University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Multi Media Security
أمنية الوسائط المتعددة

Prof. Dr. Abdulamir A. Karim
أ.د عبدالأمير عبدالله كريم

cs.uotechnology.edu.iq

Chapter One : Introduction to Multimedia Security

1. Introduction

In general , the information security has been classified into cryptography and information hiding. Traditionally, cryptography which is the process of converting ordinary plain message into unintelligible message has been used for a trusty and reliable transmission. Nevertheless, the encrypted data is a meaningless message which perhaps attracting the illegal attackers attentions. Hence, to overcome this drawback the steganography is invented in order to invisibly embedding secure data into a cover media without drawing any special suspicion from hackers.

Now a day, one of the most important information security algorithms is the information hiding. Information hiding contains two main techniques steganography and watermarking. The main usage of watermarking is for the copyright protection of the electronic products, while the steganography technique is a way of embedding secret message into digital media, so it can be considered as a secret data communication. In other words, the main purpose of steganography is to send data secretly through hiding the existence of that data in some other media. The media used to hide the secret data is called cover media, while the media which contains the hidden data is called stego media.

In the next paragraphs will introduce the main sub-disciplines of information hiding related to computer systems and give a brief history of this fascinating area of research.

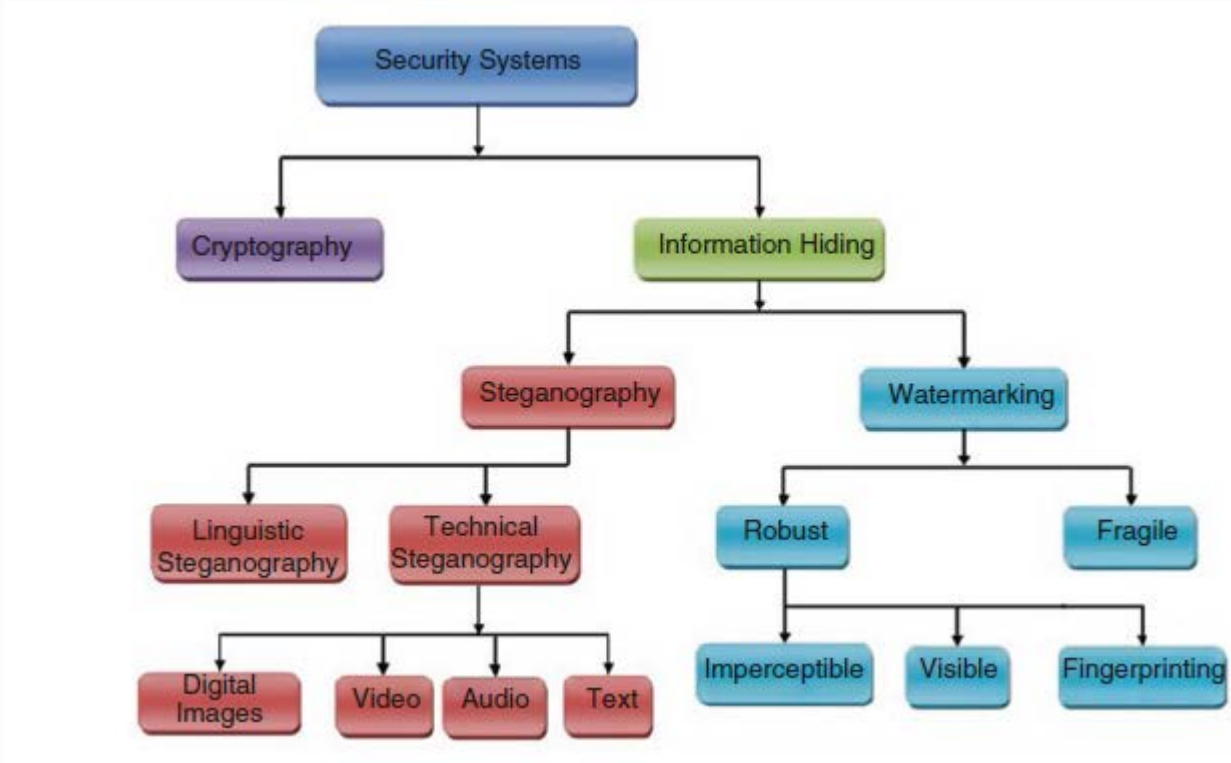


Fig. 1.1 Security system

2. Main Sub-disciplines of Information Hiding

Steganography *:* Now a day, 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 innocuous cover messages.

An important sub-discipline of information hiding is steganography. While cryptography is about "scrambling" a message so that if it is intercepted, it cannot be understood, steganography is about concealing their very existence. This modern adaptation of steganographia assumed from Greek, literally means "covered writing", and is usually interpreted to mean hiding information in other information. Examples include sending a message to a spy by ***marking certain letters in a newspaper using invisible ink, and adding sub-perceptible echo at certain places in an audio recording.***

Watermarking*:* As audio, video, and other media become available in digital form, the ease with which perfect copies can be made, may lead to large-scale unauthorized copying which might undermine 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, and the former to prosecute them.

As opposed to steganography, watermarking has the additional requirement of robustness against possible attacks. In this context, the term "robustness" is still not very clear; it mainly depends on the application, but a successful attack will simply try to make the mark undetectable. Robustness has strong implications in the overall design of a watermarking system and this is one of the reasons why we will treat steganography and digital watermarking separately in this course.

Watermarks do not always need to be hidden, as some systems use ***visible digital watermarks***, but most of the literature has focused on ***imperceptible*** (invisible, transparent, or inaudible, depending on the context) digital watermarks which have wider applications. Modern visible watermarks may be visual patterns (e.g., a company logo 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 the reader may have already noticed another fundamental **difference** between steganography and watermarking.

Steganography	watermark
1. Steganographic systems just hide secret information. 2. The external data (e.g. image, video,...) are not very important, they are just a carrier of the internal data (hidden message) which are the most important. 3. The "robustness" criteria are not important, since steganography is mainly concerned with detection of the hidden message. 4. Embed as much as possible data. 5. Steganographic communications are usually point-to-point (between sender and receiver)	1. The information hidden by a watermarking system is always associated to the digital object to be protected or to its owner (ownership protection). 2. The important data is the external data (e.g. image, video,...) , the internal data (watermark) are additional data for protecting the external data and to prove ownership . 3. Watermarking concerns potential removal by a pirate, the hidden message should be robust to attempts aimed to removing it. 4. Embed the data required to represent the watermark. 5. Watermarking communications techniques are usually one-to-many.

3. A Brief History of Information Hiding

In this section we do not intend to cover the whole history of information hiding, rather just give the important landmarks.

Steganography

The most famous examples of steganography go back to antiquity. In his *Histories*, Herodotus (c. 486-425 B.C.) tells how around 440 B.C. Histaeus shaved the head of his most trusted slave and tattooed it with a message which disappeared after the hair had re-grown. The purpose was to instigate a revolt against the Persians. Astonishingly, the method was still used by some German spies at the beginning of the 20th century. Invisible inks have been used extensively. Common sources for invisible inks are milk, vinegar, fruit juices, and urine; all of these darken when heated. Progress in chemistry helped to create more sophisticated combinations of ink and developer by the first World War, but the technology fell into disuse with the invention of "universal developers" which could determine which parts of a piece of paper had been wetted from the effects on the surfaces of the fibers. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information. This leads to the more familiar application-specific information hiding and marking technologies found in the world of secure printing. More recent innovations include special ultraviolet fluorescent inks used in printing traveler's checks. As the lamps used in photocopiers have a high ultra-violet content, it can be arranged that photocopies come out overprinted with "void" in large letters.

Linguistic Steganography

An improvement is made when the message is hidden at random locations in the cover-text. This idea is the core of many current steganographic systems. In a security protocol developed in ancient China, the sender and the receiver had copies of a paper mask with a number of holes cut at random locations. The sender would place his mask over a sheet of paper, write the secret message into the holes, remove the mask, and then

compose a cover message incorporating the code ideograms. The receiver could read the secret message at once by placing his mask over the resulting letter. In the early 16th century Cardan (1501–1576), an Italian mathematician, reinvented this method which is now known as the Cardan grille.

Cryptography

By the 16th and 17th centuries, there had arisen a large number of methods depended on novel means of encoding information (cryptography). The expanded code uses 40 tables, each of which contains 24 entries (one for each letter of the alphabet of that time) in four languages: Latin, German, Italian, and French. 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 simple correspondence letter. Another method, based on hiding messages by using music scores; in which, each note corresponds to a letter.

4. The Types of Media

There are many types of covers in which information are embedded. Some of which are public, others are not. Always, information hiding users discover new types of cover; therefore, types of covers cannot be enumerated. In every day we expect a new type of cover. Figure (1.2) illustrates the main types of file formats which may be utilized for information hiding.

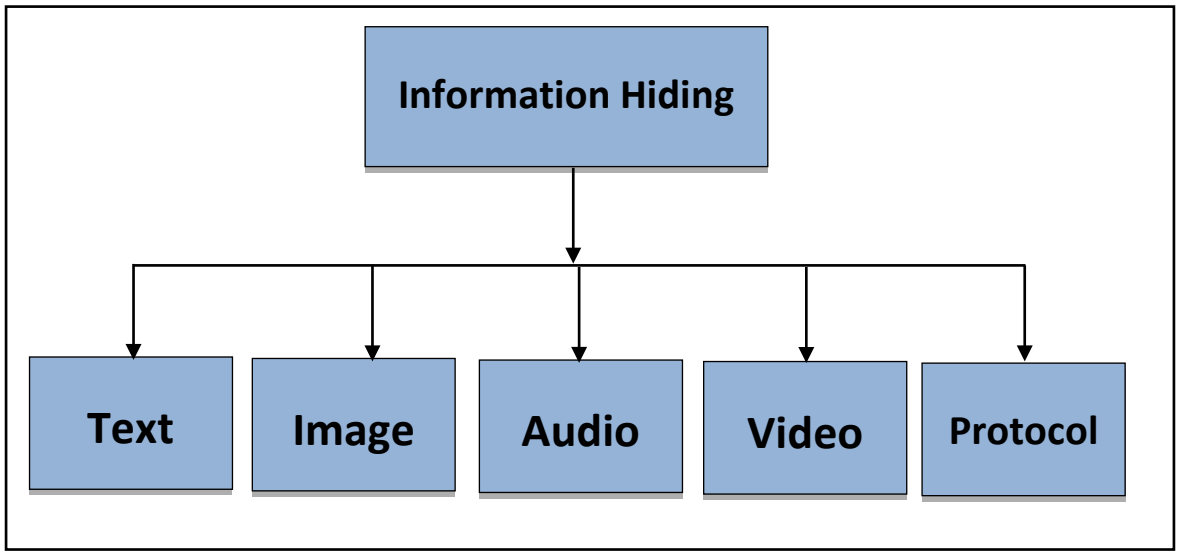


Fig. 1.2 The information hiding media.

5. Some Applications of Information Hiding

a. Unobtrusive communications are required by military and intelligence agencies: even if the content is encrypted, the detection of a signal on a modern battlefield may lead rapidly to an attack on the signaler. For this reason, military communications has to use techniques that make signals hard for the enemy to detect or jam.

b. The healthcare industry and especially medical imaging systems may benefit from information hiding techniques. As it is known that the Patient information (e.g. patient name, the date, and the physician) is separate from the medical image (e.g. MRI) itself and thus, they are sent separately, which leads to the possibility of losing the match between them. Thus, embedding the name of the patient in the image could be a useful safety measure.

   Another emerging technique related to the healthcare industry is hiding messages in DNA sequences.

c. Numerous data sources such as one's private banking information could be stored in a cover data. When it is time to uncover the hidden data in the cover message, it is possible to easily reveal the banking information and it will not be possible proving the existence of the banking secret information inside.

d. Copyright and ownership protection.

6. Security attack

It is any action that compromises the security of information owned by an organization. Security attacks are classified as either passive attacks or active attacks. A ***passive attack*** attempts to learn or make use of information from the system but does not affect system resources. An ***active attack*** attempts to alter system resources or affect their operation.

