# IT&C Security Master

- www.ism.ase.ro -

Academy of Economic Studies

Bucharest, 2022

presentation

# Multimedia Security

Assoc. prof. Mihai DOINEA, Ph.D.

mihai.doinea@ie.ase.ro

Mihai Doinea

Digitally signed by Mihai Doinea
Date: 2022.05.08 16:12:36 +03'00'

# **Agenda** for presentation

1. Course Management & Overview
2. Multimedia Security Aspects
3. Hands-On Practice

# 1

## Structure, Organization Units, Evaluation

# **Course** Management & Overview

# 1. Course Management & Overview

## Multimedia Security

- **Website:**

**www.ism.ase.ro** - Sakai Platform

- **Period:**

**May 07$^{th}$ and 08$^{th}$, 2022**
**May 14$^{th}$ and 15$^{th}$, 2022**

- **Objective:**

**A broader image upon multimedia content and how users can access byte level content in order to deal with security issues.**

## Multimedia Security – Evaluation

- **When ? June 15th , 2022**
- **Where ? in campus**
- **How ? on SAKAI - Online Web Learning Platform / oral evaluation**
- **What ? Questions on Multimedia Security, Multiple Answers / Multiple Choices theoretical and practical questions / Oral evaluation based on the quiz results**
- **How long ? 45 min. quiz;**

# Multimedia Security – Evaluation

- 40% - Practical Evaluation
  - 20% - E1(07.05) and E2(08.05);
  - 20% - P1(14.05) or P2(15.05);
- 60% - QBOE

# Multimedia Security – Course Agenda

1. **Image Standards**
   - BMP – internal representation, technical manipulation, image processing algorithms
2. **Audio Formats**
   - RIFF / WAVE – byte level access, audio representation, audio processing algorithms
3. **Video Content**
   - AVI file format – frame access, video representation
4. **Security concerns**
   - Steganography, digital watermarking, steganalysis, multimedia extraction and interpretation, security characteristics, digital rights management, content digital signature

# Multimedia Security – Bibliography


[1] ***Perspectives on Multimedia: Communication, Media and Information Technology*** – R. Burnett, A. Brunstrom, A. Nilsson, Editura Wiley, 2004, 250 pg., ISBN 9780470868638

[2] ***Multimedia: concepte si practica*** – I. Smeureanu, G. Drula, Ed. Cison, 1997, ISBN 973-96370-8-6

[3] ***Multimedia: Making it Work*** – T. Vaughan, Sixth Edition, McGraw-Hill Publisher, 2004

[4] ***Intelligent Multimedia Communication: Techniques and Applications*** – C. We Chen, Z. Li, S. Lian, Editura Springer, 2011, 300 pg., ISBN 9783642116858

[5] ***Multimedia Content and the Semantic Web: Standards, Methods and Tools*** – G. Stamou, S. Kollias, Editura Wiley, 2005, 414 pg., ISBN 9780470857533

[6] ***Compression for Multimedia*** – I. Bocharova, Cambridge University Press, 2010, 280 pg., ISBN 9780521114325

# Section Conclusions

Security aspects applied in multimedia field is needed because of a high degree of malware found in any branch of digital content.

2

Images, Sounds, Videos, Digital Signature, Steganography, Watermarking, DRM

# Multimedia Security Aspects

# 2.1 Security in Multimedia

What is **M**ultimedia **S**ecurity – **MS** ?

A combination of text, digital graphics, sound, animation and video powered by a computer or by any electronic or digital means.

An instrument used to protect against unwanted or accidental actions of malicious users.

# 2.1 History

- **1455** – Printing Press Gutenberg and Caxton
- **1780** – Discover the electricity
- **1833** – The precursor of computer, first analytical machine
- **1837** – Telegraph, Morse code
- **1839** – First photograph using negative paper
- **1858** – Transatlantic cables
- **1867** – Remington typewriter
- **1928** – Steamboat Willie, first Disney cartoon
- **1931** – First calculator – Conrad Zuse
- **1940** – First color TV broadcast
- **1946** – ENIAC - Electronic Numerator Integrator and Calculator – first digital computer

# 2.1 Security in Multimedia

- 1959 – Transistors replace vacuum tubes;
- 1965 – the optical fiber standards;
- **1970 –** birth of The Internet;
- 1971 – first email sent;
- 1977 – Apple II, first PC with colour interface;
- 1981 – first IBM PC;
- 1983 – the CD (compact disc) is announced;
- 1984 – ARPANET, the actual version Internet;
- **1985 –** Commodore Amiga, the first multimedia PC;
- 1985 – CD-ROM (CD read-only memory) is released;
- **1986 –** *The Academic American Encyclopedia* on CD;

# 2.1 Security in Multimedia

- **1988** – Macromind launches Director, an instrument for *multimedia authoring*.
- **1991** – World Wide Web is launched;
- **1991** – MP3 format is created;
- **1993** – Mosaic, first Graphic Web browser;
- **1995** – Disney launches *Toy Story*, first movie created entirely on computer (77 minutes = 4 years + 800,000 hours for graphics processing);
- 1995 – Internet becomes broader;

# 2.1 Security in Multimedia

**1996**
digital camera is open to the public;

**1997**
DVD is released;

**1998**
the first MP3 player;

**2000**
Mobile phones take off;

**2003**
Digital cameras become cheap;

**2003**
Internet Wikipedia;

**2005**
Image sharing – Flickr

**2006**
Video sharing – YouTube and Blue-ray Disc

**2007**
Web 2.0 is recognized

# 2.1 Security in Multimedia

**2010**

first 3D movies
ToyStory 3D

**2011–2015**

microdisplay and mobile projection display, see through microdisplay, mobile flexible display

**2015 – present**

Smart Things Era; Augmented and Virtual Reality

**2016 – present**

Web 3.0

**2018**

metal printing; more IoT

**2019 - present**

eSIM / iSIM revolution in IoT

# 2.1 Security in Multimedia

**By target:**

- Interactive (PTP – person to person)
  - Voice (voice)
  - Text (sms)
  - Image( mms)
  - Video (p2p calls)
  - Complex (social networking)
- Non-interactive (PTC – person to content)
  - text / pictures
  - internet
  - photos
  - video, music, movies

# 2.1 Security in Multimedia

**By domain:**

- Training (educational)
- Medical (Tomography, Echograph, RMN)
- Industrial (Simulation of a technological industrial flow)
- Geographic (Digital Maps)
- Services, Sales, Advertising

**By destination and interactivity:**

- Public and personal interest
- Local and remote
- Videoconference

# 2.1 Security in Multimedia

**Peripheral devices and hardware components:**

- Image: scanner, digital camera, film scanner, mobile device;
- Sound: sound board (analogous to digital convertor), microphone;
- Video: capture board, video camera, TV tuner;

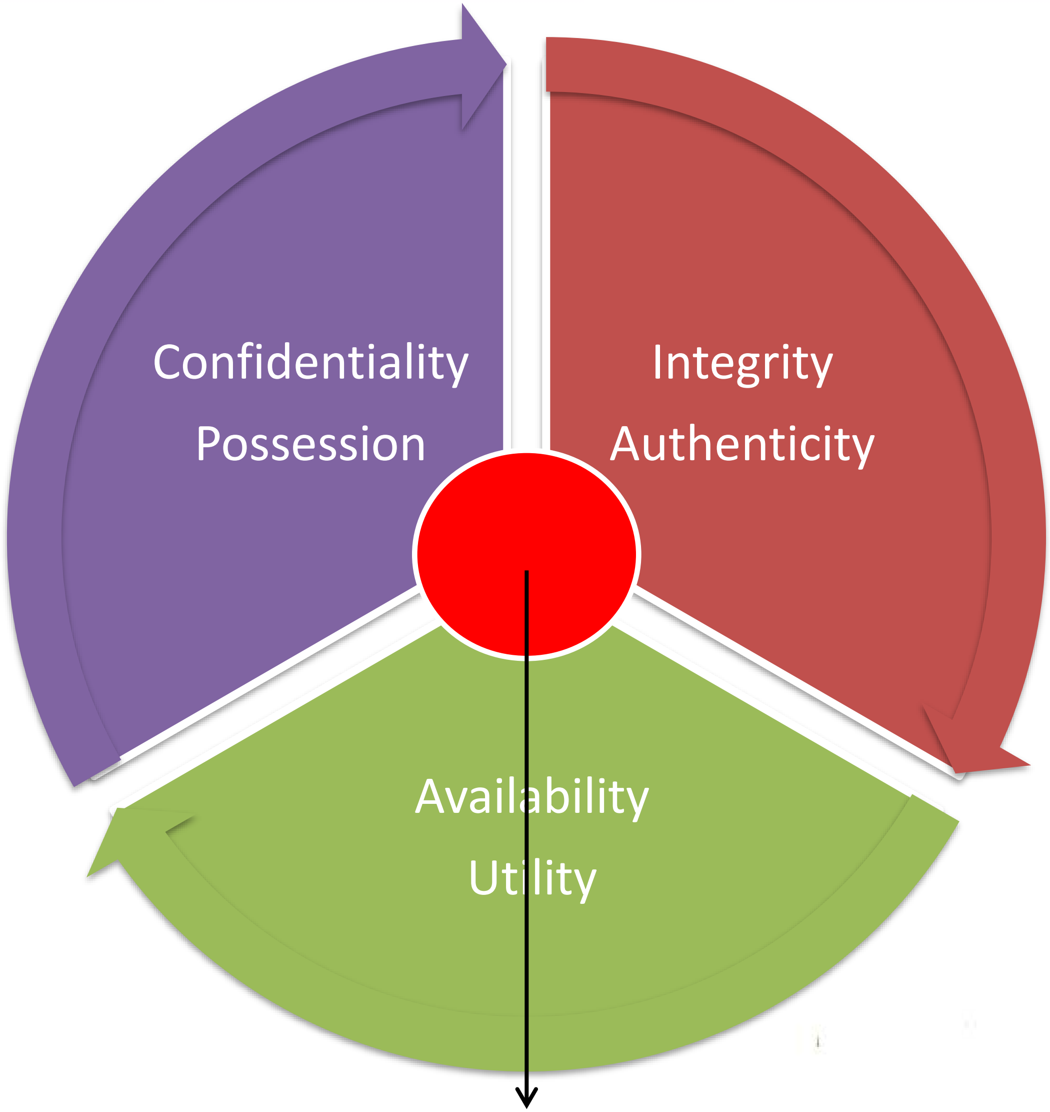
# 2.1 Security in Multimedia

**Drivers to access multimedia hardware components;**

**Specialized software for multimedia processing;**

**Specialized software for integrating security aspects;**

# 2.1 Security in Multimedia




**Multimedia Security Aspects**

# 2.1 Security in Multimedia

**Why do we need MS for ?**

- **Watermarking and authenticity**
  - Digital watermarking
  - Image self-restoration
- **Steganography and confidentiality**
  - Embedding messages in multimedia files
  - Identifying altered digital content
- **Multimedia forensic and integrity**
  - Image and Video forensic
- **Usable Security**
  - CAPTCHA, Graphical Passwords, FIDO
- **Biometrics**
  - Fingerprint or Face recognition, pattern recognition

