if the method should accommodate generalized geometrical transformations, that is rotation, nonuniform scaling, and shearing, an approach in the spatial domain is probably more suitable.

Looking at the distortion that the watermarked data is likely to undergo by either intentional or unintentional modifications, two groups of distortion can be distinguished.

- The first one contains distortions which can be considered as additive noise to the data
- Whereas the distortion in the second group are due to modifications of the spatial or temporal data geometry with the intent to introduce a mismatch between the watermark and the key used for embedding.

These two distortions or attacks are often referred to as ***destruction attacks*** and ***synchronization attacks***, respectively.

Depending on the application and watermarking requirements, the list of distortions and attacks to be considered includes, but is not limited to:

1. Signal enhancement (sharpening, contrast enhancement, color correction, gamma correction);
2. Additive and multiplicative noise (Gaussian and uniform);
3. Linear filtering (low pass- and high pass);
4. Nonlinear filtering (median filtering, morphological filtering);

5. Lossy compression (**images:** JPEG, **video:** H.261, H.263, MPEG-2, MPEG-4, **audio:** MPEG-2 audio, MP3, MPEG-4 audio);
6. Local and global affine transforms (translation, rotation, scaling, shearing);
7. Data reduction (cropping, clipping, histogram modification);
8. Data composition (logo insertion, scene composition);
9. Transcoding (H.263 → MPEG-2, GIF → JPEG);
10. D/A and A/D conversion (print-scan, analog TV transmission);
11. Multiple watermarking;
12. Statistical averaging;

3)Watermark Recovery with or without the Original Data

Watermarking methods using the original data set in the recovery process increased robustness .

- In many applications, such as data monitoring or tracking, access to the original data is not possible.
- In other applications, such as video watermarking applications, it may be impracticable to use the original data because of the large amount of data that would have to be processed.

While most early watermarking techniques require the original data for recovery, there is a clear tendency to devise techniques that do not require the original data set.

4) Watermark Extraction or Verification of Presence for a Given Watermark

As was said before, two types of watermarking schemes exist:

- Systems that embed a specific information or pattern and check the existence of the (known) information later on in the watermark recovery process. For example, copyright protection can be achieved with systems **verification** of the presence of a known watermark
- Systems that embed arbitrary information into the data. For example, used for image tracking on the Internet with intelligent agents where it might not

only be of interest to discover images, but also to classify them. The embedded watermark can be used as an image **identification** number or as a pointer to a database entry.

5) Watermark Security and Keys

In most applications, such as copyright protection, the secrecy of embedded information needs to be assured. This and related issues are often referred to as watermark security. Applications in which security is not an issue include image database indexing**. If secrecy is a requirement, a secret key has to be used for the embedding and extraction process.**

Two levels of secrecy can be identified.

- In the highest level of secrecy an unauthorized user can neither read nor decode an embedded watermark nor can he detect if a given set of data contains a watermark.
- The second level permits any user to detect if data is watermarked, but the embedded information cannot be read without having the secret key. Such schemes may, for example, be useful in copyright protection applications for images.

When designing a working overall copyright protection system, issues like secret key generation, distribution, and management (possibly by trusted third parties), as well as other system integration aspects have to be considered as well.

6) Resolving Rightful Ownership

In order to successfully resolve rightful ownership, it must be possible to determine who first watermarked a data set in case it contains multiple watermarks. This can be achieved by imposing design constraints, such as time-stamping (Time-stamping is one technique used to ascertain at what time a certain digital medium was created or signed).

Evaluation and Benchmarking قياس of Watermarking Systems

7.1 Introduction

Besides designing digital watermarking methods, an important issue addresses proper evaluation and benchmarking. This not only requires evaluation of the robustness, but also includes subjective or quantitative evaluation of the distortion introduced through the watermarking process.

In general, there is a trade-off between watermark robustness and watermark perceptibility. Hence, for fair benchmarking and performance evaluation one has to ensure that the methods under investigation are tested under comparable conditions

- **Performance Evaluation and Representation**

Independent of the application purpose type of data, the robustness of watermarks depends on the following aspects:

1) **Amount of embedded information.** This is an important parameter since it directly influences the watermark robustness. The **more** information one wants to embed, the **lower** the watermark robustness.
2) **Watermark embedding strength.** There is a trade-off between the watermark embedding strength (hence the watermark robustness) and watermark perceptibility. **Increased robustness** requires a stronger embedding, which in turn **increases perceptibility** of the watermark.
3) **Size and nature of data.** The size of the data has usually a direct impact on the robustness of the embedded watermark. For example, in image watermarking very small pictures do not have much commercial value; nevertheless, a marking software program needs to be able to recover a watermark from them.
4) **Secret information (e.g., key).** The key space, that is, the range of all possible values of the secret information, must be **large enough to make exhaustive search attacks impossible.**

Taking these parameters into account, we realize that for fair benchmarking and performance evaluation, watermarking methods need to be tested on different data sets.