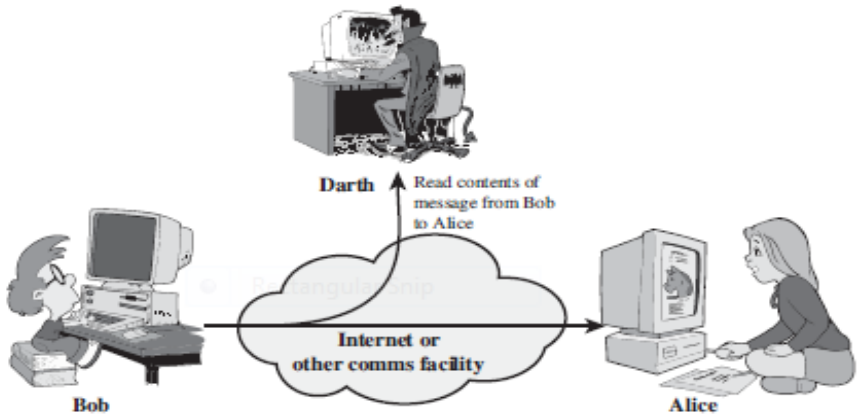
6.1 Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of transmissions. The **goal** of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

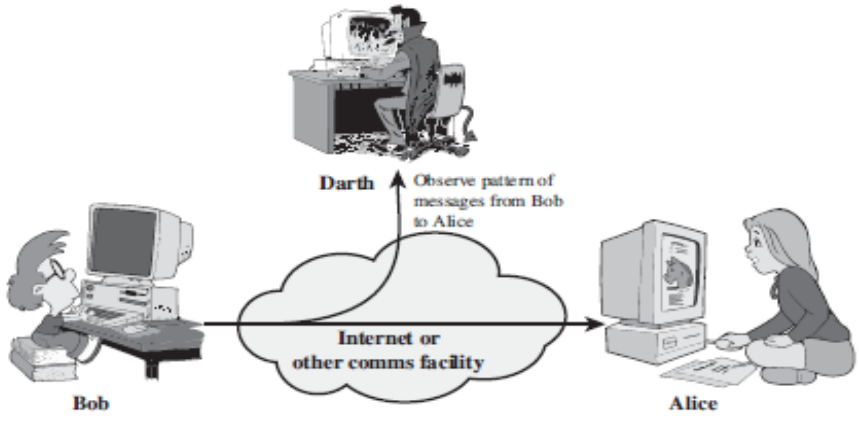
- The **release of message contents** is easily understood (Figure 1.3a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. An opponent would like to learn the contents of these transmissions.
- A second type of passive attack, **traffic analysis**, is subtler (Figure 1.3b). An opponent might be able to observe the pattern of these messages. The opponent

could determine the ***location*** and ***identity*** of communicating hosts and could observe the ***frequency*** and ***length*** of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are **very difficult to detect**, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to **prevent the success** of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on **prevention rather than detection**.



(a) Release of message contents



(b) Traffic analysis

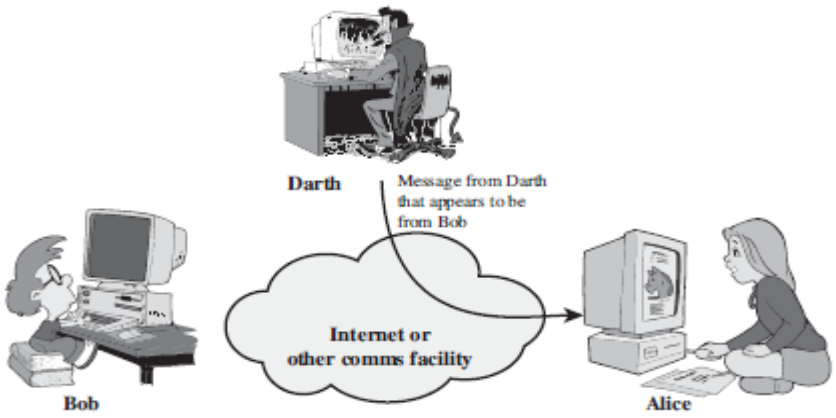
Figure 1.3 Passive Attacks

6.2 Active Attacks

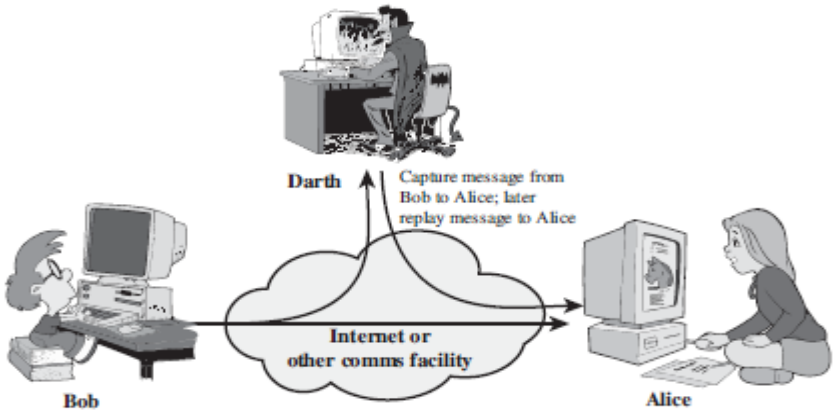
Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

- A **masquerade** takes place when one entity pretends to be a different entity (Figure 1.4a).
- **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (e.g. messages are delayed or reordered, to produce an unauthorized effect (Figure 1.4b).
- **Modification of messages** simply means that some portion of a legitimate message is altered (Figure 1.4c). For example, a message meaning "Allow John Smith to read confidential file accounts" is modified to mean "Allow Fred Brown to read confidential file accounts."
- The **denial of service** prevents or inhibits the normal use or management of communications facilities (Figure 1.4d). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

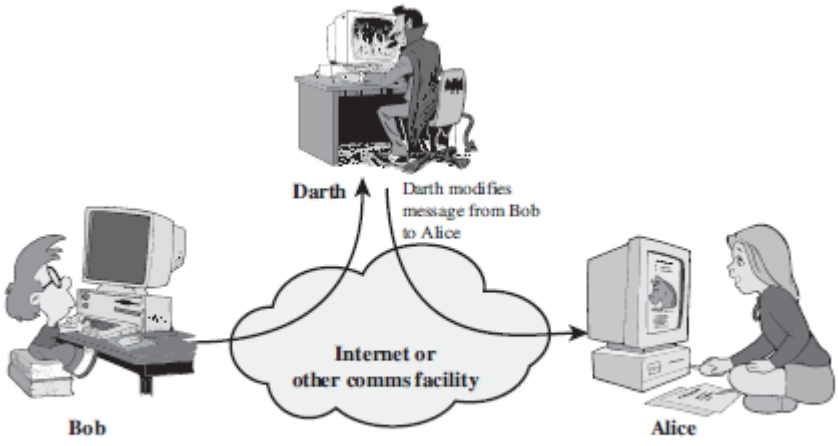
Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite **difficult to prevent active attacks** absolutely because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to **detect active attacks** and to **recover from any disruption or delays** caused by them. If the detection has a deterrent effect, it may also contribute to prevention.



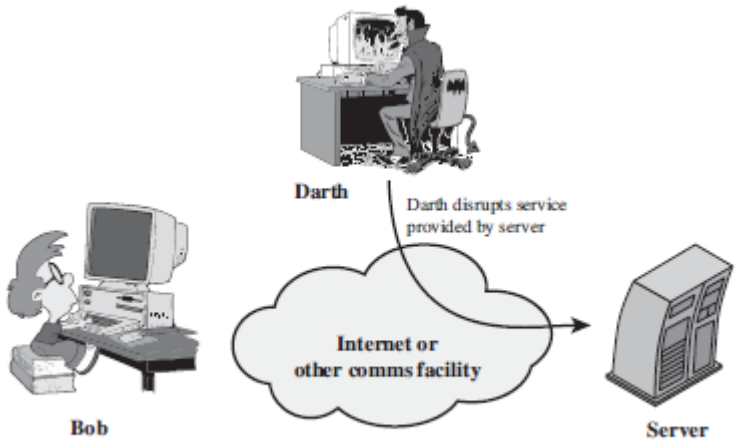
(a) Masquerade



(b) Replay



(c) Modification of messages



(d) Denial of service

Figure 1.4 Active attacks

Chapter Two : Cryptography

1. Introduction

We have less direct control over these secrets as they travel over the Internet or fly through the air on a wireless network. It is encryption that provides us with both the mechanism and confidence to store and transmit our most sensitive digital information. Encryption is a double-edged sword with serious consequences when used by criminals, terrorists and hackers.

Encryption is the conversion of data into a form, called cipher text, which cannot be easily understood by unauthorized people. Encryption starts with plain text. Plain text is the original, unencrypted message. The plain text message is in the clear and can be read by anyone. A cryptographic algorithm is then applied to the plain text, producing cipher text. Cipher text is basically a scrambled version of plain text that is unintelligible. The algorithm is the method used to encrypt the message. The key is data used to encrypt and decrypt the information. A password or passphrase is commonly used as the key.

2. Key Space

Key space is a metric that is often discussed when talking about the strength of a particular encryption scheme. The key space or key length has a ***direct impact*** on our ability to break the encryption, particularly with a brute force attack. A brute force attack tries to ***break*** the password by attempting every possible key combination until the right one is found. This is where this gets particularly troubling when you consider all the possible key permutations and how long it would take to "guess" the password. An encryption scheme with a 128-bit key would have roughly 340,282,366,920,938,000,000,000,000,000,000,000,000 possible key combinations. How long would that take a computer to guess the password? Crunching some rough numbers will give us an idea. Using one computer, guessing 500,000 passwords per second would break that key in about 21,580,566,141,612,000,000,000,000,000 years. Let's crunk up the number of computers guessing passwords to 1000. That gets us to a much more "manageable" wait time of only 21,580, 566,141,612,000,000,000,000 years. Remember these numbers represent rough estimates; the truth is that they can be much higher depending on the algorithm used. Complex encryption schemes such as

Advanced Encryption Standard (AES) can radically drop the number of attempts per second to only a few hundred.

3. Cryptography

Cryptography scrambles the contents of a file or message and makes it unreadable to all but its intended recipient. The word cryptography comes from Greek words "krypto", which means “hidden,” and "graphein", which means “to write”. Although cryptography’s importance has become more widely acknowledged in recent years, its roots are traced back 2,500 years ago. Around 400 B.C., the Spartans used an innovative method to encrypt, or hide, the meaning of military communication from unauthorized eyes. They would wrap a strip of parchment around a stick in a spiral, similar to a barber’s pole. The scribe would write the message on the parchment and then unwind it from the stick. With the parchment stretched out, the message was unintelligible. In fact, the only way to read the message, or decrypt it, was to wrap the parchment around another stick of the same diameter and equal, or greater, length. The “***secrets***” to reading the message were the dimensions of the stick and the knowledge of how to wrap the parchment. Anyone who possessed these two components could read the secret message.

Cryptography: The science of hiding the true contents of a message from unintended recipients.

Encrypt: To obscure the meaning of a message to make it unreadable.

Decrypt: To translate an encrypted message back into the original unencrypted message.

Cipher: An algorithm for encrypting and decrypting.

4. Substitution and Transposition cipher

In this section we examine a sampling of what might be called classical encryption techniques. The ***two basic building blocks of all encryption techniques are substitution and transposition.***

4.1 Substitution cipher

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.

Substitution cipher: A cipher that substitutes each character in the original message with an alternate character to create the encrypted message.

Caesar Cipher

The earliest known, and the simplest, use of a substitution cipher was Caesar cipher, which was invented by Roman Emperor Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing ***three places (key)*** further down the alphabet. For example:

plain: `meet me after the toga party`

cipher: `PHHW PH DIWHU WKH WRJD SDUWB`

The Caesar cipher uses a single key value. The key value tells how many positions to add to the plaintext character to encrypt and the number to subtract from the cipher text character to decrypt.

Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

plain: `a b c d e f g h i j k l m n o p q r s t u v w x y z`

cipher: `D E F G H I J K L M N O P Q R S T U V W X Y Z A B C`

Let us assign a numerical equivalent to each letter:

a	b	c	d	e	f	g	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13
n	o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	24	26

Then the algorithm can be expressed as follows. For each plaintext letter , substitute the cipher text letter :

$$C = E(3, p) = (p + 3) \bmod 26 \quad \ldots\ldots\ldots\ldots\ldots (1)$$

A shift may be of any amount, so that the general Caesar algorithm is :

$$C = E(k, p) = (p + k) \bmod 26 \quad \ldots\ldots\ldots\ldots\ldots (2)$$

Where takes on a value in the range 1 to 25.The decryption algorithm is simply

$$p = D(k, C) = (C - k) \bmod 26 \quad \ldots\ldots\ldots\ldots\ldots (3)$$

If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys. Figure 2.1 shows

the results of applying this strategy to the example ciphertext. In this case, the plaintext leaps out as occupying the third line.