## Is what you expected?

This was **Section 2.1 – Security in Multimedia**

**For non-repudiation:**

Digital Signature

**For protection:**

Digital Rights Management

**For confidentiality:**

Steganography

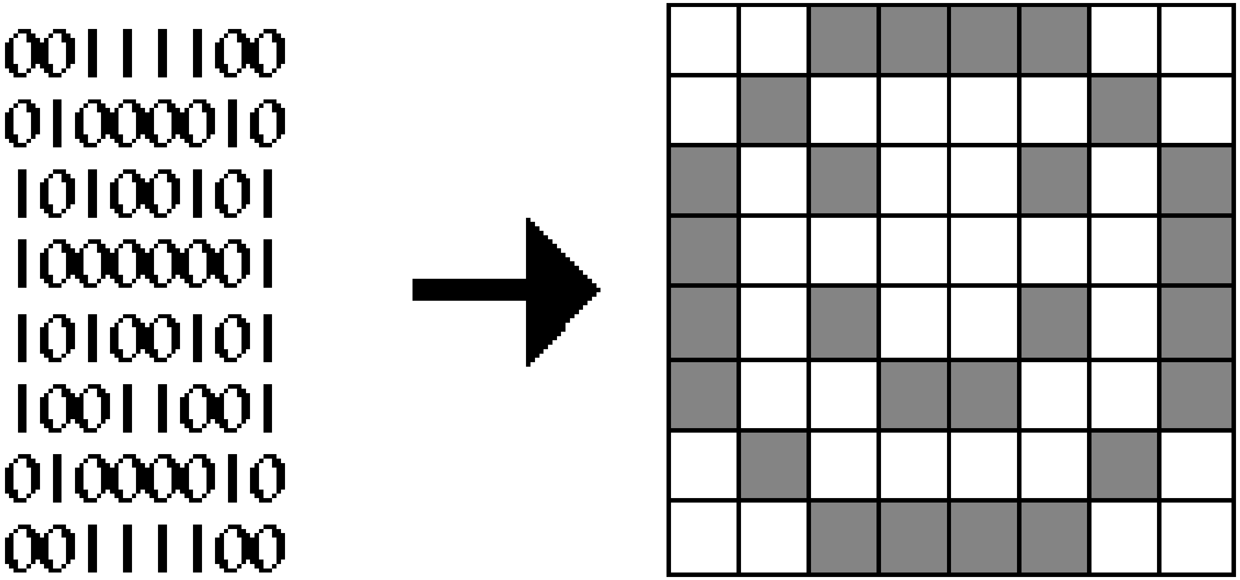
**For authenticity:**

Watermarking

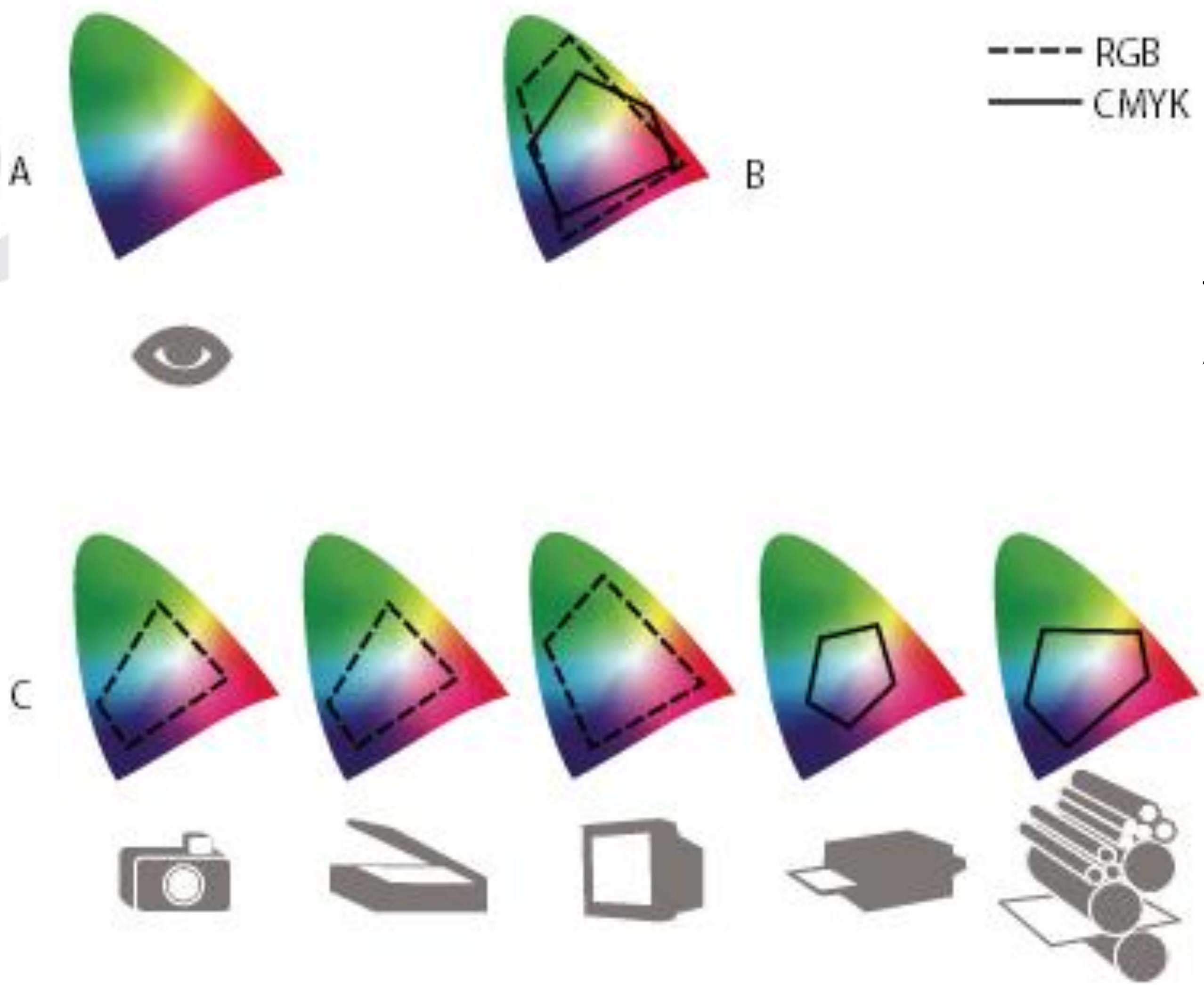
**For integrity:**

Watermarking and Steganography

# 2.2 Images


00111100
01000010
10100101
10000001
10100101
10011001
01000010
00111100

# Image



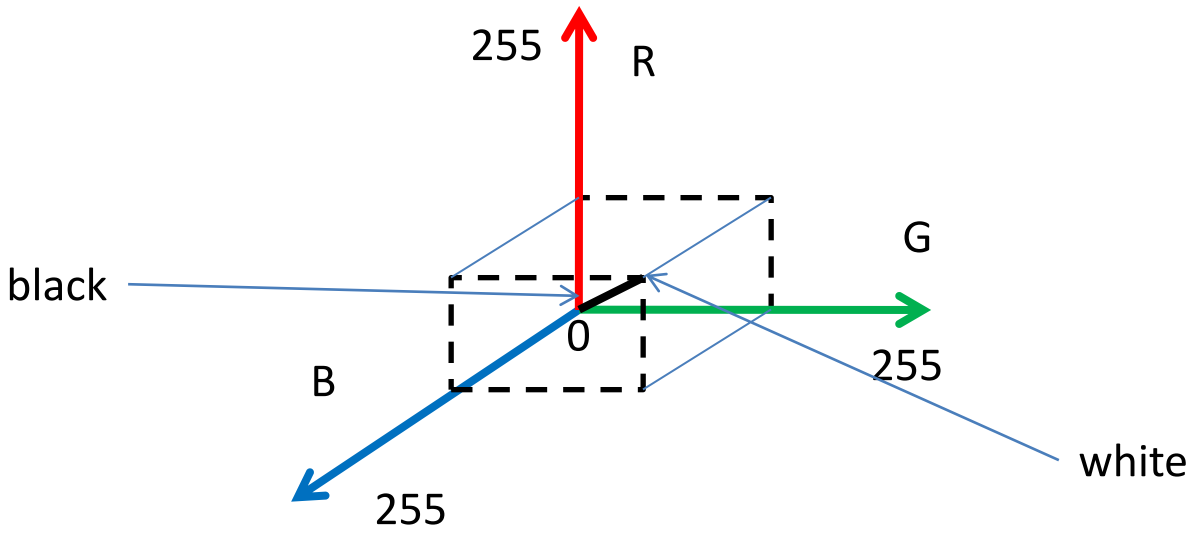

Types of color palettes:
A – CIE Lab
B – Digital images
C – Devices

Adobe Photoshop Documentation

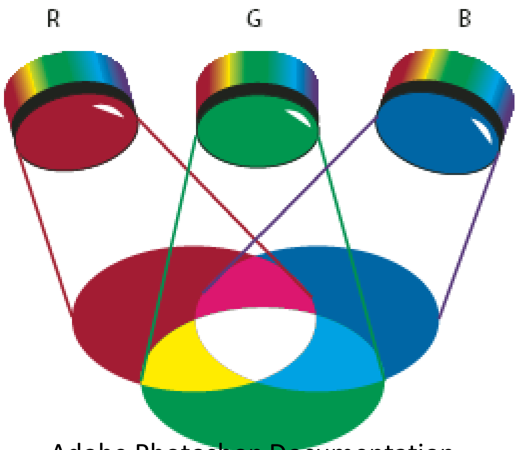
# Image

RGB

- uses 3 main colors: Red, Green, Blue
- each color with an intensity between 0 and 255
- for equal values: R=G=B, gray tones are created



# Image



Adobe Photoshop Documentation

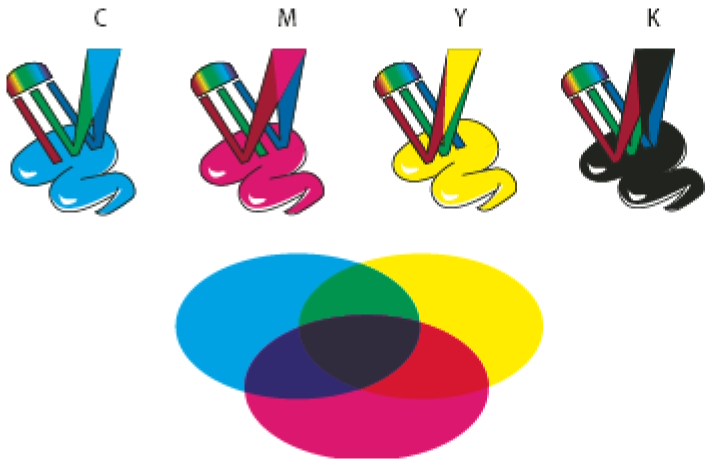
**RGB**

- additive colors
- the absence of the 3 main colors generates black (0 for each color)
- the presence of all 3 colors at maximum intensity generates white
- different color pallets: Adobe RGB, sRGB, ProPhoto RGB
- used for displaying devices

# Image

**CMYK**

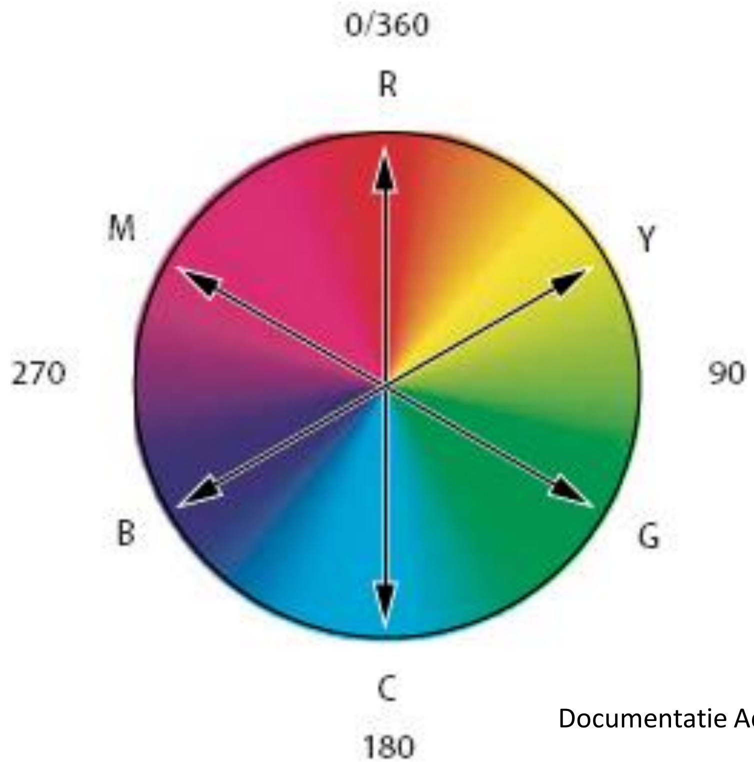
- Uses four based colors (Cyan, Magenta, Yellow, Black)
- Each color with an intensity between 0% and 100%
- Are subtractive colors
- Used for printing devices




Adobe Photoshop Documentation

# Image

RGB vs CMYK



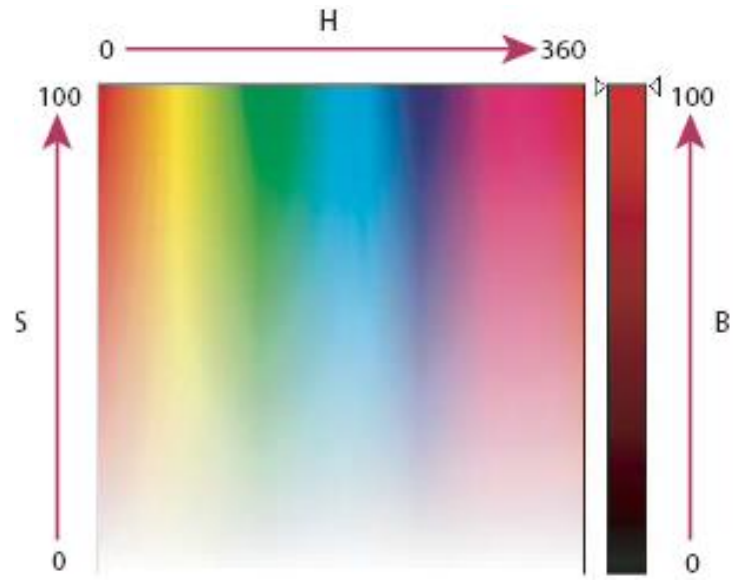

Documentatie Adobe Photoshop

# Image

HSB/HSL

- Model based on the Hue, Saturation, Brightness / Lightness properties
- Based on the human perception of colors

Adobe Photoshop Documentation




**HSB Model**

H – Hue
S – Saturation
B - Brightness

# Image

Lab Color

- defined also as CIE L*a*b (Lightness, a – green-red component, b – blue-yellow component);
- L is between 0 and 100, a and b are between +127 and – 128;
- based on the human perception of colors
- the numeric values from this model cover all the colors perceptible by the human eye;
- it’s a device independent model;

# Image

Grayscale

- Defines grayscale tones;
- In an image with a 8 bits depth there are 256 gray tones;
- Gray tones can be represented as black percentages (0% - 100%)
- Each gray pixel has a intensity between 0, black and 255 white;

# Image

Web-safe

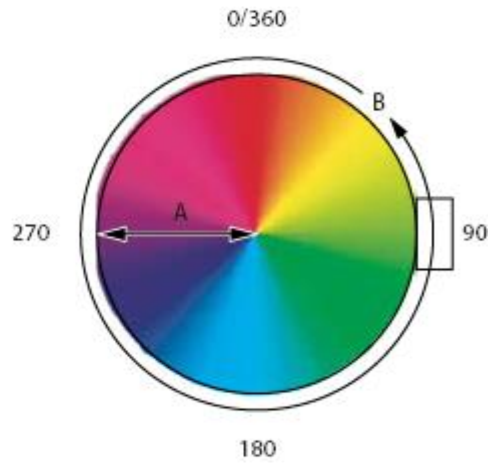
- A limited set of colors derived from RGB
- Uses 256 colors;
- The color set is used by all browser;

# Image

- Hue
- Brightness
- Saturation
- Contrast
- Color balance

# Image

Hue is the level of color reflected by an object and transmitted to the human eye;




Color pallets:
A – Saturation
B – Hue

Adobe Photoshop Documentation

# Image

## 01

### Saturation

- Describe the intensity of a color;
- Represents the level of gray color correlated with the hue, with 0% (gray) 100% (maximum intensity)

## 02

### Brightness

- High and low intensity measured from 0 % (black) to 100% (white)

# Image

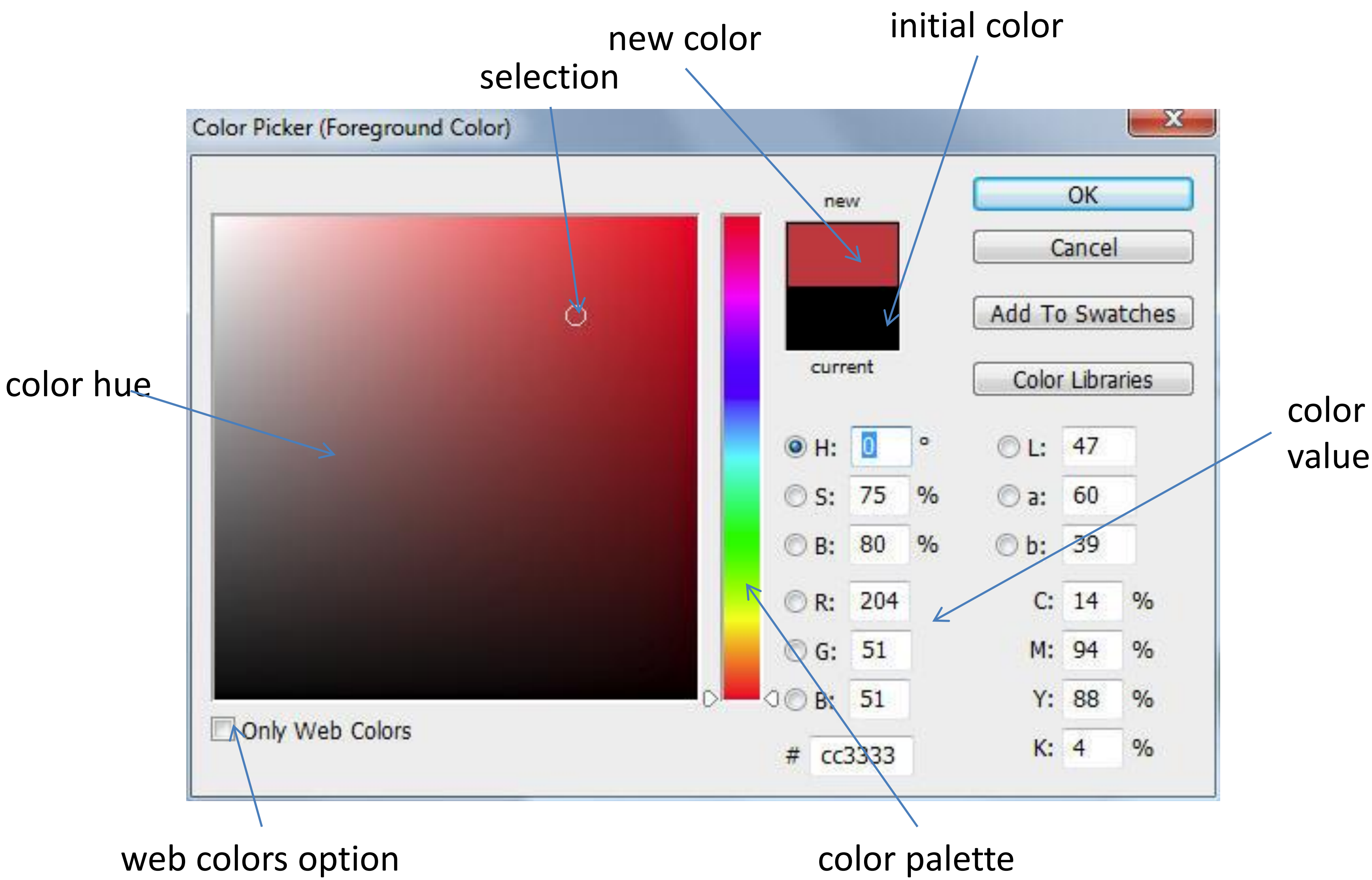
Contrast

- Measures the difference between brightest and darkest areas

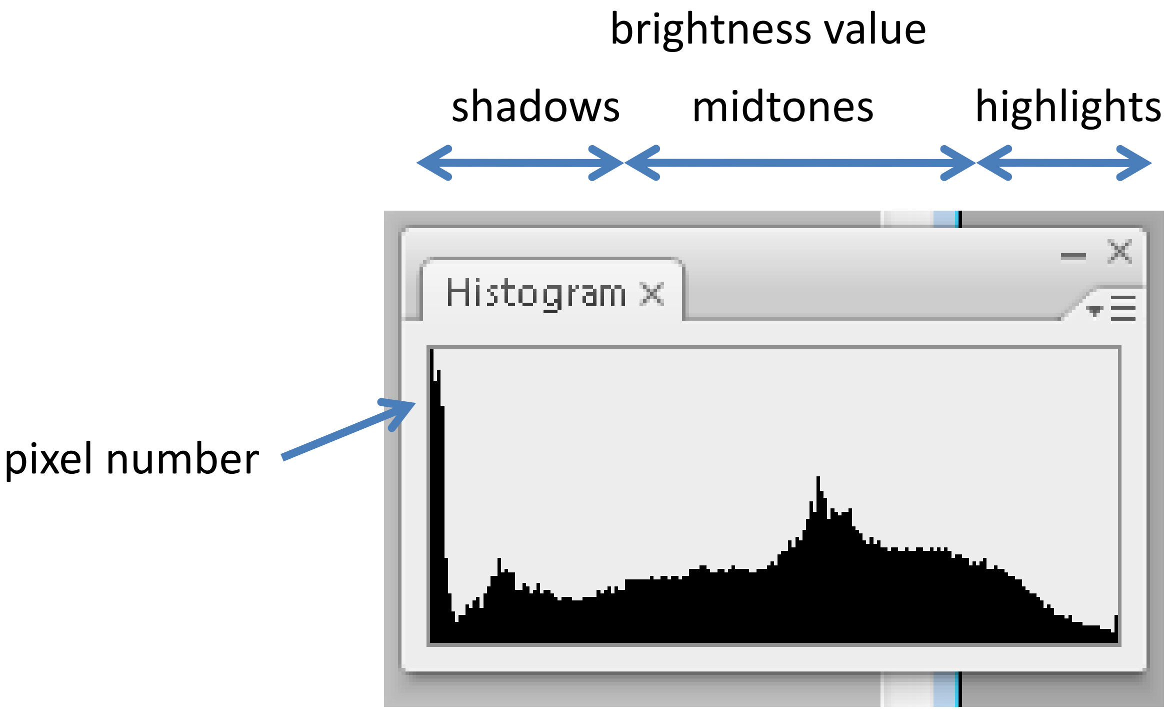
Color balance

- Analyses the color pallet distribution
- The predominant color influences all other colors that are present in the picture