Furthermore, in order to compute statistically valid results the methods have to be evaluated using many different keys and varying watermarks. The amount of embedded information is usually fixed and depends on the application. However, if watermarking methods are to be compared, it has to be assured that the amount of embedded information is the same for all methods under inspection.

As we have seen above, **there is a trade-off between the watermark perceptibility and the watermark robustness.**

7.2 Perceptibility of the Watermarks

Evaluating the perceptibility of the watermarks can be done either through **subjective tests or a quality metri**c.

- **Subjective Tests:** In subjective testing a group of people are asked to give their opinion about the quality of each image. In order to perform a subjective image quality testing, several international standards are proposed which provide reliable results. In subjective test, a testing protocol has to be followed, describing the testing and evaluation procedure. Such tests usually involve a two-step process:
- **In a first round**, the distorted data sets are rank ordered from best to worst.
- **In the second round**, the subject is asked to rate each data set, describing the perceptibility of the artifacts.

This rating can be based, for example, on the ITU-R Rec. 500 shows in Table (7.1)

Subjective tests are practical for final quality evaluation and testing, but are not very useful in a research and development environment.

Table 7.1: Example of **subjective tests** from 1 to 5 (ITU-R Rec. 500).

Rating	Impairment	Quality
5	Imperceptible	excellent
4	Perceptible, not annoying	good
3	Slightly annoying	fair
2	Annoying	poor
1	Very annoying	bad

The goal of objective IQA is to design mathematical models that are able to predict the quality of an image accurately and also automatically. An ideal objective IQA method should be able to mimic the quality predictions of an average human observer

The objective metrics are borrowed from digital signal processing and information theory and provide us with **equations** that can be used to measure the amount of distortion or error in the reconstructed image. Objective metrics are much more efficient and allow fair comparison between different methods as the results do not depend on subjective evaluations.

Table 7.2 shows the difference distortion metrics and Table 7.3 shows Correlation distortion metrics which are commonly used in image and video processing.

Nowadays, the most popular distortion measures in the field of image and video coding and compression are the Signal-to-Noise Ratio (SNR), and the Peak Signal-to-Noise Ratio (PSNR). They are usually measured in decibels (dB).

It is well known that these difference distortion metrics are not very well correlated with the human visual or auditory system.

Table 7.2 : Difference distortion metrics

Difference Distortion Metrics	**Equation**
Average absolute difference	$AD = \frac{1}{XY} \sum_{x,y} \lvert p_{x,y} - \tilde{p}_{x,y} \rvert$
Mean squared error	$MSE = \frac{1}{XY} \sum_{x,y} (p_{x,y} - \tilde{p}_{x,y})^2$
Signal-to-noise ratio	$SNR = \sum_{x,y} p_{x,y}^2 / \sum_{x,y} (p_{x,y} - \tilde{p}_{x,y})^2$
Peak signal-to-noise ratio	$PSNR = XY \max_{x,y} p_{x,y}^2 / \sum_{x,y} (p_{x,y} - \tilde{p}_{x,y})^2$

Table 7.3 : Correlation distortion metrics

Correlation Distortion Metrics	**Equation**
Correlation quality	$CQ = \sum_{x,y} p_{x,y} \tilde{p}_{x,y} / \sum_{x,y} p_{x,y}$
Histogram similarity	$HS = \sum_{c=0}^{255} \lvert f_I(c) - f_{\tilde{I}}(c) \rvert$ where $f_I(c)$ is the relative frequency of level *c in a 256-levels image*

Example: Compute MSE, SNR and PSNR for the following reference image P(x, y) and processed image Ṕ (x, y) ?

Reference image P(x, y)= $\begin{matrix} 3 & 2 & 1 \\ 1 & 2 & 1 \\ 3 & 2 & 2 \end{matrix}$

Processed image Ṕ (x, y) = $\begin{matrix} 3 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{matrix}$

MSE =

SNR=

PSNR =