KEY						
	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

Figure 2.1 Brute-Force Cryptanalysis of Caesar Cipher

There are three important characteristics which lead us to use brute force cryptanalysis:

1. The encryption and decryption algorithms are known.

2. There are only 25 keys to try.

3. The language of the plaintext is known and easily recognizable.

4.2 Transposition Techniques

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

Transposition cipher: An encryption method in which the positions of plaintext characters are shifted by a defined number of places to produce cipher text. Cipher text created with a transposition cipher is a permutation of the plaintext.

The simplest such cipher is the **rail fence** technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

The encrypted message is :

MEMATRHTGPRYETEFETEOAAT

This sort of thing would be trivial to crypt analyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example:

```
      Key: 4 3 1 2 5 6 7
Plaintext: a t t a c k p
           o s t p o n e
           d u n t i l t
           w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Thus, in this example, the key is 4312567.To encrypt, start with the column that is labeled 1, in this case column 3.Write down all the letters in that column. Proceed to column 4 which is labeled 2, then column 2, then column 1, then columns 5, 6, and 7.

To decrypt the above cipher text, divide the length of the cipher text by the key length and obtain the result (i.e. 28 / 7 = 4) , start with the column that is labeled 1, in this case column 3,Write down the first 4 letters of the cipher text (i.e. `TTNA`). Proceed to the column that is labeled 2, which is column 4, write down the second 4 letters of the cipher text (i.e. `APTM`), then column 2, then column 1, 5, 6, and 7. To obtain the plain text, read the resulted matrix row by row.

A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions.

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is re-encrypted using the same algorithm.

```
  Key: 4 3 1 2 5 6 7
Input: t t n a a p t
       m t s u o a o
       d w c o i x k
       n l y p e t z
Output: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ
```

To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is :

```
01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28
```

After the first transposition, we have:

```
03 10 17 24 04 11 18 25 02 09 16 23 01 08
15 22 05 12 19 26 06 13 20 27 07 14 21 28
```

Which has a somewhat regular structure. But after the second transposition, we have:

```
17 09 05 27 24 16 12 07 10 02 22 20 03 25
15 13 04 23 19 14 11 01 26 21 18 08 06 28
```

This is a much less structured permutation and is much more difficult to cryptanalyze.

Note

Although the Substitution and Transposition cipher are simple methods and their key is easy to break, these two methods are still used in modern and advanced encryption algorithms like **Data Encryption Standard (DES)** cipher which consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.

5. Symmetric and Asymmetric Cipher

All algorithms use some type of value to translate the plaintext to cipher text. Each algorithm performs steps using the supplied value to encrypt the data. The special value that the algorithm uses is the ***encryption key***. Some encryption algorithms use a single key, while others use more than one. As long as the sender and receiver both use the same algorithm and key, the process works.

Encryption Key: A code that enables the user to encrypt or decrypt information when combined with a cipher algorithm.

5.1 Private or Symmetric key algorithms

The easiest type of encryption to understand and use is the private key algorithm, also referred to as a symmetric key algorithm. It is symmetric because the decrypt function is a simple reversal of the encrypt function. In other words, it looks the same on both sides. (See Figure 2.2)

This type of algorithm is fast and easy to use, and a frequent choice for encrypting data. The key and the algorithm are all that is required to decrypt the file. Although this type of algorithm is common for encrypting files, it can be more difficult to use for message encryption. The ***problem*** is managing the encryption key. The key is required to decrypt a file or message. Plus, ***you have to find a way to get the key to the recipient in a secure manner.***

Private Key algorithm: An encryption algorithm that uses the **same key** to encrypt and decrypt.

The key is a value ***independent*** of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time (i.e.

two ***different keys*** will produce two ***different cipher texts*** even if the plain text and the algorithm ***are the same***).

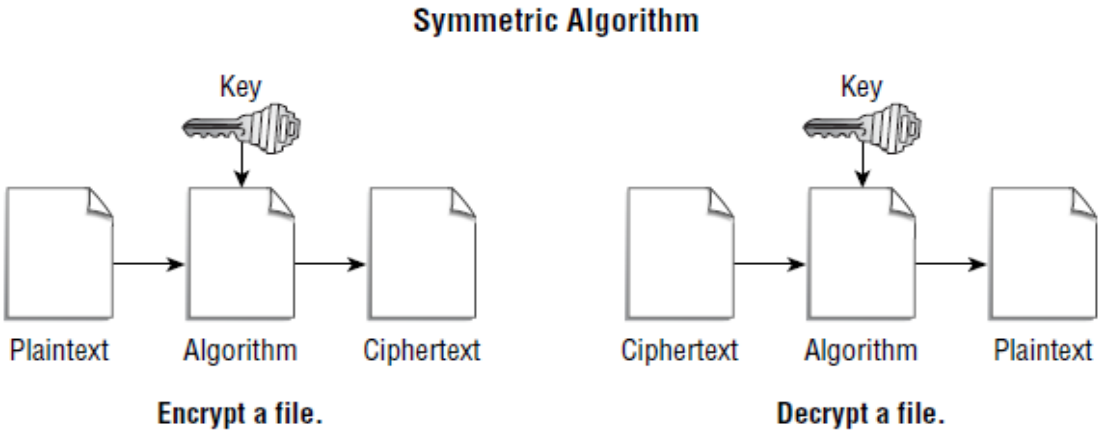


Figure 2.2 Symmetric (Private) key algorithms

Sender and receiver must have obtained copies of the secret key in a ***secure fashion*** and must keep the key secure. If someone is eavesdropping on all communication between you and your intended recipient, then he or she will likely intercept the encryption key as well as any encrypted data. With the key, they will be able to decrypt files at will. This is the main weakness point of Private key algorithms.

Many well-known symmetric encryption algorithms exist. Here are a few of the more common ones forensic investigators are likely to encounter:

Data Encryption Standard (DES)

- First published in 1977
- Adopted by the U.S. government standard for all data communications
- Uses 56-bit key (plus eight parity bits)
- Old and weak by today's standards

Blowfish

- Stronger alternative to DES
- Key size can vary from 32 bits to 448 bits

Advanced Encryption Standard (AES)

- The latest, strongest standard adopted by the U.S. government after an exhaustive competition among algorithms designs developed by leading world experts in cryptography

- Announced in 2000
- Based on the Rijndael cipher
- Key sizes are 128, 192, or 256 bits

5.2 Public, or Asymmetric Key Algorithms

The other type of encryption algorithm is the public key algorithm. This type of algorithm is also called asymmetric because the decrypt process differs from the encrypt process. An asymmetric encryption algorithm addresses the issue of key distribution by requiring two keys to complete the encrypt-decrypt process.

Asymmetric algorithms rely on one key for encryption and a ***different but related key*** for decryption. The two keys have the property that ***deriving*** the private key from the public key is ***computationally infeasible***. In other words , public-key encryption systems has a public key **e** and a corresponding private key **d**, hence, in secure systems, the issue of calculating **d** having **e** is mathematically impossible. This is called a ***key pair***. Private keys are meant to be secret and should not be disclosed to anyone. On the other hand, public keys can be distributed to anyone. The encryption algorithm uses the private key to encrypt plaintext and the public key to decrypt resulting cipher text. (See Figure 2.3)

Public key algorithm: An encryption algorithm that uses one key to encrypt plaintext and another key to decrypt cipher text.

The resulting process allows you to encrypt data with your private key. Anyone who has the public key can decrypt the file or message. This process lets anyone ***verify*** that a file or message ***originated*** from a specific person. If you can decrypt a file with Fred's public key, Fred had to encrypt it with his private key. Although this is great for sending messages and ***verifying*** the sender's identity, it doesn't add much value if all you want to do is encrypt some files.

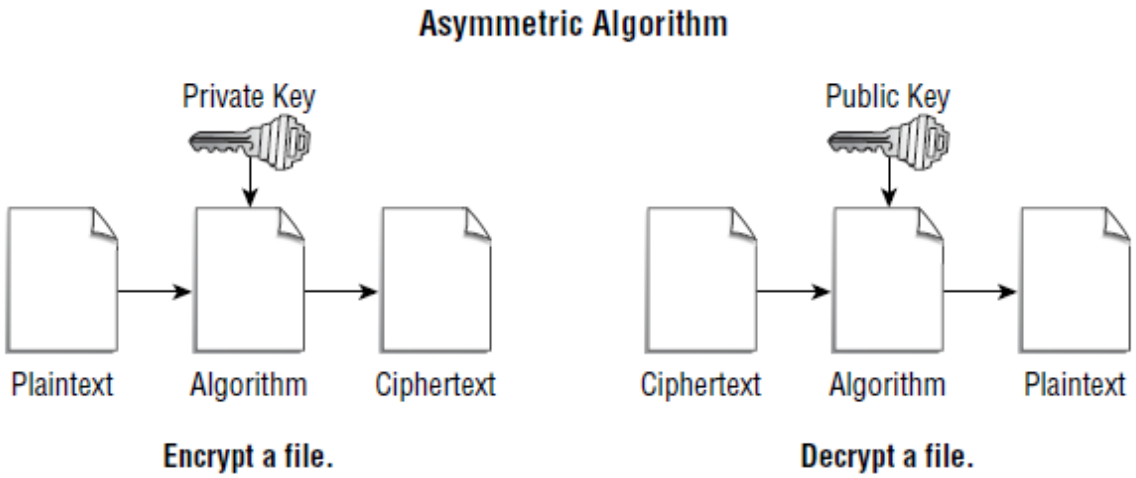


Figure 2.3 Asymmetric (Public) key algorithms

The most widely used public-key cryptosystem is RSA.

Symmetric Encryption	Public Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available (e.g. Key). 3. Knowledge of the algorithm plus samples of Cipher text must be insufficient to determine the key.	*Needed to Work* : 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available (e.g. Key). 3. Knowledge of the algorithm plus one of the keys plus samples of cipher text must be insufficient to determine the other key.

6. Cryptanalysis and Brute-Force Attack

Typically, the objective of attacking an encryption system is ***to recover the key in use*** **rather than** simply ***to recover the plaintext*** of a single cipher text. There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis**: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to **deduce** a specific plaintext or to deduce the key being used.
- **Brute-force attack**: Tries every possible key combination on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. We are using as much computing power as we can muster to guess the correct key. The more computers (or, more precisely, central processing units) we can throw at it, the faster we can break it. As you'll see, "faster" is a relative term when it comes to breaking keys.
- **Distributed attack:** A brute force attack can employ a number of idle computers and use them against the encrypted file, folder, or drive. This is known as a distributed attack since the computational burden is spread among multiple computers. Some agencies are getting quite creative in breaking encryption.

Chapter Three : Steganography

1. Introduction

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography.

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words “stegos” meaning “cover” and “grafia” meaning “writing” defining it as “covered writing”. In image steganography the information is hidden exclusively in images.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated. The strength of steganography can thus be amplified by combining it with cryptography.

Below is a brief comparison between cryptography and steganograhy :

Cryptography	Steganograhy
1. The encrypted message (cipher text) can be seen by anyone but cryptography make the cipher text not understandable. 2. The end result in cryptography is the cipher text. 3. The goal of secure cryptography is to prevent an interceptor from gaining any	1. Steganograhy hide the message with another media so that nobody can see the embedded message. 2. The end result of steganograhy is the stego-object (which hold the hidden message). 3. The goal of secure steganograhy is to

information about the plain text from the intercepted cipher text. 4. Any person has the ability to detect and modify the encrypted message. 5. Steganograhy cannot be used to adapt the robustness of cryptography system.	prevent an observer from even obtaining knowledge of the mere presence of the secret data. 4. The hidden message is imperceptible to anyone. 5. Cryptography can be used in conjunction with Steganograhy by hiding an encrypted message.

2. Steganography Terminology

Steganography is the art and science of invisible communication.

The goal of steganography is to avoid drawing suspicion to the transmission of a hidden message, so it remains undetected. If suspicion is raised, then this goal is defeated.

Steganalysis is the art of discovering and rendering such messages useless.

Steganalyst is one who applies steganalysis in an attempt to detect the existence of hidden information. In cryptanalysis, portions of the plaintext (possibly none) and portions of the ciphertext are analyzed. In steganalysis, comparisons are made between the cover-object, the stego-object, and possible portions of the message.

parallel attacks are available to the steganalyst:

• Stego-only attack. Only the stego-object is available for analysis.

• Known cover attack. The "original" cover-object and stego-object are both available.

• Known message attack. At some point, the hidden message may become known to the attacker.

3. Active and Malicious Attackers

During the design of a steganographic system special attention has to be paid to the presence of active and malicious attackers. **Active attackers** are able to change a cover during the communication process; Wendy could capture one stego-object sent from Alice to Bob, modify it and forward the result to Bob. It is a general assumption that an

active attacker is not able to change the cover and its semantics entirely, but only make minor changes so that the original and the modified cover-object stay perceptually or semantically similar. **An attacker is malicious** if he forges messages or starts steganography protocols under the name of one communication partner.

4. Principles of Steganography

The "classic" model for invisible communication was first proposed by Simmons as the "prisoner's problem." Alice[1] and Bob are arrested for some crime and are thrown in two different cells. They want to develop an escape plan, but unfortunately all communications between each other are arbitrated by a warden named Wendy. She will not let them communicate through encryption and if she notices any suspicious communication, she will place them in solitary confinement and thus suppress the exchange of all messages. So both parties must communicate invisibly in order not to arouse Wendy's suspicion; they have to set up a ***subliminal channel***. A practical way to do so is to hide meaningful information in some harmless message: Bob could, for instance, create a picture of a blue cow lying on a green meadow and send this piece of modern art to Alice. Wendy has no idea that the colors of the objects in the picture transmit information.

Throughout this course we will make the (for an actual prison perhaps unrealistic) *assumption that Alice and Bob have access to computer systems in their cells and are able to exchange messages in many different formats (e.g., text, digital images, digital sound, etc.).*

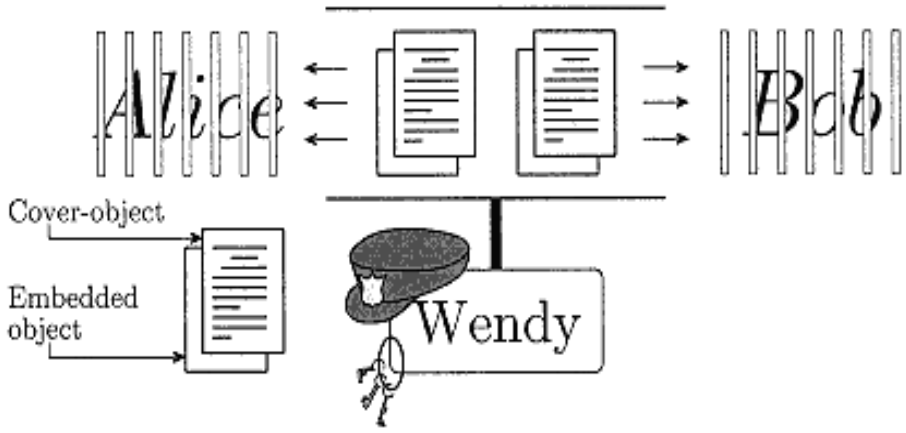


Figure 3.1 The prisoners' problem, illustrated. Courtesy of Scott Craver.

Unfortunately there are other problems which may hinder the escape of Alice and Bob. Wendy may alter the message Bob has sent to Alice. For example, she could change the color of Bob's cow to red, and so destroy the information; she then acts as an ***active*** *warden*. Even worse, if she acts in a *malicious* way, she could ***forge*** messages and send a message to one of the prisoners through the subliminal channel while pretending to be the other.

The above model is generally applicable to many situations in which invisible communication —*steganography* — takes place. Alice and Bob represent two communication parties, wanting to exchange secret information invisibly. The warden Wendy represents an eavesdropper who is able to read and probably alter messages sent between the communication partners (see Figure 3.1).

Whereas cryptographic techniques try to modify the contents of a message, steganography goes yet a bit further: it tries to hide the fact that a communication even exists. Two people can communicate covertly by exchanging unclassified messages containing confidential information. Both parties have to take the presence of a *passive*, *active* or even *malicious* attacker into account.

5. Frameworks for Secret Communication

Most applications of steganography follow one general principle, illustrated in Figure 3.2. 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 statistical pattern. Theoretically, covers could be any computer-readable data such as image files, digital sound, or written text.

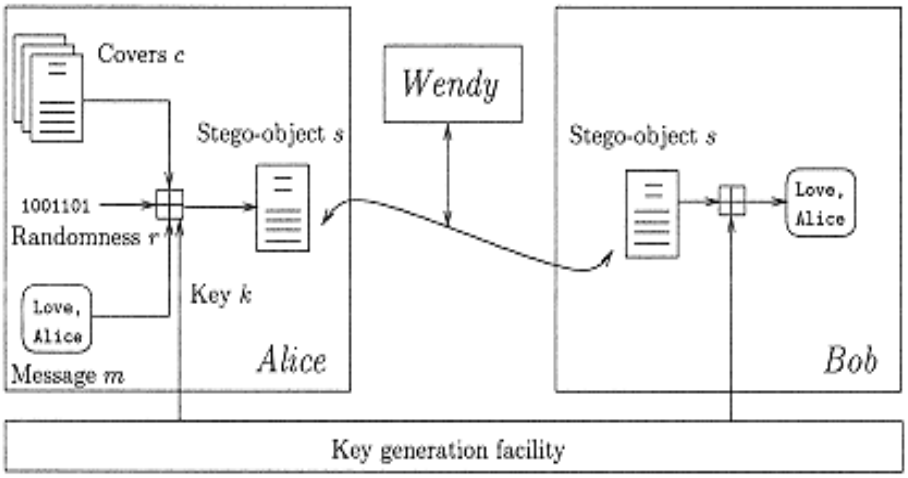


Figure 3. 2 Schematic description of steganography: Alice randomly chooses a cover c using her private random source *r* and embeds the message *m* in *c* using a key *k*, creating the stego-object *s* which she passes on to Bob. Bob reconstructs *m*
with the key k he shares with Alice.

Alice 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 $\{c_1, \ldots, c_n\}$ 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 can be replaced by secret information. As an example, due to measuring errors, any data which are the result of some physical scanning process will contain a stochastic component called ***noise***. Such random artifacts can be used for the submission of secret information. In fact, it turns out that **noisy data** has ***more advantageous properties*** in most steganographic applications.

Note

Obviously 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.

6. Types of Steganography Protocols

In the literature there are basically 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.

Pure Steganography

We call a steganographic system which does not require the prior **exchange** of some secret information (i.e. a stego-key) *pure steganography*. Formally, the embedding process can be described as a mapping $E : C \times M \rightarrow C$, where C is the set of possible covers and M the set of possible messages. The extraction process consists of a mapping $D : C \rightarrow M$, extracting the secret message out of a cover. Clearly, it is necessary that $|C| \geq |M|$.

Both sender and receiver must have access to the embedding and extraction **algorithm**, but the algorithms **should not be public**.

Definition: (Pure steganography) *The quadruple = <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.

In most practical steganographic systems the set C is chosen to consist of meaningful, and apparently harmless messages (like the set of all meaningful digital images, or like meaningful texts), 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***.

In the case of digital images or digital sound the correlation between two signals can be used as a similarity function.

If the cover is the result of some ***scanning*** process, the original cover can be digitized again and again. Due to the noise in the hardware, every process will produce a ***slightly different cover***. The sender could select one, best suitable for communication (Note that noisy cover is more ***advantageous*** than noise-free cover) . Such a technique, called **selection method of invisibility**.