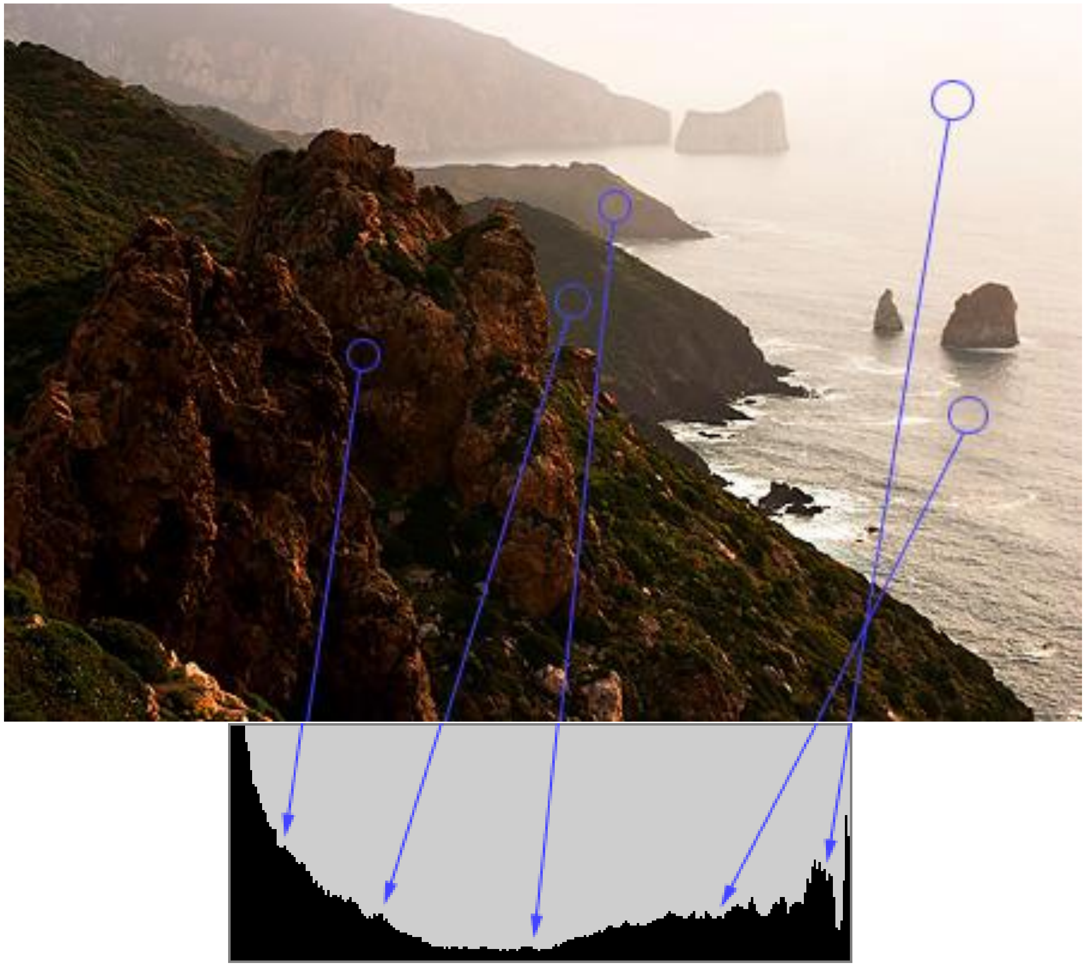
# Image


new color
initial color
selection
Color Picker (Foreground Color)
new
current
OK
Cancel
Add To Swatches
Color Libraries
color hue
color value
H: 0 °
S: 75 %
B: 80 %
R: 204
G: 51
B: 51
# cc3333
L: 47
a: 60
b: 39
C: 14 %
M: 94 %
Y: 88 %
K: 4 %
Only Web Colors
web colors option
color palette

# Image histogram



# Image




Image

# Image

## Types of images

- grid of pixels (rasterized)
- vector format

## Compression algorithms

- RLE
- Entropy Encoding

# Image - Bitmap

**Image Compression / Decompression**

- **Algorithms:**
  - with loss of information, lossy compression;
  - without loss of information, lossless compression;

  - symmetric (compression time = decompression time)
  - asymmetric (the compression process takes longer than decompression)

# Image - Bitmap

| Algorithms | Data | Category |
|---|---|---|
| RLE, Huffman | Text, image | lossless |
| JPEG | Images | lossy |
| Fractal based | Images | lossy |
| DVI | Video | lossy |
| MPEG | Video | lossy |

# Image - Bitmap

**Image Compression / Decompression**

- *HOFFMAN*
  - variable coding
  - uses the frequency of symbols
  - uses a binary unbalanced tree
- *RLE(Run Length Encoding)**
  - good for data with low diversity and high repeatability;
  - efficient when a small number of colors is used;

# Image - Bitmap

**Image Compression / Decompression**

- *RGB555*
  - reduces the total number of bits used for storing a color from 8 to 5;
- *JPEG*
  - Hybrid compression algorithm using 2 techniques
    - cosinus discret transformation (TCD)
    - Huffman coding
    - a good compression rate of almost 15:1
- *MJPEG(Motion JPEG)*
  - Animation standard;

# Image

**Bitmap format:**

- is a simple matrix, each element representing the color of a pixel;
- the image has volume;
- the image quality is dependent on the visualization scale;
- the image cannot adapt to a variable visualization scale;
- as higher the compression rate is as lower the image quality is;

# Image

**Bitmap format:**

- Digital equipment used for processing bitmap images:
  - – Scanner (DPI – Dots Per Inch)
  - – Movie scanners
  - – Digital camera (JPG, Raw)
  - – Web camera

# Image

**Bitmap format:**

- Bitmap file storing formats:
  - BMP
  - ICO (icons – 32 X 32)
  - TIFF (Tag Image File Format)
  - DIB ( Device Independent Bitmap)
  - DDB ( Device Dependent Bitmap)
  - JPG (Joint Photographic Experts Group, a bitmap compressed format with JPEG algorithm)
  - GIF ( a reduced file format used for transferring bitmap images)
  - PNG (Portable Network Graphics**)**

http://en.wikipedia.org/wiki/Comparison_of_graphics_file_formats

# Bitmap

- A bitmap is an array of bits that specify the color of each pixel in a rectangular array of pixels.
- The number of bits devoted to an individual pixel determines the number of colors that can be assigned to that pixel.
- If each pixel is represented by 4 bits, then a given pixel can be assigned one of 16 different colors ($2^4 = 16$).

# Internal structure of a BMP file

- **BITMAPHEADER**, 14 bytes, includes various data about the header of the file, such as:
  - **2 bytes**, the file signature, BM with value *4D42h*;
  - **4 bytes**, the file size;
  - **4 bytes,** reserved area;
  - **4 bytes,** the offset at which actual data begin;

# Internal structure of a BMP file

- **BITMAPINFOHEADER**, fixed length, 40 bytes:
  - **4 bytes**, the size of the info header region;
  - Image height **(4 bytes)** and image width **(4 bytes)**;
  - **2 bytes**, number of plans**;**
  - **2 bytes**, color depth, value given in number of bits per pixel;
  - **4 bytes**, compression, if it’s the case;
  - **4 bytes**, the total size of the image, if compression is applied;
  - **4 bytes**, horizontal resolution and **4 bytes**, vertical resolution;
  - **4 bytes**, the number of used colors and **4 bytes**, the number of important colors.

# Bitmap file structure

**OPTIONAL PALLETE** is used only if the bits per pixel value is lower or equal to 8;

**IMAGE DATA** is the area which contains each pixel value;

# Image - Bitmap

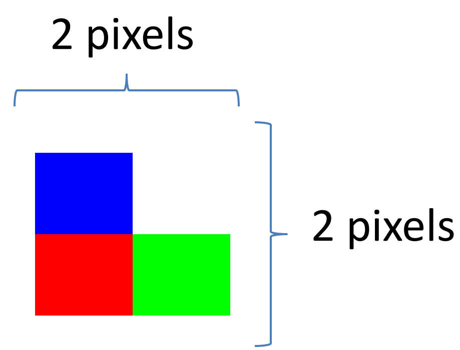
- stores digital images;
- **BMP** (Microsoft **Windows Bitmap)**
- is also known as DIB (device - independent bitmap)
- extensions: .bmp or .dib
- the content is uncompressed, being very large;
- the size of the file is directly dependent on the size of the image (height and width) but also on the depth of the image (bit depth or color depth)

# Color depth or bit depth:

- The number of bits used for pixel representation:
- Possible values equal to 1, 4, 8, 24 or 32 bits.
  - black and white images - 1 bit
  - 16 colors images - 4 bits
  - 256 colors images - 8 bits
  - images with $16,7 * 10^6$ colors (24 bits, one byte for each of the three colors in RGB representation)
  - images with transparency (RGB model + 1 byte for transparency) a total of 32 bits

# Image - Bitmap

| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 42 | 4D | 46 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 36 | 00 | 00 | 00 | 28 | 00 |
| 00000010 | 00 | 00 | 02 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 01 | 00 | 18 | 00 | 00 | 00 |
| 00000020 | 00 | 00 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 | FF | 00 | 00 | 00 | FF | 00 |
| 00000040 | 00 | FF | FF | FF | 00 | 00 | | | | | | | | | | |



# Image - Bitmap

| Structure | | Legend |
|---|---|---|
| **Bitmap Header** | | **14** Bytes area |
| | 1 | **2B**, file signature, *4D42h* |
| | 2 | **4B**, file size, *00000046h* |
| | 3 | **4B**, reserved area, *00000000h* |
| | 4 | **4B**, image data offset, *00000036h* |
| **Bitmap InfoHeader** | | **40** Bytes area |
| **Bitmap Data (pixels)** | | Pixels stored in RGB format / 1B per color |
| | 5 | **3B** pixel – Red color, *FF0000h* |
| | 6 | **3B** pixel – Green color, *00FF00h* |
| | 7 | **2B** padding after full line |
| | 8 | **3B** pixel – Blue color, *0000FFh* |
| | 9 | **3B** pixel – White color, *FFFFFFh* |
| | 10 | **2B** padding after full line |

# User defined types

The article is a collection of heterogeneous data stored in a compact region of memory. It allows users to bring together multiple variables in a single defined type entity.

```
struct  name {
 type    member_name;
};
```

# User defined types

- Bits structures allow bit access level;
- To access data at bit level programmers must:
  - define a structure which maps perfectly on the data which must be accessed;
  - load the pointer to the bit structure with the address of the respective data;
  - processing the bits without altering the meaning and internal representation of data;

# User defined types

- Bit fields:

```
struct name {
    type field_name        : width;
    int                    : width;
    …
    int   field_name       : width;
};
```

# User defined types

- The declared bit field has the following properties:
  - The type may be int, signed or unsigned;
  - The number of bits that a member can refer is specified by width;
  - If member name is missing, the number of bits specified in the width field can not be referred and accessed;
  - The fields are accessed the same way as in any other structure;
  - The address of fields can't be used;

# Bitwise operators

| | |
|---|---|
| & | • AND |
| \| | • OR |
| ^ | • OR exclusive - XOR |
| ~ | • One complement (flips bit value) |
| >> | • Right shifting |
| << | • Left shifting |

# Bitwise operators

| unsigned char x | x after operation | value of x |
|---|---|---|
| x=7 | 0000 0111 | 7 |
| x=x<<1 | ~~0~~0000 1110 | 14 |
| x=x<<3 | ~~000~~0111 0000 | 112 |
| x=x<<2 | ~~01~~1100 0000 | -64 |
| x=x>>1 | 1110 0000~~0~~ | -32 |
| x=x>>2 | 1111 1000~~00~~ | -8 |

# Bitwise operators

| & (AND) | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| \| (OR) | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

| ^ (XOR) | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

# Bitwise operators

```
  11000001
  01111111
& --------
  01000001
```

AND

```
  10000000
  00000011
| --------
  10000011
```

OR

```
  01111111
  01111000
^ --------
  00000111
```

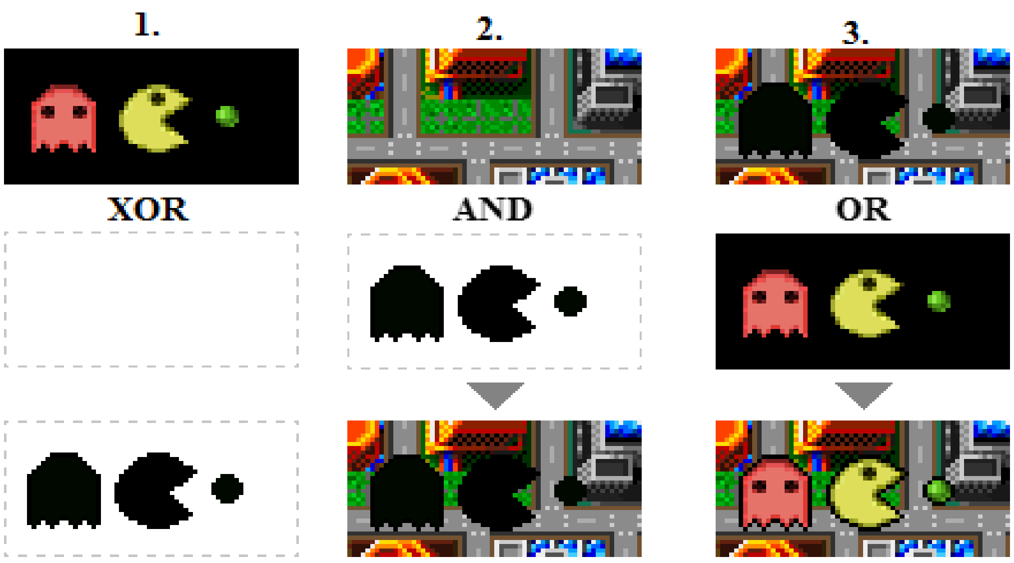
XOR

```
  00101100
~ 11010011
~ 00101100
```

NOT

# Bitwise operators

# Bitwise operators



# Image

Big picture example*

- President Barack Obama's Inaugural Address by David Bergman
  - elements: 220 images
  - dimension: 59,783 X 24,658 pixels
  - size: 1.47 gigapixels
  - http://gigapan.org/viewGigapan.php?id=15374

# Image - Bitmap

**VCX (PC PaintBrush File Format):**

- known as PaintBrush on Windows platforms;
- stores a 8 bit color image (256 colors),

  maximum dimension of 64000 * 64000 pixels
- the compression algorithm used is RLE

# Image - Bitmap

**TIFF** (Tag **Iinage** File **Format)**

- Known for storing and transferring of scanned images;
- Uses different compression algorithms and it’s very efficient: RLE,  LZW  (Lempel-Ziv-Welch) or JPEG.
- It platform independent => high level of portability

# Image - Bitmap

**ICO (Icon** Resource **File)**

- It is a *bitmap* format
- Stores small size images, used in Windows for representing pictograms
- Uses different resolutions and color pallets;
- It is easily obtained from normal images (http://www.favicon.co.uk/)

# Image - Bitmap

**JPG (Joint Photographics Experts Group**

- Stores *bitmap* images  in a compressed format
- Has a very good compression rate
- Is defined as being a standard in image compression and an image file format;

# Image - Bitmap

**GIF (Graphics Interchange Format)**

- is mostly used as a web based format;
- used for storing small size bitmap images;
- has high compression rates;
- developed by CompuServe for delivering image content easier on the web;
- uses the LZW compression algorithm.

# Image - Bitmap

**PNG (Portable Network Graphics)**

- It is an open format;
- Uses compression without loss of information;
- Used in web applications for image transfer;
- Created for replacing GIF format;

# Image – Vector format

- It deals with picture semantic (drawings → points and mathematical functions)
- Images have small sizes;
- Images are view scale independent;
- It can replace bitmap because not everything can resume to points and functions; eg: landscapes;
- The color and pixel position is determined by mathematical functions;

# Image – Vector format

- DXF (Drawing Exchange Format – for Autocad made by Autodesk)
- EPS (Encapsulated PostScript – Adobe format şi lb. PostScript)
- CGM (Computer Graphic Metafile) for transferring images across platforms;
- SVG (Simple Vector Graphics) defined for vector based web graphics

# Steganography Overview

- Security through obscurity
- Steganography historical facts
- Steganography overview
- Doing steganography
- Undoing steganography
- Steganalysis techniques

# Steganography

Steganography is the science of hiding the existence of messages embedded and passed from source to recipient into other files.