Some steganographic methods combine traditional cryptography with steganography: the sender encrypts the secret message prior to the embedding process. Clearly, such a combination increases the security of the overall communication process, as it is more difficult for an attacker to detect embedded cipher text (which itself has a rather random appearance) in a cover. Strong steganographic systems, however, do not need prior enciphering.

Secret Key Steganography

With pure steganography, no information (apart from the functions *E* and *D)* is required to start the communication process; the security of the system thus **depends entirely** on its secrecy (steganography **algorithm**). This is not very secure in practice because this violates Kerckhoffs' principle (see Section 1. 4). So we must assume that Wendy **knows** the algorithm Alice and Bob use for information transfer. In theory, she is able to extract information out of every cover sent between Alice and Bob. The security of a steganographic system should thus rely on some secret information, the **stego-key**. Without knowledge of this key, nobody should be able to extract secret information out of the cover.

A secret key steganography system is similar to a **symmetric** cipher: the sender 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.

Definition: (Secret key steganography)

The quintuple $= \langle C, M, K, D_K, E_K \rangle$*, 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 a **secure exchange** of some key, although the transmission of additional secret information subverts the original intention of invisible communication. So as in cryptography, we assume that all communication parties are able to trade secret keys through a **secure channel**. Alice and Bob could agree on a stego-key before imprisonment. However, by using some characteristic features of the cover and a secure hash function *H* it is possible to calculate a key used for secret communication directly out of the cover: ***k = H (feature)***. If the embedding process does not change the "feature", the receiver is able to recalculate the key. Obviously such a feature has to be highly "cover dependent" to reach an adequate level of security (however, the security depends on the secrecy of *H*, thus violating Kerckhoffs' principle again). If the cover is a digital image, one could take all most significant bits of the cover's color values as a "feature". This method could be also used to calculate a secret session key out of a general key k' valid for a longer period of time, if the hash function depends on k' .

Note that: Any ***slight change*** in the ***hash function input*** will lead to a ***different result***.

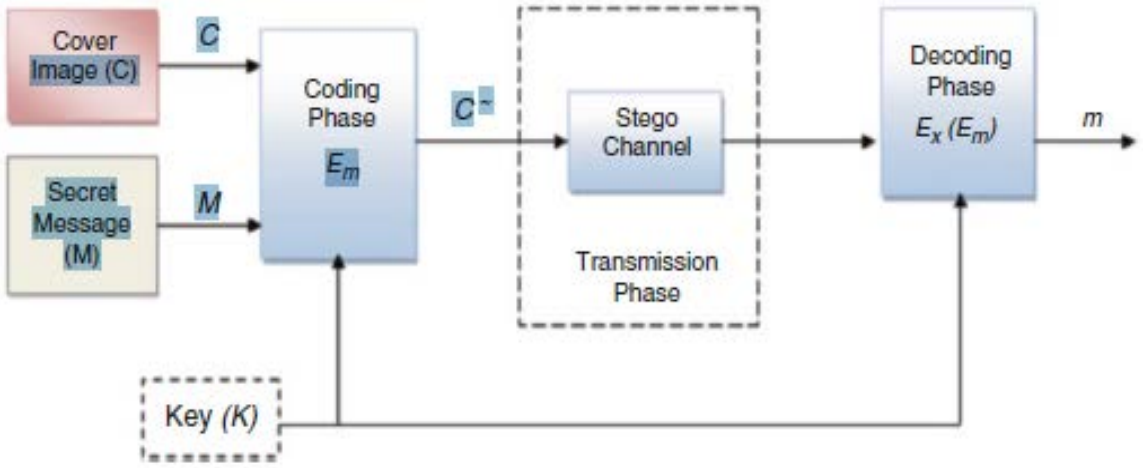


Figure 3.3 Secret key steganographic system model

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 to embed the secret message in the cover, the private key is used to reconstruct the secret message.

The sender (Alice) will use the public key during the encoding stage , while the receiver (Bob) use the private key which has a direct ***mathematical*** relationship to the general public key (i.e. the private key is ***derived*** from the public key) to decipher (extract) the secret message.

A protocol which allows public key steganography has been proposed by **Anderson**; it relies on the fact that encrypted information is random enough to "hide in plain sight": Alice ***encrypts*** the information with Bob's public key to obtain a random-looking message and ***embeds*** it in a channel known to Bob (and hence also to Wendy), thereby replacing some of the "natural randomness" with which every communication process is accompanied. We will assume that both the ***cryptographic*** algorithms and the ***embedding*** functions are publicly known. Bob, who cannot decide a priori if secret information is transmitted in a specific cover, will suspect the arrival of a message and will simply try to ***extract*** and ***decrypt*** it using his private key. If the cover actually contained information, the decrypted information is Alice's message.

Since we assumed that Wendy knows the embedding method used, she can try to *extract* the secret message sent from Alice to Bob. However, if the encryption method produces random-looking cipher text, Wendy will have no evidence that the extracted information is more than some random bits. She thus cannot decide if the extracted information is meaningful or just part of the natural randomness, unless ***she is able to break the cryptosystem***.

A more crucial point is that: If the stego-message is not targeted towards a ***specific person***, but for example is posted in an Internet group, the problem worsens. Although the protocol also works in this case (only the intended receiver ***can decrypt*** the secret message, since only he has the correct private key) all possible receivers have to try to decode every posted object.

7. Categories of Steganography Based on Cover Media

Steganography can be classified as text, image, video and audio steganography **depending on** the cover media used to embed the secret data.

7.1 Written Text Steganography

Documents may be modified to hide information by manipulating positions of lines and words. Text files can be use to carry information through adding spaces, tabs, "invisible" characters, "extra" lines and "extra" spaces.

Many ways have been proposed to store information directly in messages. Infrequent typing or spelling errors could be introduced, commas omitted, and words replaced by synonyms. Most of them are not serious options, as they degrade the text heavily. Additionally the embedding task requires the interaction of the user, it therefore cannot be automated.

Secret information can be stored in the ***size*** of interline or interword spaces. If the space between two lines or two words is smaller than some threshold, a "0" is encoded, otherwise a "1". A similar method can be used to transmit information in ASCII text: infrequent additional white space characters are introduced to form the secret message.

As an example of hiding information in text is known as a **null cipher or open code**. The secret message is camouflaged in an innocent sounding message. The following is such a null cipher:

" Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed. Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming anyday"

Taking the ***third letter*** **(key = 3)** in each word the following message emerges :

"Send Lawyers, Guns, and Money"

Another examples by using word shifting can be used to help identify an original document. A similar method can be applied to display an entirely different message. Take the following sentence:

"We explore new steganographic and cryptographic algorithms and techniques throughout the world to produce wide variety and security in the electronic web called the Internet"

and apply some word shifting algorithm to obtain the following sentence:

"We explore new steganographic and cryptographic algorithms and techniques throughout the world to produce wide variety and security in the electronic web called the Internet"

By overlapping the two sentences, we obtain the following:

"We explore new steganographic and cryptographic algorithms and techniques throughout the world to produce wide variety and security in the electronic web called the Internet"

This example is achieved by shifting the words explore, world, wide and web up by one point and shifting the word the down by one point. Independently, the sentence containing the shifted words appears harmless, but combining this with the original sentence produces a different message:

explore the world wide web.

It is an open question whether secure and robust steganography is possible with text messages. An attacker can simply try to **reformat** the text and so **destroy** all information encoded in the text format. Additionally, text messages can be stored in different formats (like HTML, Postscript, PDF, or RTF); ***the conversion from one format to another*** might also be harmful to the embedded message.

7.3 Image Steganography

The image steganography is the process in which we hide the secret data within an image so that there will not be any perceived visible change in the original image.

This type of steganography exploits the weakness of the human visual system (HVS). HVS cannot detect the variation in luminance of color vectors at collection of color pixels.

In this steganography , images are commonly used as a cover file. There are different file formats available for digital image and for these file formats different algorithms are available such as Least Significant Bit (LSB) , Masking and Filtering and image transformation.

7.3 Video Steganography

In video steganography a large amount of data can be hidden inside the cover file (which is the video file) exploiting the fact that it is a flow of a sequence of images and sound.

The separation of video into audio and images (frames) results in the efficient method for data hiding . The use of video file as a carrier medium for steganography is more eligible as compared to other techniques.

7.4 Audio Steganography

Steganography in general relies on the imperfection of the human auditory and visual system. Audio Steganography takes advantage of the human psychoacoustics phenomenon of the human auditory system. Psychoacoustics or auditory masking property renders a weak ton imperceptible in the presence of a strong tone in its temporal or spectral neighborhood.

In audio Steganography , secret message is embedded into digitized audio signals which result in a slight modification in the binary stream of a sound file.

Existing audio steganographic techniques can insert messages in WAV, AU, and even MP3 sound files. There are a number of methods like LSB Coding, Phase Coding, Spread Spectrum, Echo hiding which are used for audio steganography.

8. Steganographic Techniques

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

Steganographic Techniques can be classified in six categories, although in some cases an exact classification is not possible:

- **Substitution systems** substitute parts of a cover with a secret message;
- **Transform domain techniques** embed secret information in a transform space of the signal (e.g., in the frequency domain);
- **Spread spectrum techniques** adopt ideas from spread spectrum communication;
- **Statistical methods** encode information by changing several statistical properties of a cover and use hypothesis testing in the extraction process;
- **Distortion techniques** store information by ***signal distortion*** and measure the ***deviation*** from the ***original cover*** in the decoding step;
- **Cover generation methods** encode information in the way a cover for secret communication is created.

In the following section the most commonly used technique which is the **Substitution** method will be discussed.

Substitution Systems and Bitplane Tools

A number of methods exist for hiding information in various media. These methods range from LSB coding—also known as bitplane or noise insertion tools—manipulation of image or compression algorithms in order to modify image properties such as luminance. 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

From the spatial domain image steganography techniques, the LSB technique is considered the most common one, because rather than its simplicity, the amount of modified information caused by altering the LSB is very low**.** This approach is common in steganography and is 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.

The image formats typically used in such steganography methods are lossless and the data can be directly manipulated and recovered. Some of these steganography methods apply ***compression*** and ***encryption*** in addition to steganography services. These services provide better security of the hidden data. Even so, the bitplane methods are rather brittle and vulnerable to corruption due to small changes they made to the carrier.

The embedding process consists of choosing a subset $\{j_1, \ldots, j_{l(m)}\}$ of cover elements and performing the substitution operation $c_{ji}\, m_i$ on them, which exchanges the LSB of c_{ji} by m_i (m_i 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.

LSB steganography relies on the fact that replacing one or more of the last 1-4 bits of cover image's pixels is not perceptible by human visual system; the resulted stego-

image is identical to the cover image because the modification of the cover image pixels do not produce a major difference in the image. Hence, altering the values of 3-LSB bits of image's data will make the image's modification invisible for any human eyes, due to the sleight difference in colors it makes. Figure 3.4 shows the percentage of the information that conveyed by each bit of one byte of data.

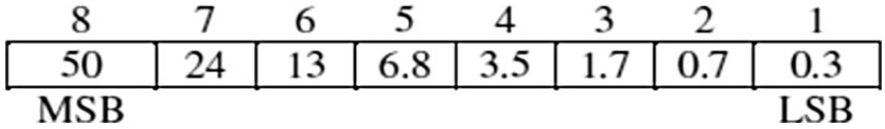


Figure 3.4 The amount of information each bit of one byte of data can hold.

This basic scheme of LSB is presented in example 3.1 and 3.2.

Example 3.1 Embedding process: least significant bit substitution in **color** image suppose that we have three adjacent pixels (nine bytes) with the following RGB color space :

R	**G**	**B**
10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

Now suppose we want to hide the following 9 bits of data 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** are the embed pixels):

R	**G**	**B**
1001010**1**	0000110**0**	1100100**1**
1001011**1**	0000111**0**	1100101**1**
1001111**1**	0001000**0**	1100101**1**

Example 3.2 Embedding process: least significant bit substitution in **grey scale** image Assume that the bit stream to be hidden is

10101100101111010110101011001101001101101010

And there are two blocks in which the above bit stream can be hidden , assume we want to hide 2 bits in the first block and 3 bits in the second. Below is an illustration to the LSB embedding process.

Cover_image

15	200	5	250	100	5
60	2	0	120	116	10
10	8	100	50	110	25

Stego_image

14	202	7	253	98	6
60	2	3	123	114	11
11	9	101	51	106	26

Secret bits

10101100101111010110101011001101001101101010

First block	Second block

15= 00001111=00001110=14	250=11111010=11111101=253
200=11001000=11001010=202	100=01100100=01100010=98
5=00000101=00000111=7	5=00000101=00000110=6
60=00111100=00111100=60	120=01111000=01111011=123
2=00000010=00000010=2	116=01110100=01110010=114
0=00000000=00000011=3	10=00001010=00001011=11
10=00001010=00001011=11	50=00110010=00110011=51
8=00001000=00001001=9	110=01101110=01101010=106
100=01100100=01100101=101	25=00011001=00011010=26

Figure 3.5: example of the LSB embedding process

Hiding in edge pixels

By using the human eyes, it will be very difficult to observe the difference between cover image and stego-image. However, when the number of least significant bits for each cover pixel exceeds three bits, the LSB technique generally causes a noticeable (i.e. attracting attention) distortion in the cover image. In order to decrease the distortion caused by LSBs technique, several methods for LSB steganography has been presented, the well known are ***Adaptive methods*** which alter the number of LSBs in each pixel, for instance (**hide more bits in edge pixels and less bits in smooth pixels**) . Adaptive methods possess a ***better image quality*** than the traditional LSB substitution methods. However, this is accomplished with the **cost** of ***decreasing the payload*** capacity.

Note

For the example 3.2 above the 2 bits will be embed in the smooth pixels and the 3 bits will be embed in the edge pixels.

Problems of Least Significant Bit Substitution