Steganography = steganos + graph = hidden writing

steganos = hidden

grapho = writing

# Steganography vs. Cryptography

| Similarities | Differences |
| --- | --- |
| hides messages | makes use of cover-objects |
| both uses keys | knows about the existence of a hiding message but not of its content |
| both uses carriers for transmitting the encoded message | the existence of the message and its content is hidden |

# Steganography principles

- Pure steganography
  - The principle used is similar to what "*Security through obscurity*" principle states, which is: the methods used in the encoding processes should not be public knowledge or hide anything about the encoding process.
- Key based steganography
  - uses public or secret key
  - Auguste Kerckhoffs's principle : "*A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.*"
- NIST security principle:
  - "*System security should not depend on the secrecy of the implementation or its components.*" is a more recent version of Kerckhoffs's doctrine.

# Main objective

Steganography

conceals information

in other, seemingly

innocent media.

# Tackled security issues

- transmitting information in a secure manner;
- uses cryptography for hiding the private key;
- protect embedded message from tampering;
- conceals the existence of the message inside the cover object;
- prevents the extraction of the hidden message;

# Types of Steganography

**Cryptographic aspects:**

- Pure steganography
- Key-based steganography

**Cover-object manifestations:**

- Digital steganography
- Physical steganography

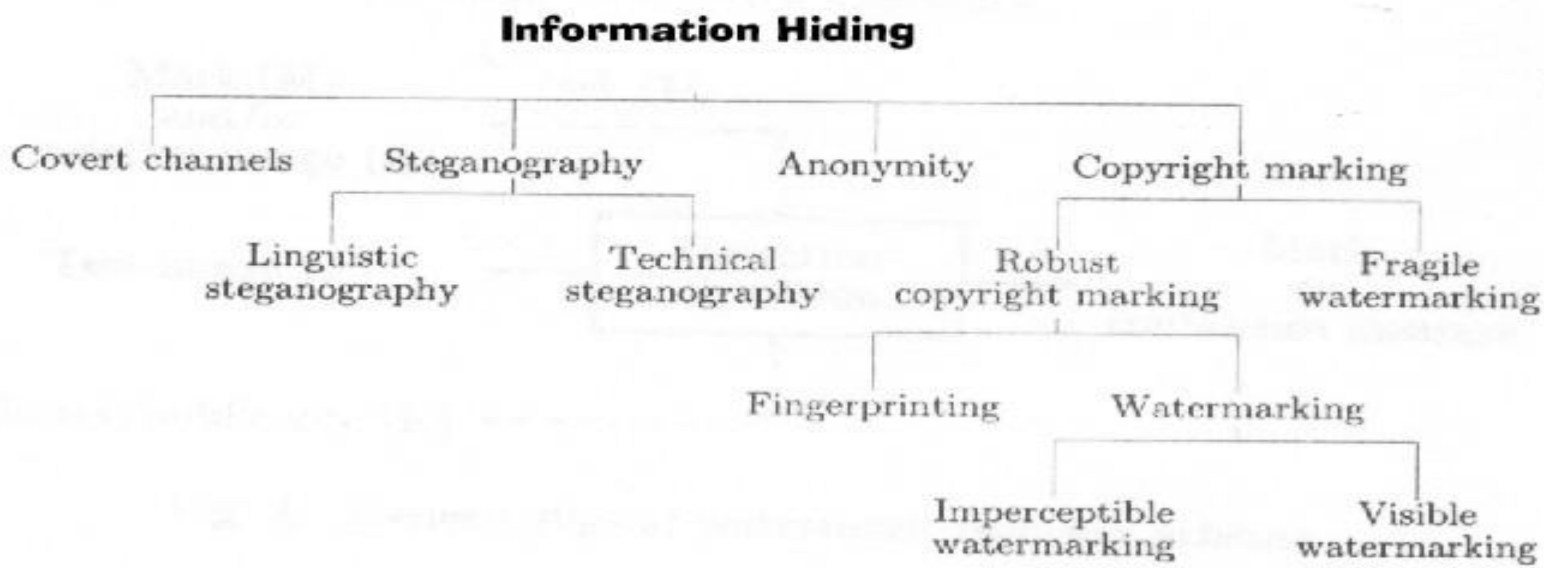
**Types of cover-object:**

- WLAN packages
- Steganophony
- Images
- Sounds
- Video

# Steganography historical facts

- Herodotus (484-425 BC) – security through obscurity
- Pliny the Elder (23 – 79 AD) – first invisible ink
- Renaissance Steganography
  - (1499 AD), Johannes Trithemius – first printed book on cryptology
  - (1665 AD), Steganographica, Gaspari Schotti
  - Giovanni Battista Porta – invisible ink on a hard boiled egg shell
  - 1870 – The Pigeon Post into Paris;
- Modern age
  - 1945 – the use of invisible ink in the second world war
  - 2001 – film industry, A beautiful mind

# Steganography




A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, "Information Hiding – A Survey", Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062-1078, July 1999

# Steganography vs. Watermarking

Steganography: techniques specific to a 1 to 1 communication;

Watermarking: techniques specific to a 1 to many communication;

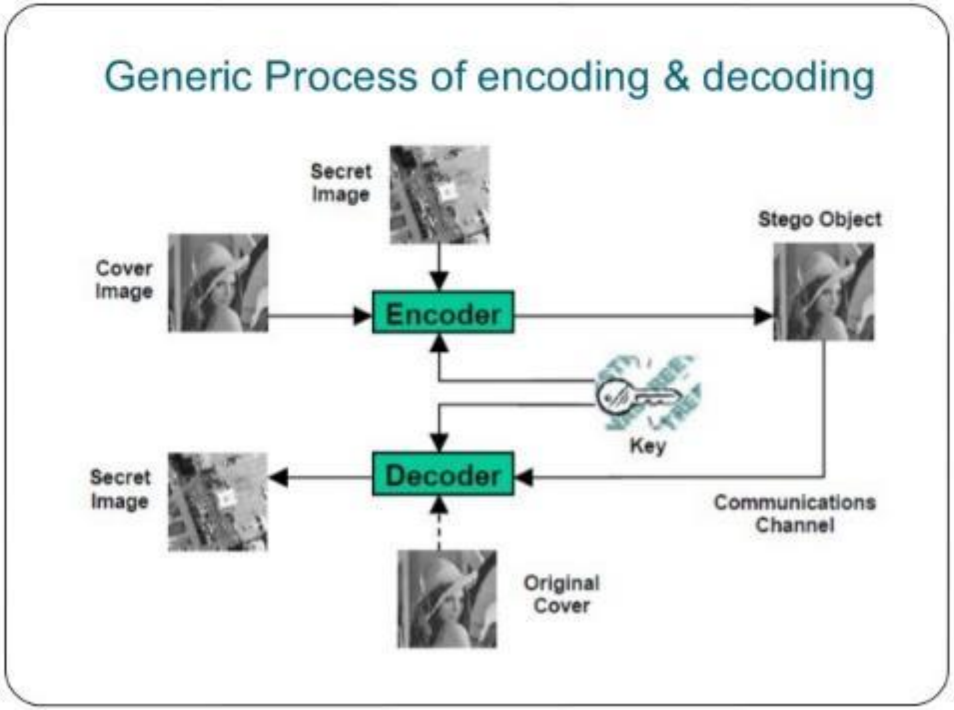
# Steganography elements

Involved components:

- Embedded message
- Cover-object
- Stega-key
- Encoding algorithm
- Stego object

Characteristics:

- Total capacity
- Robustness
- Undetectable
- Security



# Steganography

**Cover file** is the document in which data is hidden;

**Embedded file** is the document hidden into the cover file.

# Steganography

- Main objective:
  - Hiding data in other data;
- Secondary objectives:
  - Prevent the discovery of a message into a cover object
  - Prevent the extraction of an embedded message (robustness)
  - Prevent that the message will suffer as little damage as possible when altering the carrier

# Embedded message

- Must be of optimal length
- Must not include something related with the cover-file
- Must know and analyze the cover object in order to determine what type of carrier can be used
- Can be additionally encrypted to offer a better protection in case the message is found

# Cover file

**Images:**

- alteration of the least significant bit;
- applying masks and filters;
- image transformation algorithms;

**Sound:**

- encoding with the use of the LSB by substitution;
- encoding a message by using the parity bit;
- phase coding using the Discrete Fourier Transformation;
- spread spectrum using the frequency spectrum;

**Video:**

- conventional, image oriented; intra frame, spatial domain, transform domain;
- using compression techniques combined with steganography by using movement vectors or error prediction rate for hiding messages;
- spatial and temporal prediction;

# Steganography keys

- Can be private keys with the use of an symmetric encryption algorithms, such as: AES based on Rijndael cipher;
- Public key systems can also be used using asymmetric encryption algorithms, such as: RSA based on two large prime numbers;
- The keys must be of optimal length (256 for AES, 2048 for RSA);

# Steganography approaches

- Statistical formulation:
  - False Positive $P_{fp}$ = P(detect message | no message);
  - Detection Probability = P(detect message | message);
  - Dispersion and Distribution;
- Mathematical algorithms:
  - Discrete Cosine Transformation;
  - Discrete Fourier Transformation;
- Information theory means:
  - Embedding algorithms;
  - Cover-object analysis;

# Steganography Algorithms

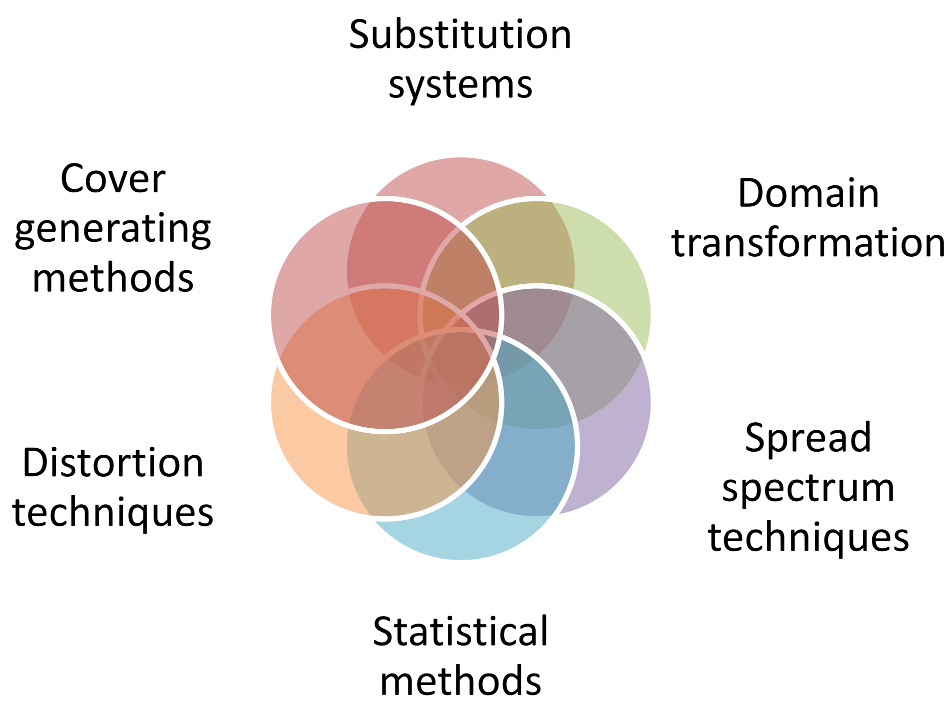
Substitution systems: Least significant Bit Substitution – very weak

Domain transformation:  Discrete Cosine Transform – weak

Spread spectrum techniques – Direct-Sequence or Frequency-Hopping Scheme – good

Distortion techniques – need access to the original cover – major flaw

Cover generating methods – resource consuming



# Steganography – Steganalysis

Steganalysis is the science which studies the techniques used for concealing messages inside cover files.

Steganalysis identifies the presence of an embedded message into a cover file.

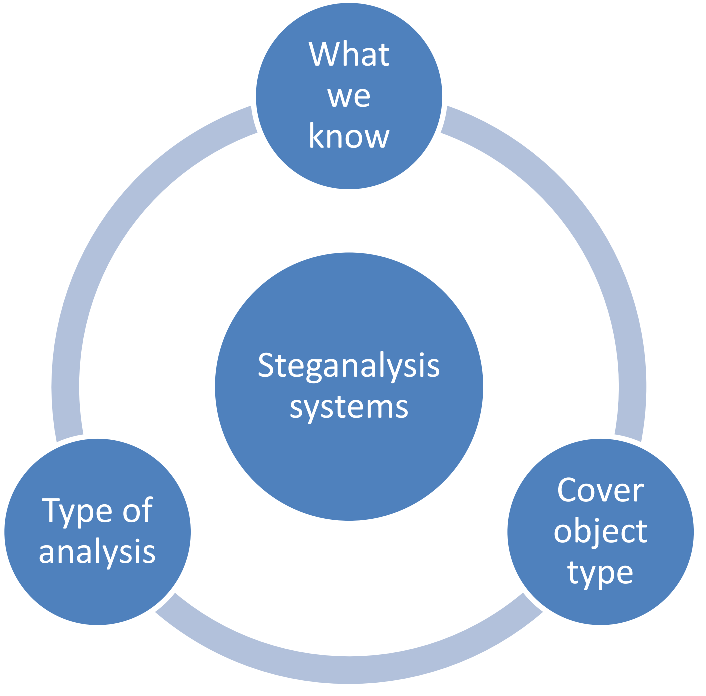
# Steganalysis

The objectives are to accurate identify the presence of a embedded message, keep the consistency of the message when trying to retrieve it and minimize the false positive alerts.

# Steganalysis concept

- Steganalysis reflects upon the steganography techniques applied in cover objects
- Identifies if a possible cover object contains or not an embedded message
- The art or science of discrimination between hidden message and cover object
- Must identify and analysis noise in the cover-objects
- Nonetheless message does not have to be extracted

# Steganalysis classification systems



# The available entropy about stega-content

Only the stega-object is available

The stega-algorithm is known

The original cover object is available

# Steganalysis techniques cover type based

- Analysis of alteration of the least significant bit;
- Identifying mask and filters applied  to spatial domain;
- Analysis of Domain Transformation;
- Statistical analysis of the parity bit;
- Frequency spectrum analysis;
- Intra/Inter frame structural analysis;
- Spatial and temporal prediction analysis;

# Types of analysis

- Statistical steganalysis
  - Least Significant Bit
  - Spread Spectrum Analysis
  - Discrete Cosine Transform
- Target based steganalysis
  - The destruction of the message
  - The analysis of the cover object
  - The identification of message presence

# Steganalysis

- Methods of detecting the use of Steganography
  - Visual Detection (JPEG, BMP, GIF, etc.)
  - Audible Detection (WAV, MPEG, etc.)
  - Statistical Detection (changes in patterns of the pixels or LSB – Least Significant Bit) or Histogram Analysis
  - Structural Detection - View file properties/contents
    - size difference
    - date/time difference
    - contents – modifications
    - checksum

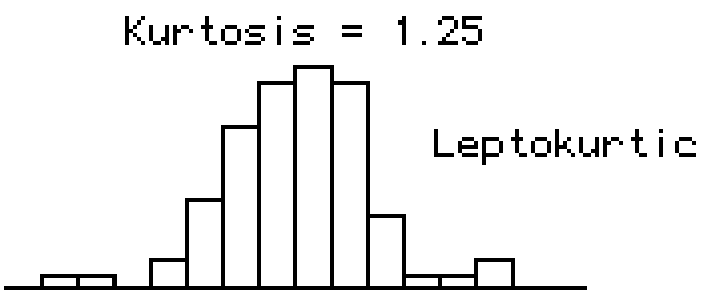
# Methods of detection

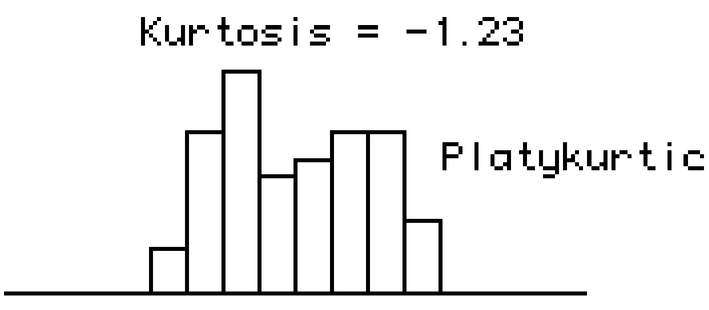
- Categories
  - Anomaly
    - Histogram analysis
    - Change in file properties
    - Statistical Attack
    - Visually
    - Audible
  - Signature
    - A pattern consistent with the program used

# Methods of detection

- Kurtosis
  - The degree of flatness or peakedness of a curve describing a frequency of distribution