First problem is : 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 *l(m)* will normally have fewer bits than cover-elements *l(c)*, the embedding process will be ***finished long before*** the end of the cover. In this case, the sender can leave all other cover elements unchanged. This can, however, lead to a serious security problem: the first part of the cover will have ***different statistical properties*** than the second part, where no modifications have been made; hence attacker will suspect secret communication. To overcome this problem, for instance the public domain program PGMStealth enlarges the secret message with random bits (so that *l(c)*

= *l(m)*) in an attempt to create an equal change in randomness at the beginning and the end of the cover. The embedding process thus changes far more elements than the transmission of the secret would require. Unfortunately, the probability that an attacker will suspect secret communication increases.

Second problem remains to be solved: in which way should the cover pixels cji be chosen? The 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* used ***as a seed*** for a random number generator, they can create a random sequence $k_1, ..., k_{l(m)}$ and use the elements with indices for information transfer.

$$j_1 = k_1$$
$$j_i = j_{i-1} + k_i, \quad i \geq 2 \qquad (3.2)$$

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 k_i and therefore the entire sequence of element indices j_i , thus it will be an easy mission for the attacker to recover the embedded secret message if he has the key.

Distortion Techniques

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.

9. Characteristics of Steganography

- **Hiding capacity :** represent the capacity of the data that can be hidden without distorting the carrier file. It determine the number of bytes that can be covered within the carrier file without damaging it. For instance, in image steganography,

it is important to hide **as much as possible** data inside the carrier image without changing its brightness, without making it blurry and without changing its size. This would be a key element in making the hidden data imperceptible and the carrier image innocent and unsuspicious.

- **Imperceptibility:** represent the imperceptibility of the carrier file after hiding the secret data in it. It refers to the ability of the steganography algorithm to hide data in an **undetectable way** so much so that no one can see any visible artifacts or distortion in the carrier file. It therefore avoids drawing the suspicions and obscures the fact that the secret communications is taking place.
- **Irrecoverability** : the irrecoverability of the hidden data in case they were detected refers to how much an intercepted carrier file can be easily decoded and reversed so as to extract the hidden data inside it. An irrecoverable steganography algorithm makes it hard for eavesdroppers and unauthorized third parties to recover the hidden data from the carrier file despite knowing that steganography has been employed.

10.Performance Evaluation

In general, human beings are the better evaluator of every vision system, however, it is not constantly possible for human being to evaluate those vision systems in a quantitative manner. Hence, in order to evaluate any proposed method, it is necessary to provide a quantitative evaluation measures.

The PSNR (Peak Signal to Noise Ratio) is the parameter that assess the quality of Stego image with respect to the original image. It calculates the imperceptibility of the Stego image. In simple form we can say that it calculates and analyzes that how much similar two images are i.e. the similarity between Stego image and original image . Higher the value of PSNR of stego image higher will be the quality of Stego image or we can say that higher will be the imperceptibility of hidden message behind the pixels of an image.

MSE (Mean Square Error) is the parameter that calculates the magnitude of average error between the original image and stego image. The difference between the observed values of original and stego image are squared and then their average is calculated .

RMSE is used mainly in ***case of large errors*** because it provides relatively high weight to these errors. So, RMSE is very demanding when large errors are undesirable in carrier file. The small we get the value of RMSE higher will be the quality of system.

If I_C and I_S are the cover image and the stego-image respectively, then the MSE and PSNR can be calculated using the following equations:

$$MSE = \frac{1}{M \cdot N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [I_s(i,j) - I_c(i,j)]^2 \qquad (1)$$

$$PSNR = 10 \log 10 \left(\frac{C_{max}^2}{MSE} \right) \qquad (2)$$

where M : number of rows in the cover images
N : number of columns in the cover images
C_{max} : maximum value that the cover image pixels can hold (e.g. for gray scale image the maximum value is 255
$I_C(i,j)$: is the intensity of (i,j) th pixel in cover image.
$I_S(i,j)$: is the intensity of (i,j) th pixel in stego image.

Obviously, a higher PSNR means a better quality of embedding operation, in other words, the stego-image is identical to the cover image.

Example 3.3:
Compute the MSE and PSNR to the images in example 3.2 above.

Cover_image

15	200	5
60	2	0
10	8	100

Stego image

14	202	7
60	2	3
11	9	101

MSE = $(15-14)^2 + (200-202)^2 + \ldots\ldots + (100-101)^2 / (3 \times 3)$
= $(1 + 4 + 4 + 0 + 0 + 9 + 1 + 1 + 1) / 9 = 21/9 = 2.333$

PSNR = $10 \log 10 \frac{(255)^2}{2.333} = 10 \log_{10} 27871 = 10 \times 4.445$
= 44.45

This imply a good quality of embedding operation.

Chapter Four : Watermarking

1. Introduction

Digital watermarking is the method of embedding secure data (the watermark) into digital multimedia content e.g. image or video which may be visible or invisible. This is used to verify the content or to recognize the identity of the digital content's owner.

Digital watermark has acquired this importance, because they contribute to the conservation of copyright and authorship and ownership rights in the digital world, in light of the increasing piracy and illegal copying operations, especially over the Internet. With the growing E-commerce, a big problem for this type of trade, increasing the need for IT reservation of these rights.

Two other technologies that are closely related to steganography are watermarking and fingerprinting. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography. These requirements of a good steganographic algorithm will be discussed below. In ***watermarking*** **all** of the instances of an object are "marked" in the **same way**. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection. With ***fingerprinting*** on the other hand, **different unique marks** are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties.

As opposed to steganography, in watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge – sometimes it may even be **visible** – while in steganography the imperceptibility of the information is crucial. A successful attack on a steganographic system consists of an adversary observing that there is information hidden inside a file, while a successful attack on a watermarking or fingerprinting system would not be to detect the mark, but to remove it.

Steganographic methods are usually not robust against modification of the data, or have only limited robustness and protect the embedded information against technical modifications that may occur during transmission and storage, like format conversion, compression, or digital-to-analog conversion.

Watermarking, on the other hand, has the additional notion of resilience against attempts to remove the hidden data. Thus, watermarking, rather than steganography principles are used whenever the cover-data is available to parties who know the existence of the hidden data and may have an interest removing it. 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.

Furthermore, digital watermarking may also be used for fingerprinting applications in order to distinguish distributed data sets.

2. Watermark Terminology

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, practical working methods and systems have been developed.

Digital watermark is a hidden message inside a digital image, or audio, or digital video file or other digital files that are traded commercially file. This message is stored inside the contents of the same file, there is no need for additional storage space. Space is very important and limited, therefore this letters (tag) are often small, which contain a limited amount of data, the number often. This could be a watermark for the product name, publisher name, company data, serial number, or a special definition of the buyer to ensure his rights to ownership of what he bought and protected in cases of investigation number. Copies may also be allowed to explain his production of them.

Digital watermarks can be classified according to **human perception** into:

a. ***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. The application of visible watermarks to video is of course also possible and under some circumstances one might even think of embedding an **audible** watermark into audio. For the rest of this chapter we will focus on imperceptible watermarks.

b. ***Invisible watermarks***: The locations in which the watermark is embedded are secret, only the authorized persons extract the watermark. Some mathematical calculations are required to retrieve the watermark. This kind of watermarks is not viewable by an ordinary eye. Invisible watermarks are more secure and robust than visible watermarks.

Digital watermarks can be classified according to **usage** into:

a. ***Fragile watermarks*** are watermarks that have only very limited robustness. They can be destroyed easily with slight modifications in the watermarked data. They are applied to **detect** modifications of the watermarked data, rather than conveying inerasable information.

b. ***Robust watermarks,*** 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. A practical implication of the robustness requirement is that watermarking methods can typically ***embed much less*** information into cover-data than steganographic methods. Robust watermark are used mainly in ***copyright protection***.

c. ***Fingerprinting*** is the term that denotes special applications of watermarking. In fingerprinting ***different unique marks*** are embedded in ***distinct copies*** of the carrier object that are supplied to different customers. This enables the intellectual property owner to ***identify customers*** who break their licensing agreement by supplying the property to third parties. 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.

3. 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). Figure 4.1 shows the generic watermark embedding process. The

input to the scheme is the watermark, the cover-data and an **optional public or secret key.** The watermark can be of any nature such as a number, text, or an image. The key may be used to ***enforce security***, which is the prevention of unauthorized parties from recovering and manipulating the watermark. All practical systems employ at least one key, or even a combination of several keys. In combination with a secret or a public key the watermarking techniques are usually referred to as secret and public watermarking techniques, respectively. The output of the watermarking scheme is the *watermarked data.*

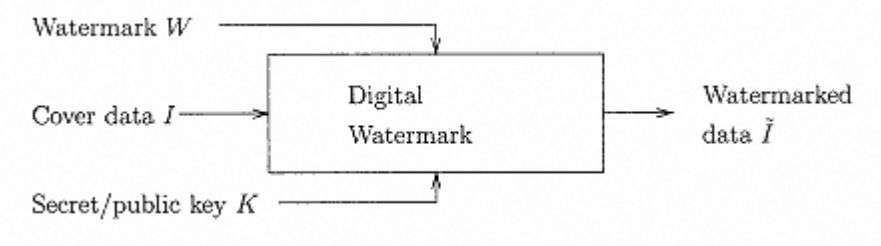


Figure 4.1 Generic digital watermarking scheme.

The generic watermark recovery process is depicted in Figure 4.2. Inputs to the scheme are the watermarked data, the secret or public key, and, **depending on the method**, the original data and/or the original watermark. The output is either the recovered watermark *W* or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the data I' under inspection.

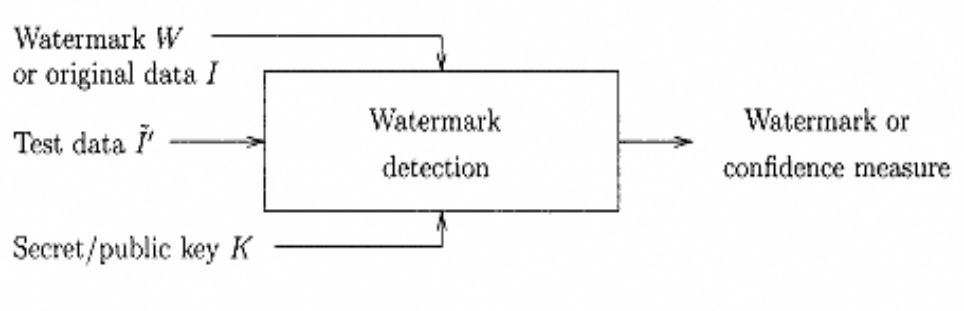


Figure 4.2 Generic watermark recovery scheme.

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.

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

a. **Private watermarking** (also called nonblind watermarking) systems require at least the ***original data*** *I* to extract the watermark *W* from the watermarked data I′. ***Type I*** systems ***extract the watermark*** *W* from the possibly distorted data I′ and use the ***original data*** *I* as a ***hint*** to find where the watermark could be in I′. ***Type II*** systems besides, require a copy of the embedded watermark *W* for extraction and just yield a "yes" or "no" answer to the question: does I′ contain the watermark *W*? $(I' \times I \times K \times W \rightarrow \{0, 1\})$.

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.

b. **Semiprivate watermarking** (or semiblind watermarking) does not use the original data *I* for ***detecting*** the watermark *W* from the watermarked data I′. This type of watermarking also answers the same question: does I′ contain the watermark *W*?

$(I' \times K \times W \rightarrow \{0, 1\})$.

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.

c. **Public watermarking** (also referred to as *blind* or *oblivious* watermarking) remains the most challenging problem since it requires neither the original data *I* nor the embedded watermark *W*. Indeed, such systems really extract *n* bits of information (the watermark) from the marked data : $I' \times K \rightarrow W$.

Depending on the application, the input to both generic schemes is usually in the form of uncompressed or compressed data. For obvious reasons the watermarking techniques should exploit the nature of the input. It would not make much sense if a watermarking scheme for MPEG-2 video requires single decompressed video frames to perform the watermarking since this would include a decoding and re-encoding procedure and make the entire watermarking process computationally too expensive. However, it should be noted that for some applications it may be interesting to design

watermark detection schemes, which allow for a watermark to be detected regardless of the domain where it was embedded.

4. Watermark Basic Requirement

Depending on the watermarking application and purpose, different requirements arise resulting in various design issues. The most common requirements 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 (pixels, edge, features, etc.) that are **used** for watermark embedding are only modified by a small amount.
- **Robustness.** Robustness of the watermarked data against **modifications** and/or **malicious attacks** is one of the key requirements in watermarking. The robustness of the algorithm can be **defined** as the ability of the watermark detector **to extract** the embedded watermark after applying common **signal processing operations (**e.g. image cropping, resizing, rotating, and compression). The ultimate watermarking method should resist any kind of distortion introduced by standard or malicious data processing. Thus, practical systems must implement a **compromise** between ***robustness*** and the competing requirements like ***invisibility*** and ***information rate***.
- **Watermark security.** In most applications, such as copyright protection, the secrecy of embedded information needs to be assured. Watermark security implies that the watermark should be difficult to **remove or alter** without **damaging** the host signal. If secrecy is a requirement, for example, as soon as a copyrighted image is opened in a photo editing software program the user is informed by a note indicating that the image is protected.
- **The use of keys.** 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 or 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.

Requirements for having a successful watermark in Summary

In summary, the below requirements have to be taken into consideration when designing successful watermarking techniques:

- Watermark should be imperceptible as to not interfere with the "viewing" of the data (see, listen, watch, etc.).
- Watermark must survive common data modification, such as for image, cropping, resizing, compression, etc.
- Watermark must be difficult or impossible to remove, at least without having to **destroy** the original data noticeably.
- There should be a way to detect the watermark by the appropriate authorities when required (i.e. using Key).

5. Watermarking Applications

The requirements that watermarking systems have to comply with are always **based** on the application. Thus, before we review the requirements and the resulting design considerations, we will present some applications of watermarking. 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.

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.

Fingerprinting for Traitor Tracking

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. Because the distributions of individually watermarked copies allow collusion attacks, the embedded watermarks have to be designed as collusion-secure. 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.

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.

Watermarking for Content Authentication

In authentication applications, the objective is to **detect modifications** of the data and hence, **verify the originality** of the object contents. This can be achieved with so called "**Fragile Watermarks**" that have a low robustness to certain modifications like compression, but are impaired (damaged) by other modifications. For this purpose, fragile watermark is mainly applied to content authentication because it is very sensitive to attacks; i.e. it can **detect slight changes** to the watermarked file. 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.

6. Evaluation and Benchmarking of Watermarking Systems

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:

- **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.
- **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 algorithm**, which in turn ***increases*** perceptibility of the watermark.
- **Size and nature of the watermarked data.** The size of the watermarked 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***. In addition to the size of the data, the nature of the watermarked data also has an important impact on the watermark robustness; ***ordinary*** image does ***not require*** a robust watermark.
- **Secret information (e.g. key).** Although the amount of secret information has no direct impact on the perceptibility of the watermark and the robustness of the watermark, it plays an important role in the security of the system. 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. The reader should also keep in mind that many security systems fail to resist very simple attacks because the system designers did not obey basic cryptographic principles in the design.

Below is a brief comparison between cryptography and watermark:

Cryptography	watermark
1. The encrypted message (cipher text) can be seen by anyone but cryptography make the cipher text not understandable. 2. The end result in cryptography is the cipher text. 3. The goal of secure cryptography is to prevent an interceptor from gaining any information about the plain text from the intercepted cipher text. 4. Any person has the ability to detect and modify the encrypted message.	1. Watermark is invisible or perceptually visible depending on the requirement. 2. The information hidden by a watermark system is always associated to the digital object to be protected or to its owner. 3. The important data is the external data (e. g. image or video) the internal data are additional data for protecting the external data and to prove ownership. 4. The hidden message cannot be modified or removed by anyone.

Chapter Five : Digital Forensic

1. Introduction

Forensic sciences are a scientific method of collection, investigation and analysis used to solve some kind of legal problem. ***Digital forensics*** isn't limited to computers. It encompasses any kind of electronic device that can store data. These devices include cell phones, smart phones, tablets, and GPS units.

Digital forensics is used in a number of areas, not just in catching identity thieves and internet predators. For example, it's being used on the battlefields to gather intelligence. The rapid exploitation of information pulled from cell phones and other devices is helping digital forensic troops identify and eliminate terrorists and insurgents.

Your computer will betray you. This is a lesson that many criminals, politicians, and ordinary citizens have learned the hard way. You are leaving a trail, albeit a digital one; these 1s and 0s capture our "footprints" as we go about our daily life.

Cell phone records, ATM transactions, web searches, e-mails, and text messages are a few of the **footprints** we leave. As a society, our heavy use of technology means that we are literally drowning in electronically stored information.

The impact of our growing digital dependence is being felt in many domains, not the least of which is the legal system. Everyday, digital evidence is finding its way into the world's courts. This new form of evidence presents some very significant challenges to our legal system. ***Digital evidence*** is considerably ***different from paper documents*** and can't be handled in the same way. Change, therefore, is inevitable.

Although forensic science has been around for years, digital forensics is still in its infancy. It's still finding its place among the other more established forensic disciplines, such as **DNA and toxicology**. Standards and best practices are still being developed.

1.1 WHAT IS FORENSIC SCIENCE?

Forensics is the application of science to **solve** a legal problem. In forensics, the law and science are forever integrated. Neither can be applied without paying respect to the other. The best scientific evidence in the world is **worthless** if it's inadmissible in a court of law.

1.2 WHAT IS DIGITAL FORENSICS?

The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources in a manner that is acceptable in a legal proceeding for the purpose of facilitating or furthering the reconstruction of events found to be criminal.

Digital forensics encompasses much more than **just** laptop and desktop computers. Mobile devices, networks, and "cloud" systems are very much within the scope of the discipline. It also includes the analysis of images, videos, and audio (in both analog and digital format). The focus of this kind of analysis is generally authenticity, comparison, and enhancement.

Digital Forensic investigations

A process that uses science and technology to examine digital evidence so that the results are admissible in court of law, to answer questions about events that occurred.

Digital evidence

Is any information of probative value that is either stored, transmitted in binary form. This field includes not only computers in the traditional sense but also any data found in digital storage devices.

2. Digital Forensic Main Concepts

Computer forensics has become a hot topic in computer security circles and in the legal community. In looking at the major concepts behind computer forensics, the main emphasis is on the below concepts:

- Identify meaningful evidence.
- Determine how to preserve the evidence.
- Extract, process, and interpret the evidence.
- Ensure that the evidence is acceptable in a court of law.

Key skills in computer forensics are knowing the best places to look for evidence, and knowing when to stop looking. These skills come with time and experience.

3. USES OF DIGITAL FORENSICS

Digital forensics can be used in a variety of settings, including criminal investigations, civil litigation, intelligence, and administrative matters.

3.1 Criminal Investigations

Although those investigations certainly focus on conventional evidence, they are by no means the only one. In today's digital world, electronic evidence can be found in almost any criminal investigation conducted. Homicide, robbery, and burglary are just a few of the many examples of "analog" crimes that can leave digital evidence.

3.2. Civil Litigation

Now a day, the use of digital forensics in civil cases is big business. As part of a process known as Electronic Discovery (eDiscovery), digital forensics has become a major component of much high civil litigation.

In a civil case, both parties are generally entitled to examine the evidence that will be used against them prior to trial. Today, parties no longer talk about filing cabinets, ledgers, and memos; they talk about hard drives, spreadsheets, and file types. Some paper-based materials may come into play, but it's more the exception than the rule. The rules of evidence or federal rules, govern how digital evidence can be admitted during civil litigation.

Digital evidence can quickly become the focal point of a case, no matter what kind of legal proceeding it's used in. The legal system and all its players are struggling to deal with this new reality.

Electronic Discovery or eDiscovery

Refers to any process in which stored information is located, collected, prepared, reviewed and secured with the intent of using it as **evidence** in a civil or criminal legal case.

Electronic discovery produces electronic documents for litigation. Data that is created or stored on a computer, computer network, or other storage media are included in e-discovery. **Examples** of such are e-mail, word-processing documents, plaintext files, database files, spreadsheets, digital art, photos, and presentations.

3.3. Intelligence

Terrorists, the purview of our intelligence agencies, have also joined the digital age. Terrorists have been using information technology to communicate, recruit, and plan attacks. **In Iraq**, our armed forces are exploiting intelligence collected from digital

devices brought straight from the battlefield. This process is known as DOMEX (Document and Media Exploitation). DOMEX is paying large dividends, providing actionable intelligence to support the soldiers on the ground.

3.4 Administrative Matters

Digital evidence can also be valuable for incidents other than litigation and matters of national security. Violations of policy and procedure often involve some type of electronically stored information, **for example**, an employee operating a personal side business, using company computers while on company time. That may not constitute a violation of the law, but it may warrant an investigation by the company.

4. DIGITAL FORENSIC TOOLS

Digital forensic tools make our work much more efficient or even possible. There are tools for specific purposes as well as tools with broader functionality. They can come in the form of both hardware and software. They can be commercial tools that must be purchased or they can be open source that are freely available. There are advantages and disadvantages to all. Keep in mind, no single tool does everything or does everything exceedingly well. As such, it's a good practice to have multiple tools available. Using multiple tools is also a great way to validate your findings. The same results, with two different tools, significantly increase the reliability of the evidence.

4.1 Hardware

There are many hardware tools out there designed and built specifically for digital forensics. **Some of these tools** include cloning devices, cell phone acquisition devices, write blockers, portable storage devices, adapters, cables, and more.

As you might expect, digital forensics is heavily dependent on an **assortment of hardware** such as PCs, servers, write blockers, cell phone kits, cables, and so on. Figure 1 shows a well-equipped digital forensic workstation.

Computers are the backbone of any digital forensics lab. So as an examiner you will need the **best** computer workstation you can afford. Digital forensic exams require quite a bit of computing power. These jobs can tax even the best systems and crush those that don't measure up. A good exam machine has multiple multi-core processors, as much RAM as you can get (the more the better), and large, fast hard drives. Forensic software

manufacturers provide detailed lists of minimum and suggested hardware requirements. Straying below the minimums is done at your own risk.

Digital forensics is no longer a "PC centric" endeavor. Small-scale devices such as cell phones and GPS units are pouring into labs across the country. These devices require different hardware from that used on laptops and desktops.

Several companies make hardware cloning devices. If you recall, a **forensic clone** is a ***"bit stream"*** copy of a particular piece of media such as a hard drive. These tools can really speed up the process, cloning multiple drives at once. They can also provide write protection, hash authentication, drive wiping, an audit trail, and more.

OTHER EQUIPMENT

The hardware and software we discussed earlier are not the only equipment needed. **Crime scene kits** are very useful outside the lab. These kits are preloaded with all of the supplies an examiner would need in the field to collect digital evidence. **Kits contain** standard items such as pens, digital camera, forensically clean storage media, evidence bags, evidence tape, report forms, permanent markers, and the like.

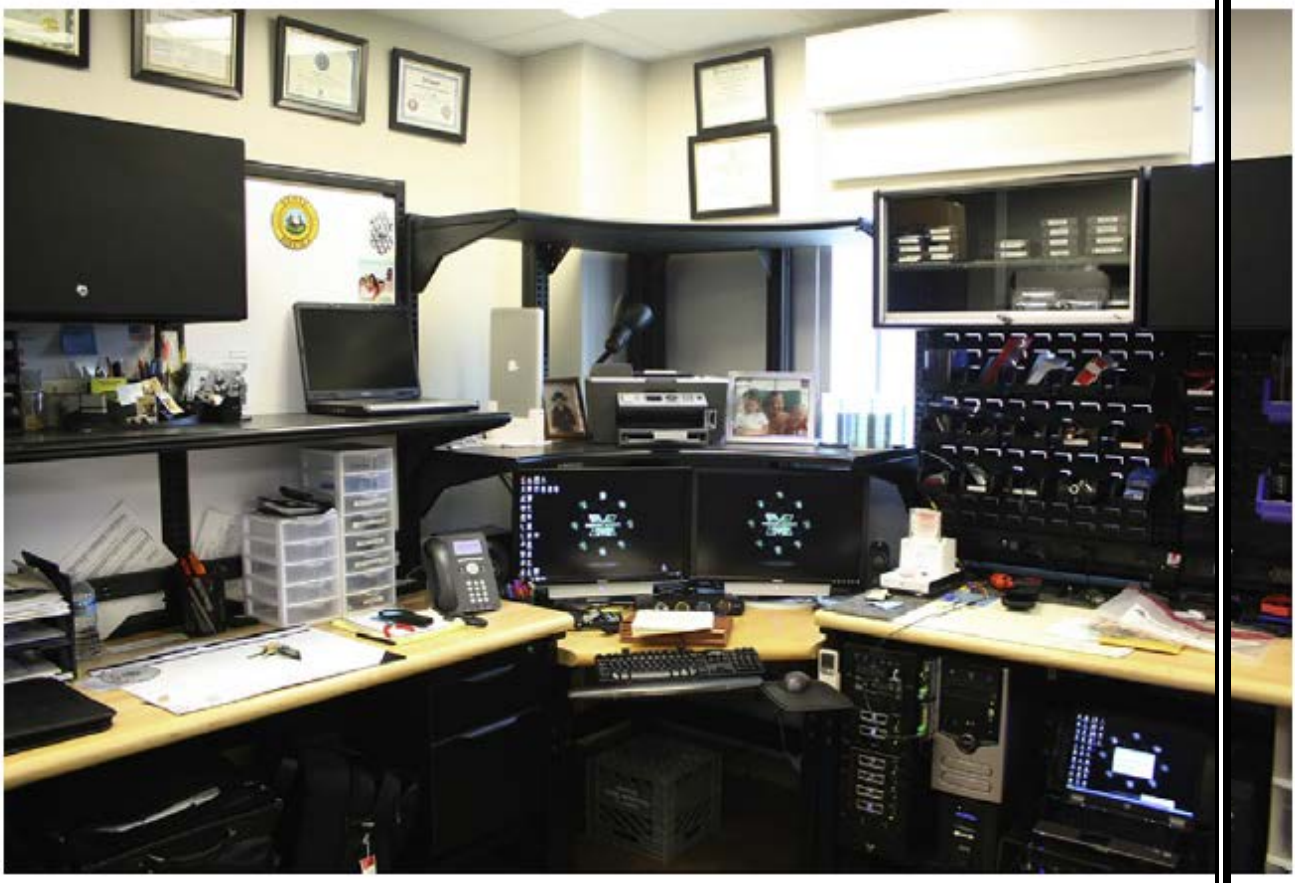


FIGURE 1 : One of the workstations in the West Virginia State Police Digital Forensics Lab located at the Marshall University Forensic Science Center.

4.2 Software

There is a wide array of digital forensic software products on the market today. Some are general tools that serve a variety of functions. Others are more focused, serving a fairly limited purpose. These applications tend to focus on a very specific type of evidence, e-mail or Internet, for example. When selecting software, a choice needs to be made between going with open source tools or a commercially produced product. There are advantages and disadvantages to both. Factors such as cost, functionality, capabilities, and support are some of the criteria that can be used to make this decision.

One of the more popular open source tools is **SANS Investigative Forensic Toolkit (SIFT)**. SIFT Workstation is a powerful, free, open source tool. It's built on the Linux Ubuntu operating system. **This tool is capable** of file carving as well as analyzing file systems, web history, recycle bin, and more. It can also analyze network traffic and

volatile memory. It can also generate a timeline, which can be immensely helpful during an investigation.

As for commercial tools, two of the most popular general software tools are **Forensic Toolkit (FTK)** from AccessData and EnCase from Guidance Software. Both are excellent and can make exams easier and more efficient. These applications can perform a multitude of tasks, including:

- Searching
- E-mail analysis
- Sorting
- Reporting
- Password cracking

The search tools in these products are particularly powerful, and give examiners the capability to drill down to precisely the information they are looking for. Here is a quick list of some of the `:

- E-mail addresses
- Names
- Phone numbers
- Keywords
- Web addresses
- File types
- Date ranges

As helpful as these tools can be, they do have some limitations. The reality is that no single tool does it all. For that reason, budget permitting, labs need to have a variety of tools available.