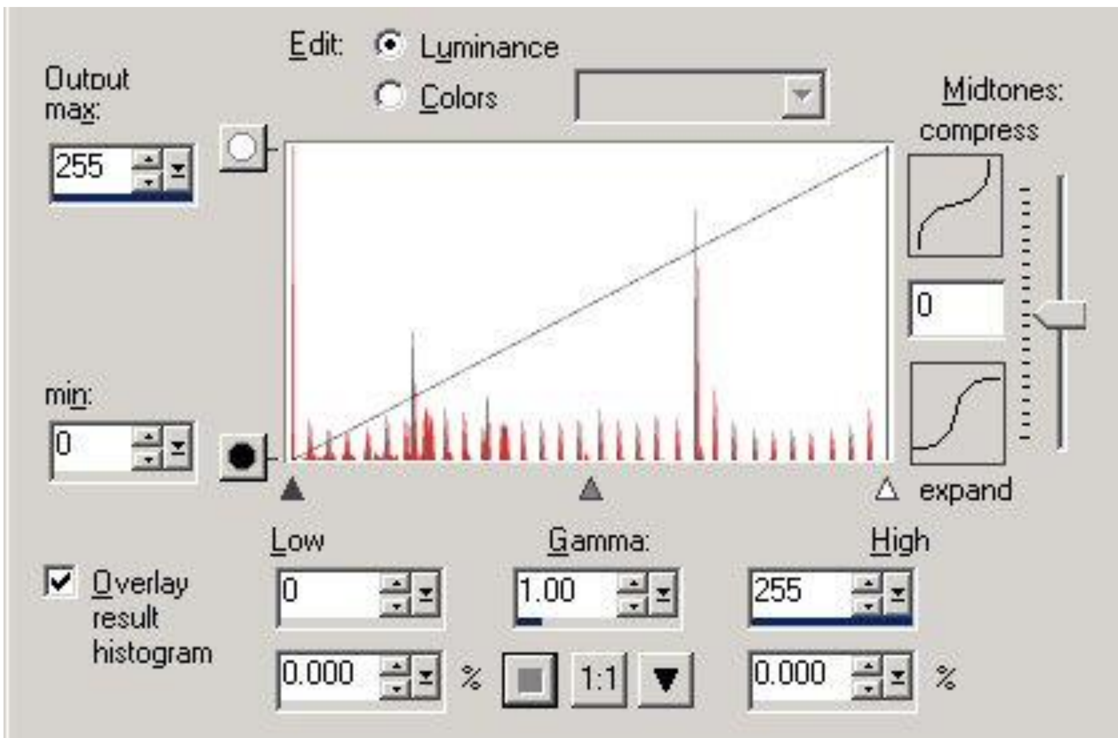
    – Random House Dictionary



# Methods of detection

- By comparing histograms, we can see this histogram has a very noticeable repetitive trend.



# Steganography

# Steganography

1. Can use a key for altering the original message, this way preventing message decryption if the hidden content was revealed;
2. Must use a function for spreading the message through the entire content of the cover file;
3. Combining the key with the dispersion function for decreasing the detection degree;

# Steganography

Steganography image techniques:

- Modifying the least significant bit;
- Applying masks and filters;
- Image transformation algorithms;

# Steganography

Hiding a character at the least significant bit of each color

24 bit bitmap => 3 bytes / color

| Value | Cod | Pixel | Hidden character |
|---|---|---|---|
| 10101111 | R | 1px. | A = ASCII (65 Dec / 41 Hex) |
| 01101110 | G | | |
| 11111110 | B | | |
| 10101110 | R | 2px. | |
| 01101100 | G | | |
| 11111110 | B | | |
| 10101111 | R | 3px. | |
| 01101101 | G | | |
| 11111111 | B | | |

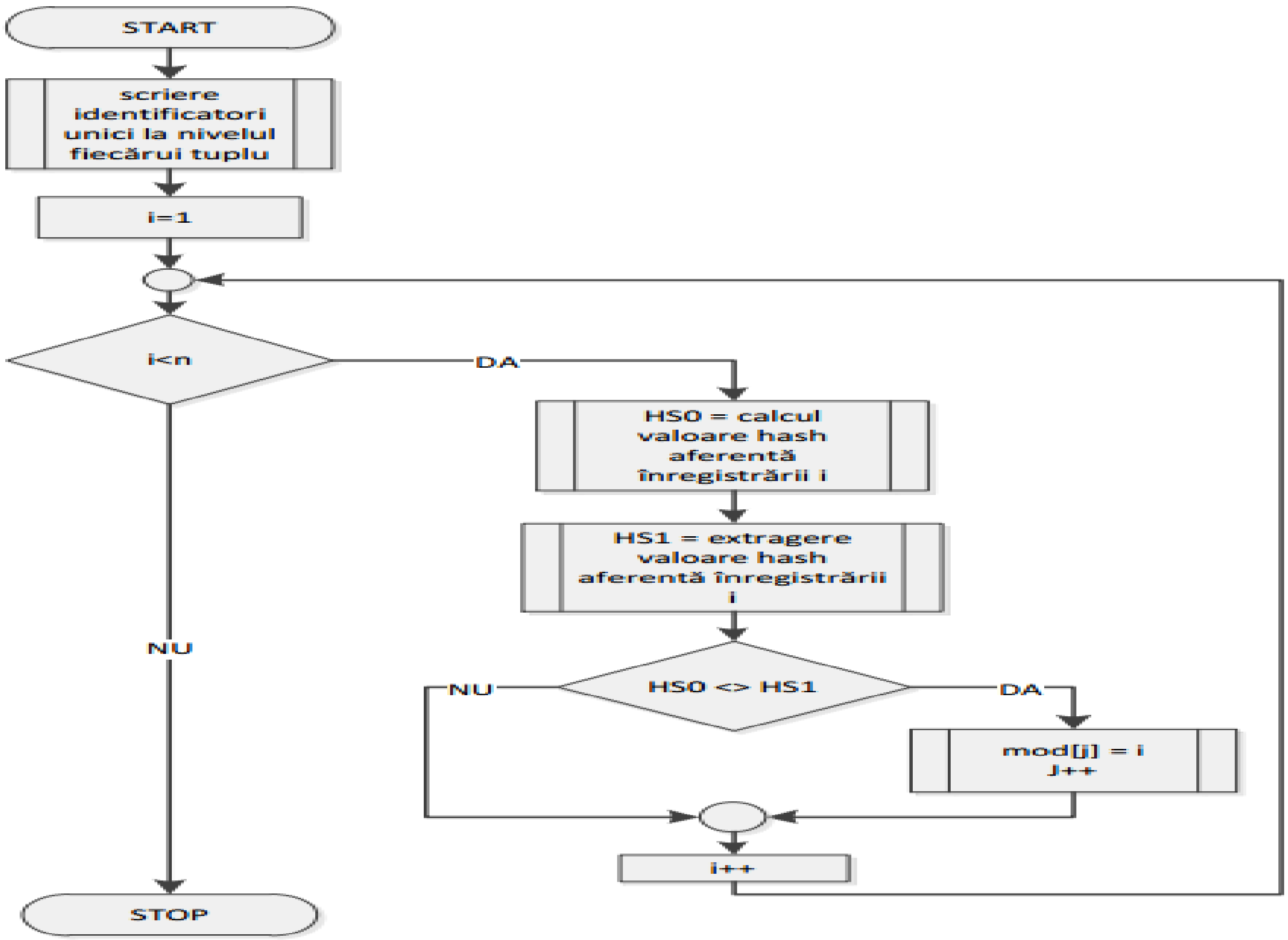
# Integrity Verification

| | IDMEAL | DESCRIPTION | NOKCAL | NOPROTEINS | NOLIPIDS | NOCARBO | NOFIBERS | TYPE | IMAGE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 123 | 1123 | 107,25 | 9,75 | 7,5 | 0,75 | 0 | 1 | (BLOB) |
| 2 | 67 | 167 | 347,5 | 123,2 | 19,265 | 188,75 | 26,45 | 1 | (BLOB) |
| 3 | 64 | 164 | 403,75 | 127,5 | 7,725 | 505,5 | 91,5 | 1 | (BLOB) |

STEG_BLOB = STEG (**BLOB, HASH(67 || 167 || 347,5 || … || 1)**)

| BEFORE | AFTER |
|---|---|
|   |   |

# Integrity Verification



# Steganography - tools

- Steganos
- S-Tools
- StegHide
- Invisible Secrets
- JPHide
- Camouflage
- Hiderman

**Is what you expected?**

This was **Section 2.2 – Imagine**

Color pallets (RGB, CMYK, HSB, Lab, Grayscale, Web-safe)

Characteristics (Hue, Brightness, Saturation, Contrast, Color balance, Bit depth, Resolution)

Types of images (bitmap, vector)

File formats (TIFF, ICO, JPG, GIF, BMP, PNG, EPS, SVG)

Compression

Steganography

Watermarking

# Bibliography for Section 2.2 – Images

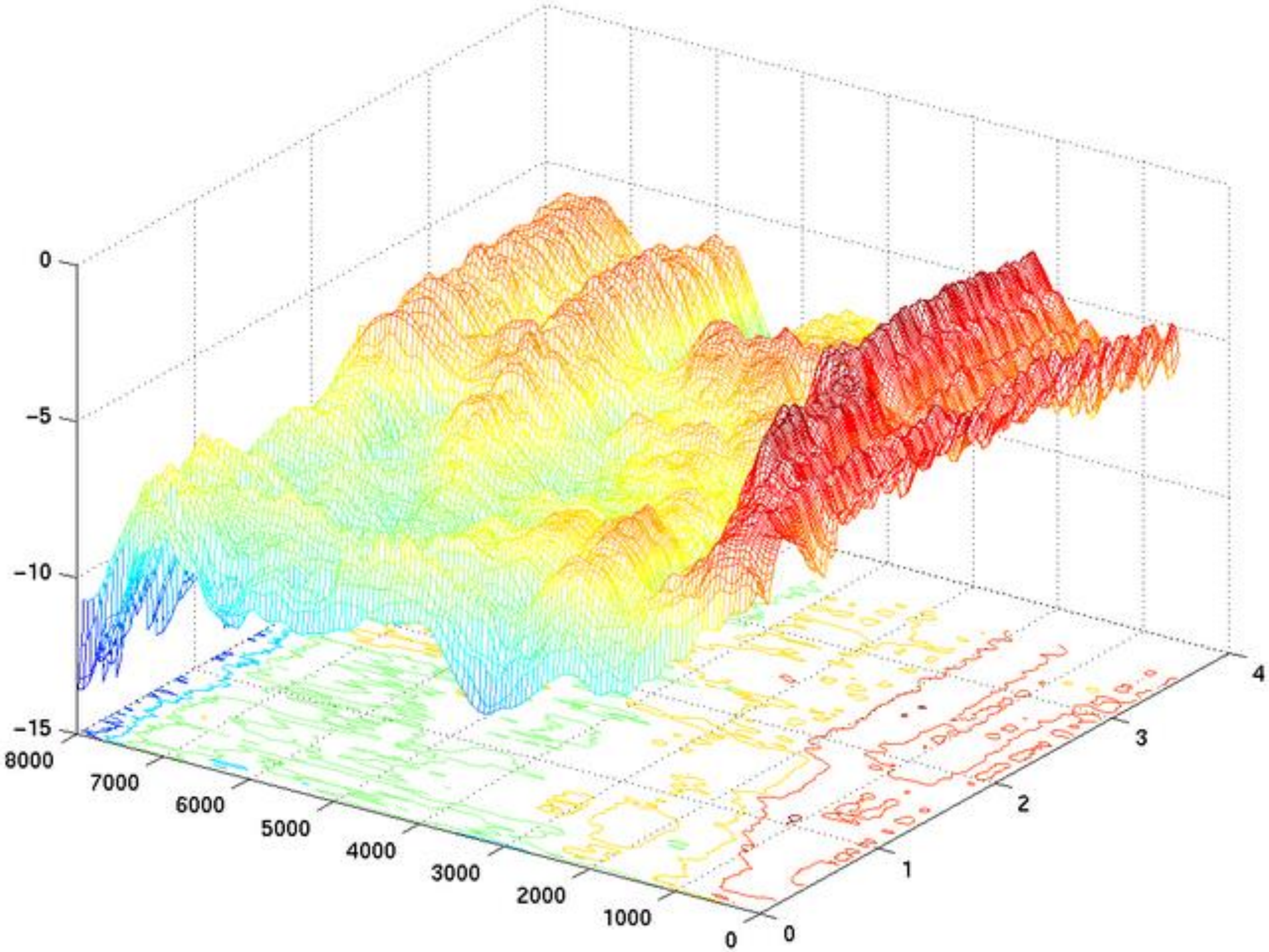
Auguste Kerckhoffs, “La cryptographie militaire”, Journal des sciences militaires, vol. IX, pp. 5–38, Jan. 1883, pp. 161–191, Feb. 1883.

Karen Scarfone, Wayne Jansen and Miles Tracy, “Guide to General Server Security”, National Institute of Standards and Technology, 2008

P. Petitcolas, R. J. Anderson, M. G. Kuhn, “Information Hiding – A Survey”, Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062-1078, July 1999

Zaidoon Al-Ani, A.A. Zaidan, B.B. Zaidan and Hamdan Alanazi, “Overview: Main Fundamentals for Steganography”, Journal of Computing, vol. 2, Issue 3, March 2010, ISSN 2151 - 9617

# 2.3 Sound



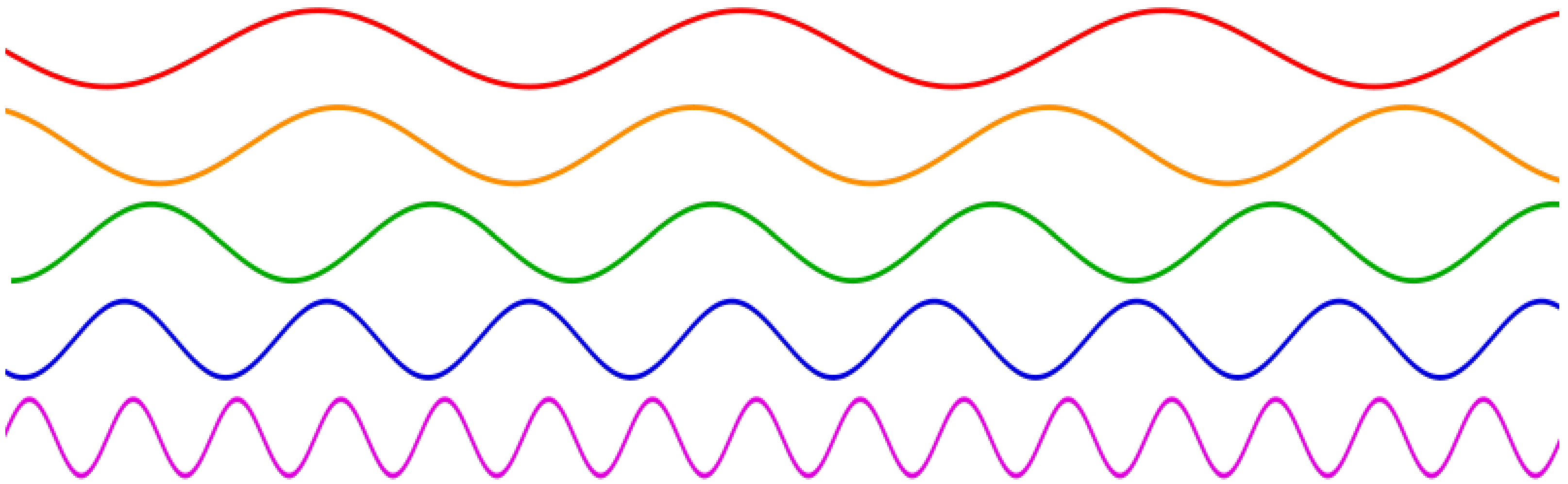
# Audio - Sound

Characteristics:

- Frequency
- Wavelength
- Amplitude
- Intensity
- Speed
- Direction
- Pitch

# Audio – Analogous Sound

Frequency



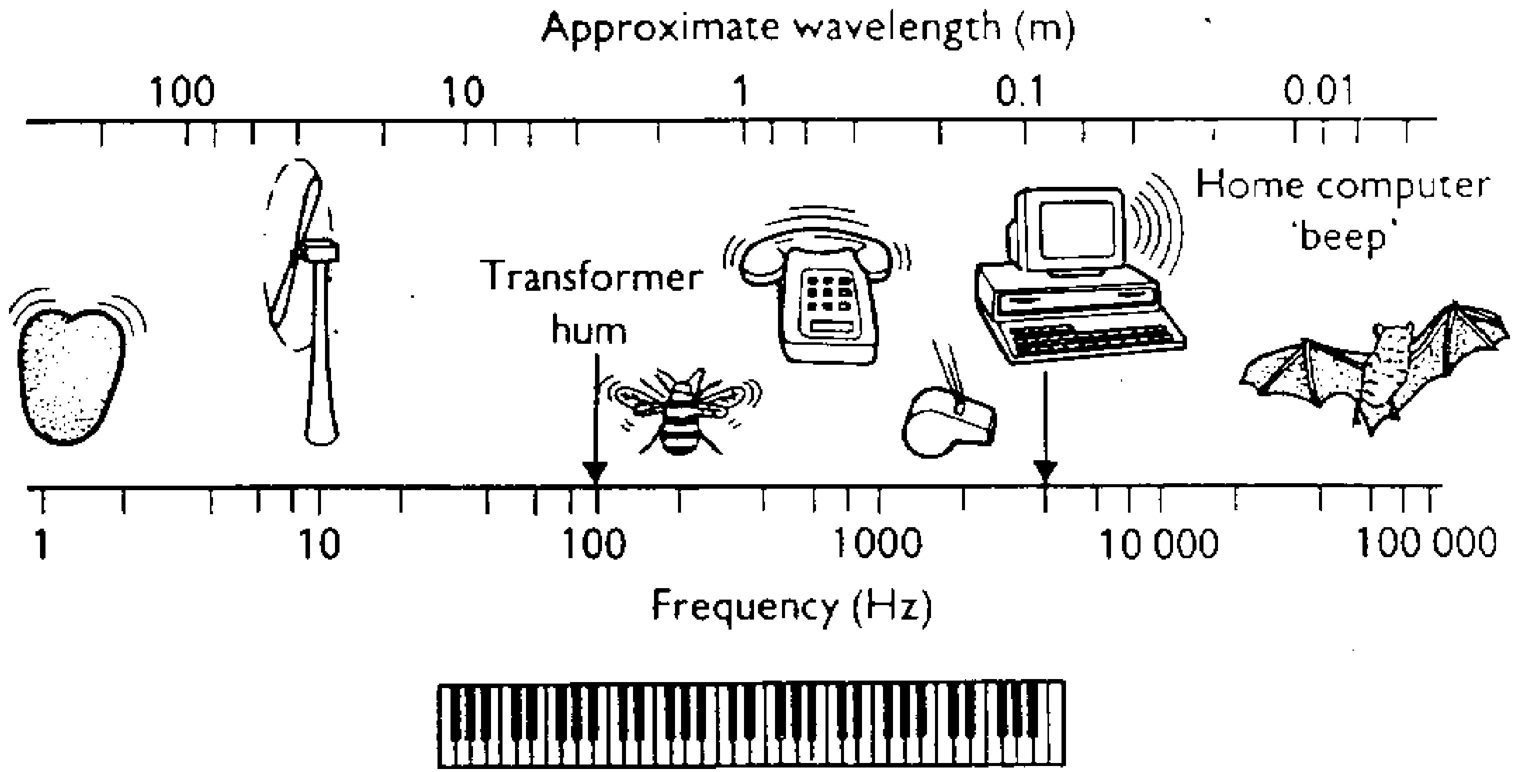
[WIKI]

# Audio – Analogous Sound

Frequency:

- is measured in Hz (Hertz)
- 1 Hz -> one oscillation per second
- We can perceive frequencies between: 20 - 20,000 Hz

# Audio – Analogous Sound




[http://personal.cityu.edu.hk/~bsapplec/characte.htm]

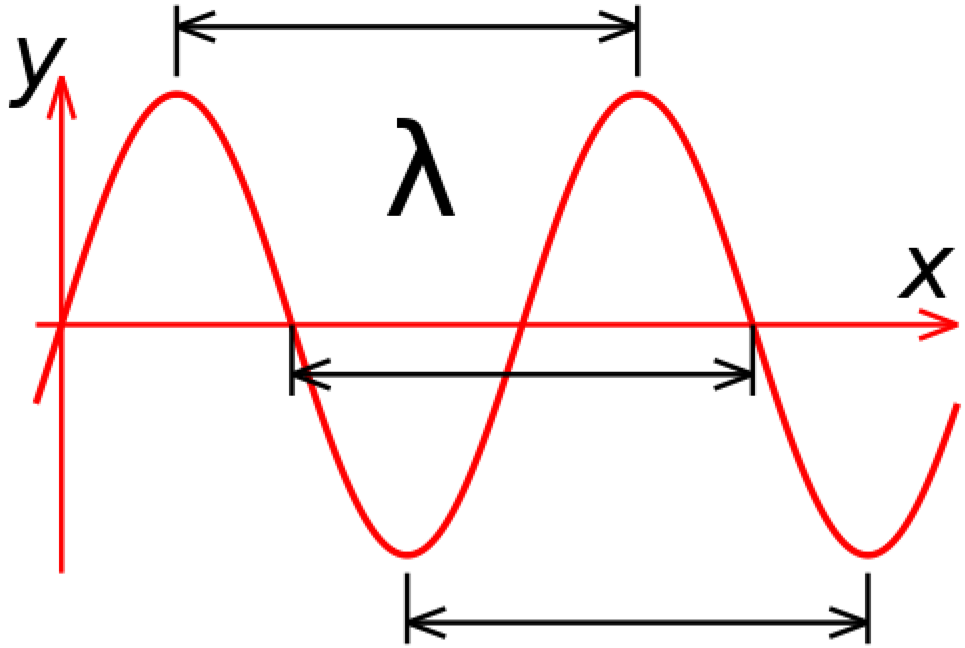
# Pitch

pitch = frequency of sound

- middle C in equal temperament = 261.6 Hz
- the **pitch** of a **sound** is determined by the rate of vibration, or frequency, of the **sound** wave

# Audio – Analogous Sound

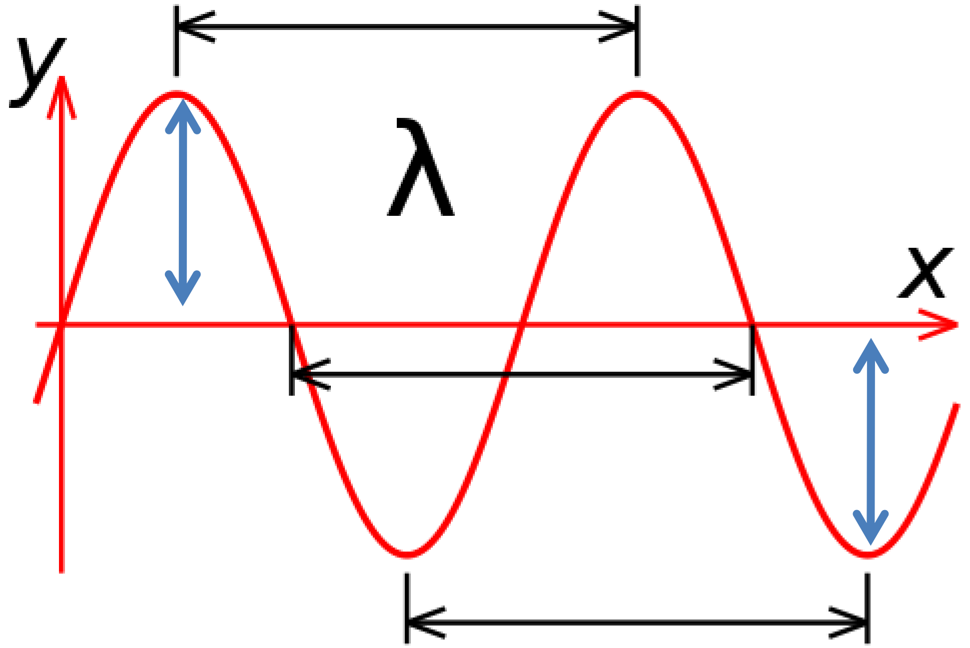
Wave length




[WIKI]

# Audio – Analogous Sound

Amplitude




[WIKI]

# Intensity

The sound intensity may be expressed in decibels above the standard threshold of hearing $I_0$.

$$I(dB) = 10\log_{10}\left[\frac{I}{I_0}\right]$$

$$I(dB) = 10\log_{10}\left[\frac{10{,}000 I_0}{I_0}\right] = 10 \cdot 4 dB = 40 dB$$

# Speed

The speed of sound is the distance travelled per unit time by a wave sound as it propagates through an elastic medium.

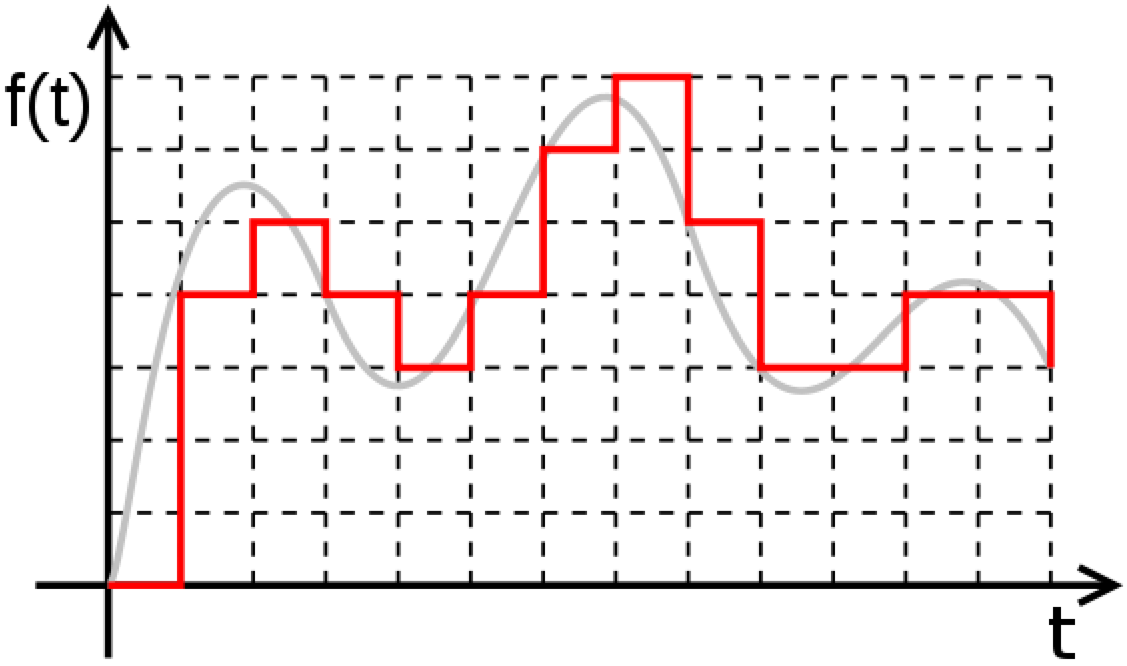
The speed of sound in dry air is approx.:

$$(331.4 + 0.6 * Tc) \text{ m/s}$$

# Audio – Analogous Sound

| Medium | Speed (m/s) |
|---|---|
| Air | 343 |
| Water | 1,372 |
| Concrete | 3,048 |
| Glass | 3,658 |
| Iron | 5,182 |
| Lead | 1,219 |
| Steel | 5,182 |
| Hard Wood | 4,267 |
| Soft Wood | 3,353 |

# Audio – Digital Sound




[WIKI]

# Audio - Codec

Used for encoding and decoding sound

It also uses sound compression algorithms

with loss of information, lossy compression

without loss of information, lossless compression

An audio file often uses a specific type of codec

# Audio - Codec

MP3

- MPEG-1 or MPEG-2 Audio Layer 3
- Defined in 1993 and widely used over the Internet
- Uses a compression algorithm with loss of information (lossy compression): at 128 kbit/s the mp3 audio file is almost 11 times smaller than its audio cd equivalent (.cda)
- The compression is based on eliminating some frequencies that are not perceived by the human ear

# Audio - Codec

MP3

- The compression is made by setting a *bit rate* which gives the number of Kb used for each audio second
- The audio quality is directly dependent on the size of the audio file
- Possible bit rate values: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 and 320 kbit/s
- For a CD audio track, the bit rate is of 1,411.2 kbit/s

# Audio - Codec

MP3 bit rate:

- *variable bitrate* (VBR)
- *constant bitrate* (CBR)
- 1997 the first audio player – Winamp
- 1998 the first portable MP3 player – MPMan

# Audio - Codec

FLAC

- Free Lossless Audio Codec
- Launched in 2001;  a more stable version was hardly released in 2007
- Implements a lossless compression algorithm – the quality is identical with the raw file
- The compression rate is almost 2:1

# Audio Formats

- Ways of storing sound in digital format
- Audio formats classified by compression:
  - without compression: WAV, AIFF, .cda (Audio CD Track) (~ 10 MB per minute);
  - with a maximum of 2:1 lossless compression: FLAC, Apple Lossless, MPEG-4 SLS, MPEG-4 ALS, MPEG-4 DST, Windows Media Audio Lossless (WMA Lossless);
  - with lossy compression: MP3, Windows Media Audio (WMA);

# Audio

| Algorithm | Object | Category |
|---|---|---|
| ADPCM | Sound | Lossy compression |
| MPEG - Audio | Sound | Lossy compression |

# Uncompressed audio format

WAV based on RIFF (Resource Interchange File Format)

AIFF based on IFF (Interchange File Format)

44.100kHz, 16 bit, 2 channels =>

(44100 * 16 * 2) = 176,4 KBytes/sec. (bitRate)

# WAVE - PCM

| Field | Length | Content |
|---|---|---|
| ckID | 4 | Chunk ID: "RIFF" |
| cksize | 4 | Chunk size: 4+*n* |
| WAVEID | 4 | WAVE ID: "WAVE" |
| WAVE chunks | *n* | Wave chunks containing data about formats and actual content |

# WAVE - PCM

For a file with the following characteristics:

- $N_c$ channels
- $N_s$ the total number of blocks
- each block containing $N_c$ samples
- sampling rate is $F$ (blocks per second)
- each sample has a total of $M$ bytes

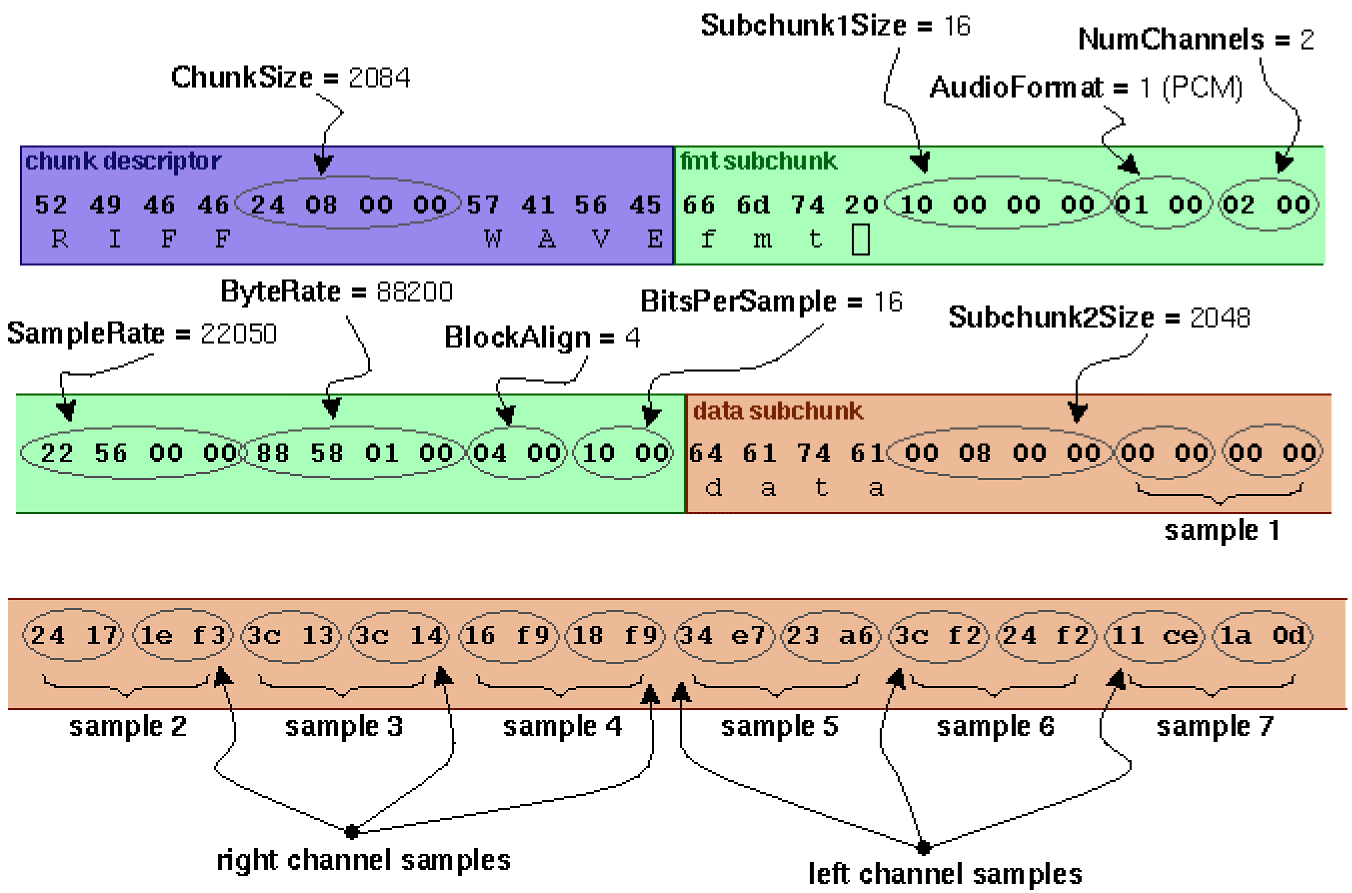
| Field | Length | Content |
| --- | --- | --- |
| ckID | 4 | Chunk ID: "RIFF" |
| cksize | 4 | $4 + 24 + (8 + M * N_c * N_s + (0 \text{ or } 1))$ |
| WAVEID | 4 | WAVE ID: "WAVE" |
| ckID | 4 | Chunk ID: "fmt " |
| cksize | 4 | Chunk size: 16 |
| wFormatTag | 2 | WAVE_FORMAT_PCM |
| nChannels | 2 | $N_c$ |
| nSamplesPerSec | 4 | $F$ |
| nAvgBytesPerSec | 4 | $F * M * N_c$ |
| nBlockAlign | 2 | $M * N_c$ |
| wBitsPerSample | 2 | rounds up to $8 * M$ |
| ckID | 4 | Chunk ID: "data" |
| cksize | 4 | Chunk size: $M * N_c * N_s$ |
| sampled data | $M * N_c * N_s$ | $N_c * N_s$ channel $M$-byte samples |
| pad | 0 or 1 | Padding byte if $M * N_c * N_s$ is odd |