More and more specialty tools are coming on the market. These tools focus on one aspect of digital evidence such as e-mail or web-based evidence. These can bring some additional capabilities to the table that some multipurpose tools don't.

5. Types of Digital Forensics

As described above, digital forensics is—essentially, the science of gathering evidence from digital devices. That said, there are so many different digital devices and media available today that the science of computer forensics can be divided According to forensic target or digital devices involved in an investigation.

Digital forensics includes several types, and following are some of its most well-known types:

5.1 File System Forensics: Data on a physical medium, such as a hard drive or flash drive, is organized, labeled, and governed by a file system (e.g. File Allocation Table FAT). File System Forensics is generally used for discovering the locations of files that are more useful as evidence than the file system itself.

5.2 **Memory forensics (i.e., RAM forensics):** Despite being called RAM forensics, this term actually refers to the application of forensic techniques on any/all volatile memory, which includes RAM, caches (of all levels), and registers. Memory forensics must be performed during live analysis, because the contents of volatile memory are permanently lost when the system is shut down.

5.3 Operating System Forensics: To perform operating system forensics, the investigator must have deep and thorough knowledge of multiple operating systems, as well as the ability to understand the meaning of logs generated by different operating systems.

5.4 Multimedia Forensics: Multimedia forensics refers to the application of computer forensics techniques on files that contain more audio/visual data than text, such as sound recordings, music files, videos, and pictures. There are many possible cases where multimedia files would be useful as evidence: Pirated music files, sound and video recordings of crimes, and illegal pornographic images, are all good examples.

5.5 Network Forensics: IP Tracing and Network Traffic Monitoring are the major components of Network Forensics. The main objective is to look for evidence of illegal activities that involve a transfer of files or information. It is important to note that most applications of Network Forensics make use of the Internet.

The analysis of social media accounts could be considered a combination of Network and Multimedia Forensics, depending on which techniques are used.

5.6 **Database Forensics:** Databases are, understandably, full of different types of information. The data can be investigated for its malicious uses, or to determine how/whether some legitimate data was stolen or deleted. Sometimes, the database itself is valuable information as well as the relations between tables in the database

can reveal important details of how, for example, a criminal organization, is structured.

5.7 **Mobile Device Forensics:** There are also many different types of mobile devices: smart phones, GPSs, Personal Digital Assistants (PDAs), and digital cameras, and all of them use different operating systems and have different capabilities, storing different types of data.

A ***mobile phone*** might contain taped conversations, digital pictures, texts and emails, contact lists, and sometimes even digital video recordings.

The goal of ***GPS*** forensics on the other hand is to look for information like waypoints, directions, routes, favorites, etc., in order to figure out the travel patterns of a suspect.

5.8 E-mail Forensics : Emails are just as useful to forensic investigators. A lot of information can be found in even the most ordinary emails. However, as they can be analyzed to discover details about the sender and his/her motives, and even submitted as court evidence.

5.9 **Financial Forensics**: Financial criminal activities such as health care fraud, financial institution fraud, mortgage fraud, insurance fraud, mass marketing fraud, and money laundering, are increasing during the past year. Today, as digital media is used to store extreme amounts of financial information in multiple and complex systems, trends in fraudulent activities have changed from the traditional forgeries in accounting books and receipts, to new frauds like modifying financial files and deleting or altering important data.

6. CLONING

Forensic clone is a “bit stream” copy of a particular piece of media such as a hard drive. A forensic clone is an ***exact***, bit for bit copy of a hard drive. It’s also known as a bit stream image. In other words, every bit (1 or 0) is duplicated on a separate, forensically clean piece of media, such as a hard drive. Why go to all that trouble? Why not just copy and paste the files? The reasons are significant. **First**, copying and pasting only gets the ***active data***. That is, data that is ***accessible to the user***, these are the files and folders that users interact with, such as a Microsoft Word document. **Second**, it does NOT get the data in the unallocated space, including ***deleted and partially overwritten***

files. **Third**, it doesn't capture the ***file system data***. All of this would result in an ineffective and incomplete forensic exam.

6.1 HASHING

How do we know our clone is an exact duplicate of the evidence drive? The answer comes in the form of a hash value. A hash is a unique value generated by a cryptographic hashing algorithm. Hash values (functions) are used in a variety of ways including cryptography and evidence integrity. Hash values are commonly referred to as a "digital fingerprint" or "digital DNA" Any change to the hard drive, even by a single bit, will result in a radically different hash value. Therefore, any tampering or manipulation of the evidence is readily detectable.

6.2 Uses of Hashing

Hash values can be used throughout the digital forensic process. They can be used after the cloning process to **verify** that the clone is indeed an exact duplicate. They can also be used as an **integrity check** at any point it is needed. Examiners often have to exchange forensic images with the examiner on the opposing side. A hash value is sent along with the image so that it can be **compared** with the original. This comparison verifies that the image is a bit for bit copy of the original. The relevant hash values that were generated and recorded throughout the case should be kept and included with the final report. These digital fingerprints are crucial to demonstrating the integrity of the evidence and ultimately getting them before the jury.

7. Documenting and Collecting Evidence

- Treat every piece of evidence as it will be used in court.
    - Goal is to recover as much evidence without altering the crime scene
- If the device containing the evidence is a cell phone, you will need to isolate the phone from the network signal to prevent evidence from being destroyed.
- Document the scene: Photographs are an excellent way to document the evidence and the scene. You will photograph the entire scene (e.g., the entire room, not just the computer on the desk).
- Preservation of the evidence is critical :
    - Must not alter data or changing files in anyway.

- Always work on copies, never the original.
- Never changing time or date stamps
- Can't just boot up or will alter HDD (Overwriting unallocated disk space can happen when re-booting).
- Remove HDD and place in forensic computer.

8. Evidence Storage

When the evidence is not actively being examined, it must be stored in a secure location with limited access. One of the best solutions is a **data safe**. These safes come in multiple sizes and are specifically designed to protect digital evidence from **theft and fire**. Some types of digital media are very vulnerable to heat (tape, for example). A data safe is able to keep the media at an acceptable temperature long enough (hopefully) for the fire to be extinguished.

Evidence storage locations must be **kept locked at all times** when not actively being used. A log or audit trail should also be maintained detailing who entered, when they entered, and what they removed or returned.

Access to evidence storage and other sensitive areas can be controlled by a variety of means including pass codes and key cards. Electronic controls have some distinct advantages over keys. One significant advantage is the ability to log each and every time an individual accesses a restricted area. This audit trail can be very helpful in monitoring and verifying the chain of custody.

9. Computer Forensic Capabilities

- Recover deleted files
  - Deleted data: Some deleted files may be recovered.
  - Reveal hidden data.
  - Extract encrypted data.
- Find out what external devices have been attached and what users accessed them
- Determine what programs run
- Recover web pages
- Recover emails and users who read them

- Recover chat logs
- Determine file servers used
- Discover document's hidden history
- Recover phone records and SMS text messages from mobile devices
- Find malware البرمجيات الخبيثة and data collected

10. Current Day Cyber Crime

- Identity Theft
- Theft of Company Secrets (documents, customer or employee lists)
- Phishing Masquerading التنكر والتصيد
- Embezzlement (money or information) الأختلاس
- E-mail Fraud/Spam
- Anonymous Slandering التشهير
- Harassment تحرش ، مضايقة
- Spy ware / Virus / Malware/ Bots
- Bandwidth theft
- Financial Crime
- Child Pornography
- Piracy
- Credit Card Fraud

11. Breaking Passwords

Passwords

A string of characters that security systems use to authenticate, or verify, a user's identity. Security systems compare passwords a user provides during login to stored values for the user account. If the value provided (password) matches the stored value, the security subsystem authenticates the user. Most operating systems store passwords when users create login accounts.

There are two main reasons for investigators to crack passwords. First, you may need a password to log in to a computer or access a resource. Second, you may need a password or key to access encrypted data that may be vital to the success of the investigation.

Password discovery is similar to Key discovery. Forensic investigators need to find, deduce, or crack the encryption to get to the key. The biggest difference between cracking passwords and cracking encryption keys is that cracking encryption keys is usually much harder and takes far longer. The simple explanation is that the plaintext for a password is generally limited to a couple dozen characters. The plaintext for a file could be gigabytes. Cracking the encryption key takes substantially longer than cracking a password.

Breaking passwords, or cryptanalysis, can be daunting or practically impossible. In order to give us the best chance for success, we'll need to use any advantage we can get. There are multiple ways to break passwords; some are technical, some are not. Sometimes it's as simple as asking. Options include brute force attacks, dictionary attacks, and resetting passwords. They can all yield positive results. We'll dig into these attacks more in an upcoming section.

The good news is that, in most cases, we are still dealing with people, and they represent the weakest point in this entire process. Humans can be both lazy and careless, giving us the chance we need to crack the encryption. Far too many people use simple passwords that are easy to break. Some of the best include "password", or the ever-popular "1234", Birthdays, pet names, or the name of our favorite sports team are also used routinely. Memorizing long random passwords is not easy or convenient for the majority of us. Even if a strong password is used, oftentimes it is written down on a Post-It note and stuck to the monitor. Furthermore, encryption keys can be left unsecured and subject to compromise.

Capturing the RAM of a running machine can also help in breaking passwords. You've probably entered a password on a web site at one time or another. As you entered your password, dots appeared, concealing the text as you type. What you may not realize is that the actual password is recorded in RAM. Failing to grab the RAM from a running machine could truly be a missed opportunity.

What exactly qualifies as a strong password? According to Microsoft, a strong password uses a variety of letters, numbers, punctuation, and symbols, and has a minimum length of fourteen characters.

11.1 Brute Force Attacks

On the other end of the spectrum is the brute force attack. A brute force attack simply attempts every possible password combination until it finds a match. If the utility attempts to use every possible combination, it will eventually succeed. However, the amount of time required depends on the complexity of the password. The longer the password, the more time it will take to crack.

Brute force attacks should never be your primary method for cracking passwords for two reasons. First, brute force attacks are slow. They can take a substantial amount of investigative time. Second, the length of the password may not be known. In this case, the utility will have to try many, many combinations that won't succeed before finding the right one.

11.2 Dictionary Attack

A dictionary attack is the simplest and fastest attack. The cracking utility uses potential passwords from a predefined list of commonly used passwords. The password dictionary stores the list of passwords. The larger the dictionary, the higher the probability the utility will succeed (but the longer it will take to attempt the entire dictionary file).

Dictionary attack: An attack that tries different passwords defined in a list, or database, of password candidates.

The reason this type of attack works so well lies in human nature. People tend to use common, easy-to-remember passwords. Most would be surprised to find their favorite password in a password dictionary. Any passwords found in a password dictionary are too weak and should be changed.

A dictionary attack is more precise, using words and phrases that can be collected from **multiple sources**. For example, the dictionary file is made up of every word found on the suspect hard drive .Other dictionary sources could be terms commonly used in certain criminal circles such robbery or narcotics trafficking. Dictionaries can also contain words from specific sources such as web sites.

Gathering information about the terms and words associated with these interests could provide clues to the suspect's password. We want to know the name of our subject's children and pets. We want to know their hobbies and interests, the movie they follow, etc. This information can be used to build a dictionary of potential passwords.

A number Password Recovery **Toolkit** are available in market. For example we can enter a total of seven words of information including names, birth date, and some keywords related to the suspect. From the seven words, the toolkit then generates over twenty-six hundred permutations.

11.3 Hybrid Attack

The final type of attack, the hybrid attack, combines the dictionary and brute force attack methods. In a hybrid attack, the utility starts with a dictionary entry and tries various alternative combinations. For example, if the dictionary entry were "lord," the hybrid attack utility would look for these possible alternatives:

- Lord
- L0rd
- lord
- l0rd

Hybrid attack: A modification of the dictionary attack that tries different permutations of each dictionary entry.

Regardless of the type of utility used, there is Toolkit available that can help you get the passwords you need to access evidence.