# RIFF - Wave

| Byte Offset | Name | Field Size(bytes) | Notes | |
|---|---|---|---|---|
| 0 | ID | 4 | Letters "RIFF" | RIFF Section |
| 4 | Size | 4 | Size of this file less 8 | |
| 8 | Format | 4 | Letters "WAVE" | |
| 12 | ID | 4 | letters "fmt" | Format section |
| 16 | Size | 4 | Size of format section | |
| 20 | Format | 2 | 1=uncompressed PCM | |
| 22 | Channels | 2 | 1=mono,2=stereo, more for other formats | |
| 24 | Sample Rate | 4 | 44100, 16000, 8000 etc. | |
| 28 | Byte Rate | 4 | SampleRate * Channels * (BitsPerSample/8) | |
| 32 | Block Align | 2 | Channels * (BitsPerSample/8) | |
| 34 | BitsPerSample | 2 | 8,16,24 or 32 | |
| 36 | ID | 4 | The letters "data" | Data Section |
| 40 | Data Size | 4 | Size of the data below | |
| 44 | Data | | The sound data | |

**Rest of sound data until end of data**
Format is first channel sample followed
by second channel sample then back to first
channel, etc. etc.
Each sample will be BitsPerSample in size

# RIFF - Wave



ChunkSize = 2084
Subchunk1Size = 16
NumChannels = 2
AudioFormat = 1 (PCM)
chunk descriptor
52 49 46 46 24 08 00 00 57 41 56 45
R I F F W A V E
fmt subchunk
66 6d 74 20 10 00 00 00 01 00 02 00
f m t
ByteRate = 88200
BitsPerSample = 16
Subchunk2Size = 2048
SampleRate = 22050
BlockAlign = 4
22 56 00 00 88 58 01 00 04 00 10 00
data subchunk
64 61 74 61 00 08 00 00 00 00 00 00
d a t a
sample 1
24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d
sample 2
sample 3
sample 4
sample 5
sample 6
sample 7
right channel samples
left channel samples

# Steganography

Audio steganography techniques:

- Least significant bit encoding;
- Dephasing techniques;
- Hiding a message by adding an undetected echo for the human ear;

**Is what you expected?**

This was **Section 2.3 – Sound**

1. Features (frequency, wavelength, amplitude, intensity, speed, direction, pitch)
2. Codecs (MP3, FLAC)
3. Compression
4. File formats
5. Internal structure WAV – RIFF
6. Steganography

# 2.4 Video

# Video

- Is the most complex element of multimedia field
- Is based on fast displaying of static images (frames)
- Human eye doesn’t detects additional frames above a number of 12-15 frame per second
- For a video sequence of **720*486 pixels** running with **30 frames** per second, the computer must process around **21 MB per second**

# Video

- Digital representation comes with high fidelity and numerous ways of processing digital content.
- The most important aspect is how video frames are displayed

# Video

- Computer screens and some video devices uses a video signal based on three main colours of the RGB format
- The TV band and the majority of video systems use a compound signal in which brightness, hue and frame synchronization are all combined into a one signal

# Video

- ways of displaying video signal*:*
  - interlaced video signal is used in analogic systems; two different sets of lines alternate, odds and evens, running video sequences with 25 frames/second;
  - on digital systems the progressive video signal is used, lines being displayed one after another

# Video

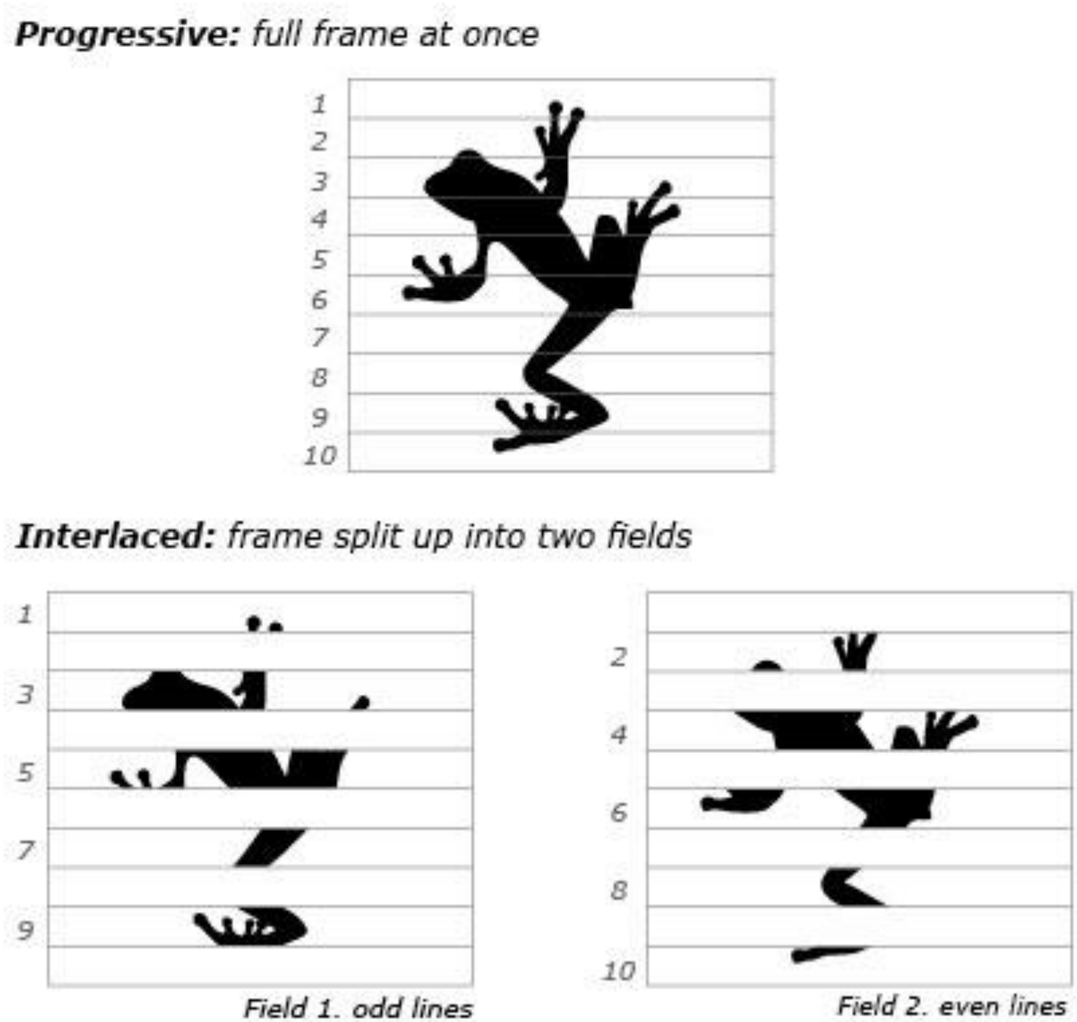
- Interlaced – the image is fully displayed but the it gets gradually clearer until its final shape
- Progressive – means that the image is rendered line by line in a progressive manner until full clarity

# Video

Interlaced


http://justsayyes.wordpress.com/2007/06/28/demonstration-of-an-interlaced-video-and-deinterlaced-video/

# Video




http://cpn.canon-europe.com/content/education/technical/progressive_video.do

# Video

- How colours are rendered:
  - Analog : brightness and hue
  - Digital  : RGB signal
- Displaying the image:
  - Analog : interlaced, small refresh rate
  - Digital  : progressive, high fidelity
- Resolution:
  - Analog : PAL -625 lines, NTSC – 525 lines
  - Digital  : PC – starting with 480.

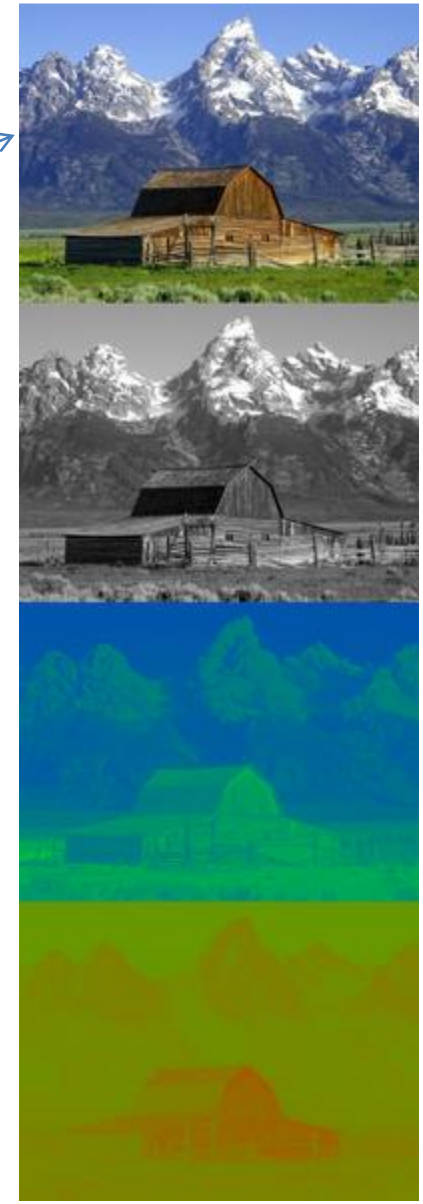
# Video

Three types of video signals:

- Component-VIDEO:
  - For analogue YUV : (Y-brightness, UV-hue)
  - For digital : RGB
- Composite-VIDEO: all the components are mixed together;
- S(separated)-VIDEO: different signal for each component.

# Analogous video

YUV



YUV was used for sending color signal in an infrastructure that uses only black and white images.

[Wiki]

# Video - analogue

Composite – Video
- RCA cables
- DVD Player – TV, Camera Video - TV

S – Video
- Uses S-Video cables
- PC – TV, Camera Video - TV

# Video - analogue

- TV analogous standards for encoding video content and producing a electronic signal:
  - **PAL**
  - **SECAM**
  - **NTSC**

# Video

- NTSC (National Television Standards Committee)
- Released in 1953
- A single video frame made up of 525 horizontal lines and running with a rate of almost 30 frames per second;
- 45 lines are used for image synchronization and the rest for the actual video content
- Displaying a frame is made by combining two fields of 240 lines each one odd and one even

# Video

SECAM (Sequential Colour and Memory)

- Released in the 60s, a system with 525 lines
- The frame rate is around 25 frames per second

PAL (Phase Alternate Line)

- Released in 1966 in the European market
- Uses 625 lines per image with a 25 frames per second
- Only 576 lines are considered active and used for image representation, the other 49 are reserved for synchronization

# Video

**HDTV (High Definition Television)**

- Programmable system based on digital television
- High resolutions that can be used also on computers

# Video

HDV (High Definition Video)

- High resolution: 1,280×720 pixels (720p) or 1,920×1,080 pixels or 4k or 8k

| Video mode | Dimension | Pixels | Display method | Frame rate (Hz) |
|---|---|---|---|---|
| 720p | 1,280×720 | 921,600 | Progressive | 23.976, 24, 25, 29.97, 30, 50, 59.94, 60, 72 |
| 1080i | 1,920×1,080 | 2,073,600 | Interlaced | 25 (50 fields/s), 29.97 (59.94 fields/s), 30 (60 fields/s) |
| 1080p | 1,920×1,080 | 2,073,600 | Progressive | 23.976, 24, 25, 29.97, 30, 50, 59.94, 60 |

# Video compression

- Makes use of redundant frames;
- Frame redundancy :
  - Spatial = intra-frame;
  - Temporal = inter-frame;
- The objective is to preserve the meaning of the each video frame;

# Video compression

- Lossy compression
- Lossless compression

Real time video compression algorithms:

- JPEG, MPEG,  DVI, M-JPEG;
- Based on the types of frame redundancy
- Compresses video content with rates from 50:1 up to 200:1

# Video

The MPEG video standard:

- MPEG1 → video compression, asynchronous sound;
- MPEG2 → digital video used in TV transmissions
- MPEG3 → high resolution for digital television
- MPEG4 → used for video streaming over the Internet

The algorithm is a hybrid one which uses:

- Spectral analysis which by means of a cosine discrete transformation identifies repeatability;
- Hoffman encoding
- Differential coding identifies the differences between frames in order to display them, uses data from previous frames for displaying the next ones;
- Predictive coding analysis the elements that are changing between frames;

# Video

Lossless video codecs:

- AVIzlib
- CamStudio GZIP
- CorePNG
- FastCodec
- MSU Lossless Video Codec
- PICVideo
- TSCC TechSmith Screen Capture Codec
- ZMBV (Zip Motion Block Video) Codec
- JPEG 2000
- YULS

# Video

Lossy video codecs:

- Cinepak
- H.264 - MPEG-4 AVC (Advanced Video Coding) sauMPEG-4 Part 10, approved for Blu-ray
- MJPEG
- JPEG 2000 intra frame video codec MPEG-1 Video (MPEG-1 Part 2)
- MPEG-1 Video (MPEG-1 Part 2)
  - Cinema Craft Encoder
  - FFmpeg

# Video

Lossless video codecs:

- MPEG-2 Video (MPEG-2 Part 2) -H.262
  - Cinema Craft Encoder
  - FFmpeg
  - InterVideo Video Decoder
  - Ligos LSX MPEG-2
  - MainConcept MPEG-2
  - TMPGEnc
- MPEG-4 ASP (Advanced Simple Profile) or MPEG-4 Part 2
  - DivX
  - FFmpeg MPEG-4
  - HDX4
  - Nero Digital
  - Xvid

# Video

- RealVideo
- Snow Wavelet Codec
- Sorenson Video, Sorenson Spark
- Tarkin
- Ffmpeg
- TruDef high definition fractal video codec
- VC-1 (SMPTE standard, subset of Windows Media Video)
- VC-3 SMPTE standard
  - Avid DNxHD
  - FFmpeg
- Windows Media Video (WMV)

# Video – storage media

| | Blu-ray Disc | HD DVD | DVD |
|---|---|---|---|
| Capacity | 25 GB | 15 GB | 4.7 GB |
| Maximum bit rate – data transfer | 53.95 Mbit/s | 36.55 Mbit/s | 11.08 Mbit/s |
| Maximum bitrate – audio + video + subtitles | 48 Mbit/s | 30.24 Mbit/s | 10.08 Mbit/s |
| Maximum bitrate – Video | 40 Mbit/s | 29.4 Mbit/s | 9.8 Mbit/s |
| Video resolution | 1920 x 1080 | 1920 x 1080 | 720×480 (NTSC)<br>720×576 (PAL) |
| Frame rate | 24 p, 50/60i | 24p, 50/60i | 24 p, 50/60 i |

# Video Processing

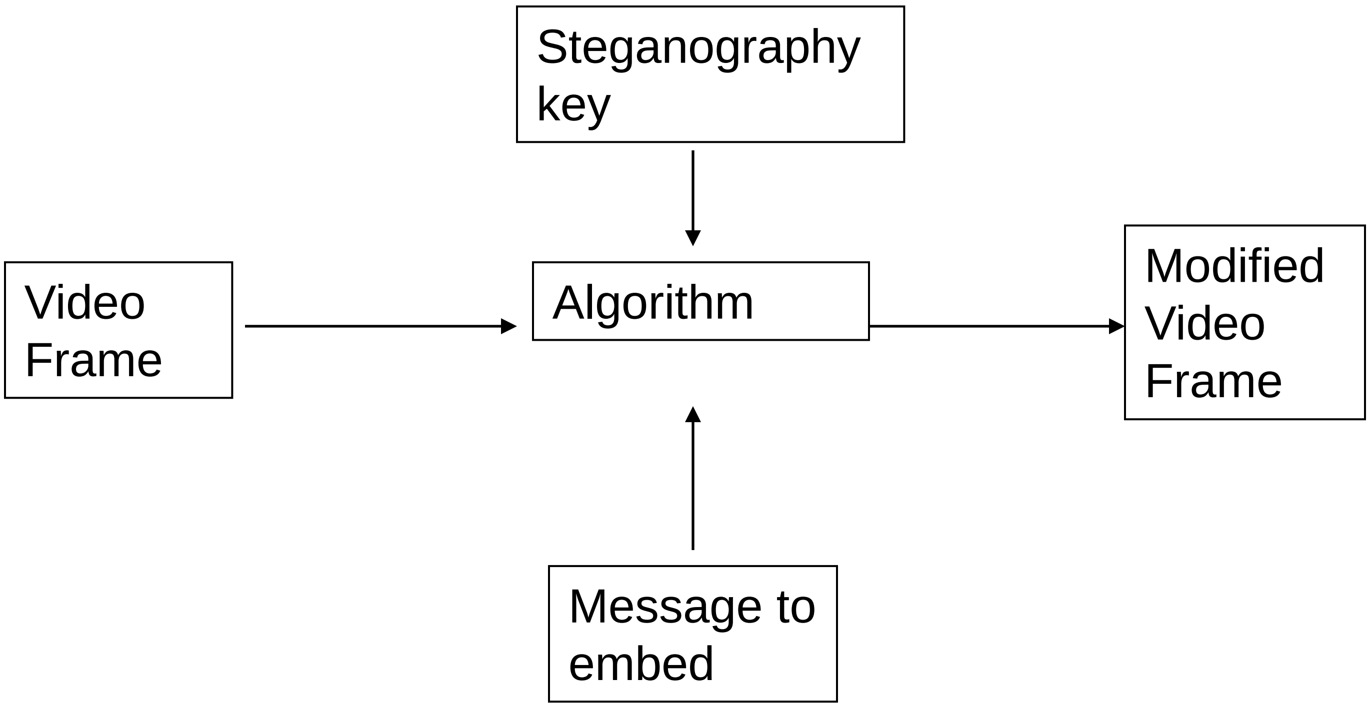
Direct access to video frames:

- TIME CODE (each frame has a unique BCD number (Binary Code Decimal), which gives the hour, minute, second and millisecond;
- FRAME CODE (based on the frame indexing mechanisms, each frame has a ordered positive number;

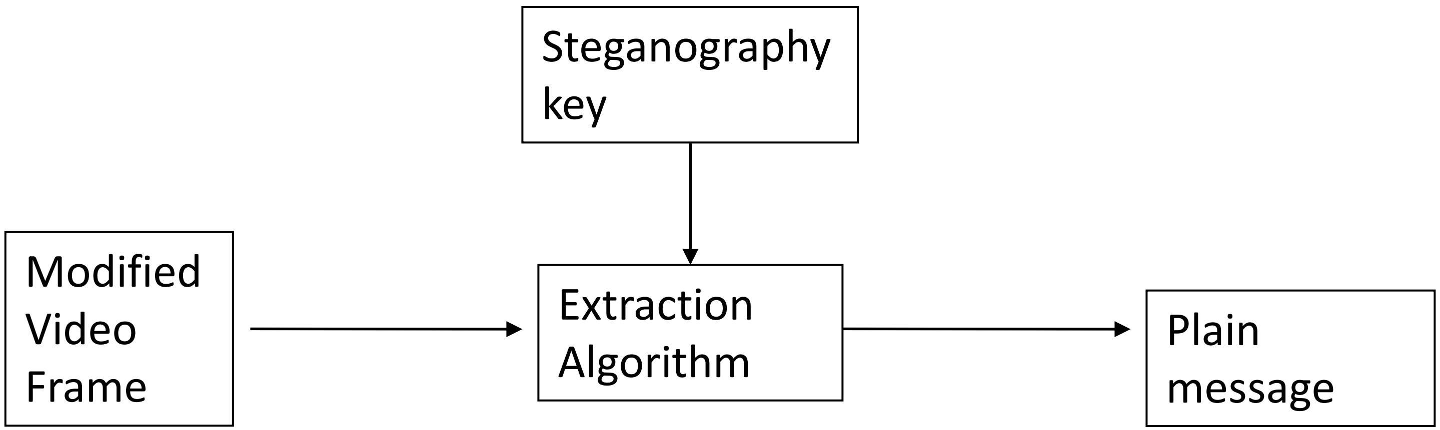
# Steganography techniques

- Conventional oriented on images such as: intra frame, spatial domain, transformed domain;
- Recent techniques based on motion vectors and compression algorithms;

# Steganography – embedding



# Steganography - extracting



# Watermarking

- Protects the authenticity and authorship by using algorithms that prevent the restoration of digital content to its natural state;
- Assures digital content integrity by using fragile watermarking techniques;

**Is what you expected?**

This was **Section 2.4 – Video**

Concept
Characteristics (speed, resolution)
Representation (analogic, digital)
Color representation (RGB, YUV)
Ways of displaying images:

- interlaced
- progressive

Compression type (spatial, temporal)
Video codecs (DivX, Xvid,Ffmpeg, Cinepak)

# 2.6 Digital Rights Management

# Digital Rights Management

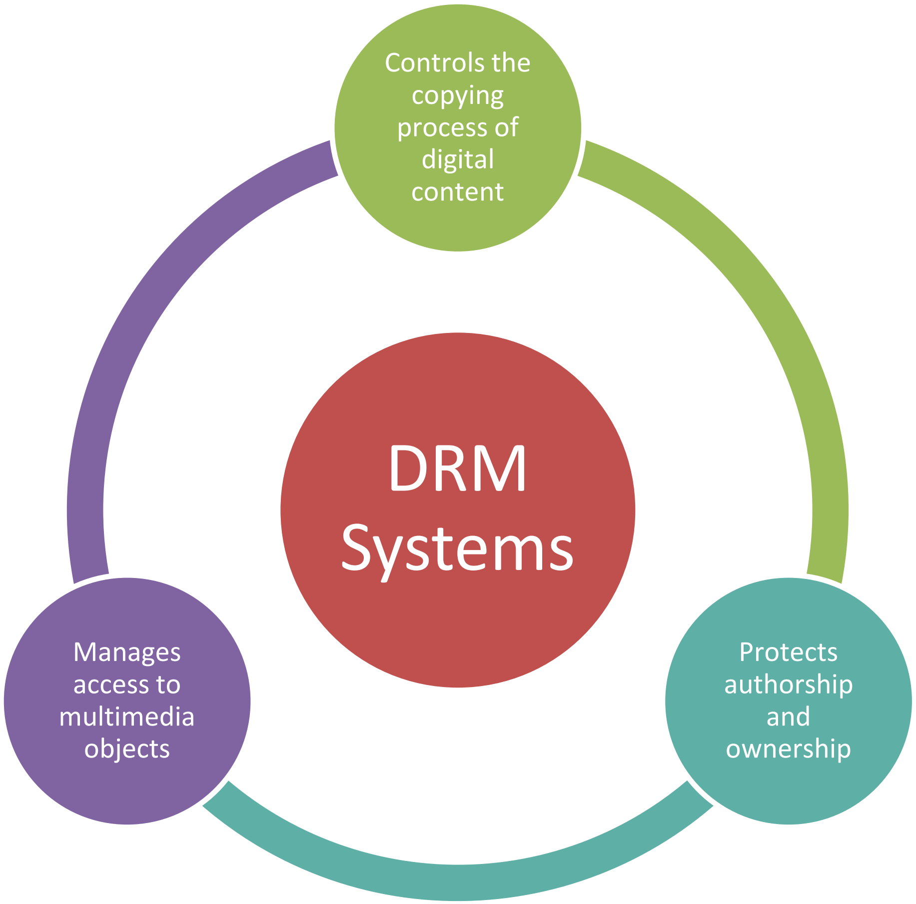
Application and enforcement of regulations provided by law for digital content by use of computer systems.

# Digital Rights Management

Objectives:

- protecting the digital content;
- protecting ownership and authorship;
- allowing temporary access to digital content for a limited period of time;
- restrict access to certain features;

# Digital Rights Management



# Digital content protection

Digital Rights Management Protection

Technological means

Contractual means

Access control

Usage control

Rights expression

# DRM Roles

From a technological perspective, a DRM system must have the following two distinctive roles that must complement each other, as follows:

- access control – control access to digital content and allow only authorized users to make use of it;
- usage control – protect rights of the copyright owner by limiting what an authorized user can do with the digital content by controlling the entire distribution chain of multimedia material;

# DRM Methods

The role of a DRM system is to protect or limit the access to the digital content by unauthorized consumers by:

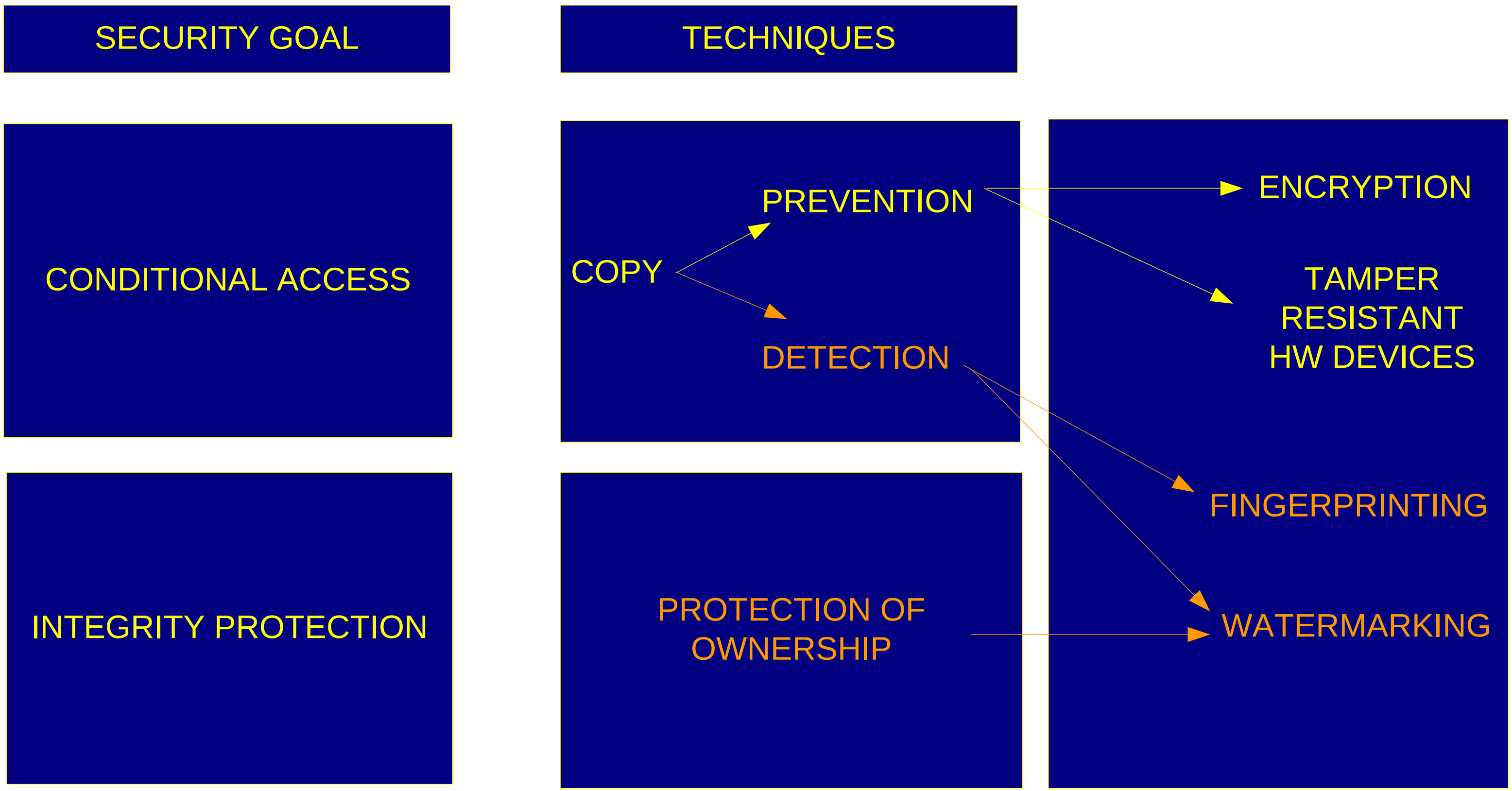
- limit the access by using encryption based digital containers;
- rights lockers architectures used to allow users to access their own digital materials from different types of devices;
- copy generation management systems, or CGMS, limits the number of copies that a user can make from its digital content; after this limit the content becomes unusable.

# DRM Attacks

DRM system face two major types of attacks that can allow a malicious user to access digital content without permission:

- attack for finding the encryption key; every user that has access to the encryption key is allowed to view and manipulate the digital content;
- unencrypted content capturing; for this attack users must exploit flaws in the standardization procedure of a DRM system.

# Digital Rights Management



# 2.5 Digital signature

# Digital signature

Digital signature is a bi-univocal connection between a digital document and its author used in order to reflect a security feature called non-repudiation.

# Digital signature

Objectives:

- ensures the authenticity of signed documents;
- detects changes by maintaining the integrity;
- identifies the signer uniquely - non-repudiation;

# Digital signature

Creating public and private keys:

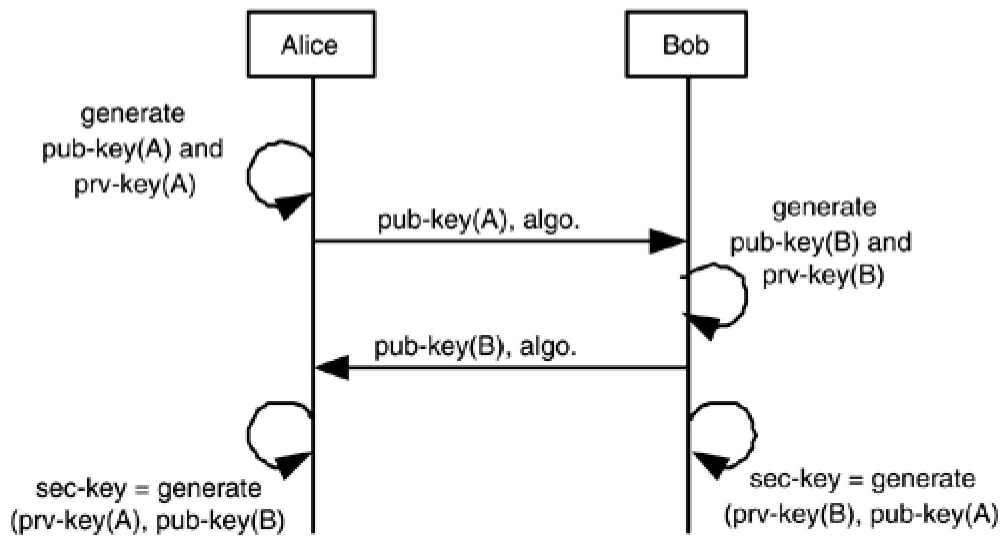
keytool -genkey -alias sender -keystore testkeystore.ks -keyalg RSA/EC

keytool -genkey -alias recev -keystore testkeystore.ks -keyalg RSA/EC

# Digital signature

1. Encrypting the message with a secret symmetric key
2. Concatenating the symmetric key with the hash of the key and of the message
3. Encrypting the resulted string using the public key of the recipient
4. Signing data send with the private key of the sender
5. Validating the signature with the public key of the sender
6. Decrypting the string with the private key of the recipient for extracting the symmetric key
7. Checking integrity of the key using the retrieved hash value
8. Decrypting the message using the symmetric key of which integrity was already verified
9. Computing the hash value of the message extracted
10. Validating the hash value of the extracted message with the one received

# KeyAgreement



# Section Conclusions

2.1 Security in Multimedia
2.2 Image
2.3 Sound
2.4 Video
2.5 Digital signature
2.6 Digital Rights Management

ISM Issues Summary
**for easy sharing**

3

Steganography, Zoom, Compression, Sound, Digital Signature

# **Hands-On** Exercises

# Thanks!

Our business card is


**IT&C Security Master**
Master Program

Calea Dorobantilor, No.15-17
010552 Bucharest
ism@ase.ro
http://www.ism.ase.ro
T +40 21 319 19 00 - 310


We support