# *Image/Video Forensics: Steganografia*

***Prof. Sebastiano Battiato***

***battiato@dmi.unict.it***

IMAGE PROCESSING LABORATORY

# Introduzione (1/2)

La circolazione e la condivisione delle informazioni ha, nella nostra epoca basata sulla comunicazione, un'importanza cruciale. Spesso però c'è il **desiderio** o la **necessità** di mantenere queste comunicazioni **riservate**, in modo che nessuno, tranne il legittimo destinatario, possa acquisire informazioni ritenute sensibili.

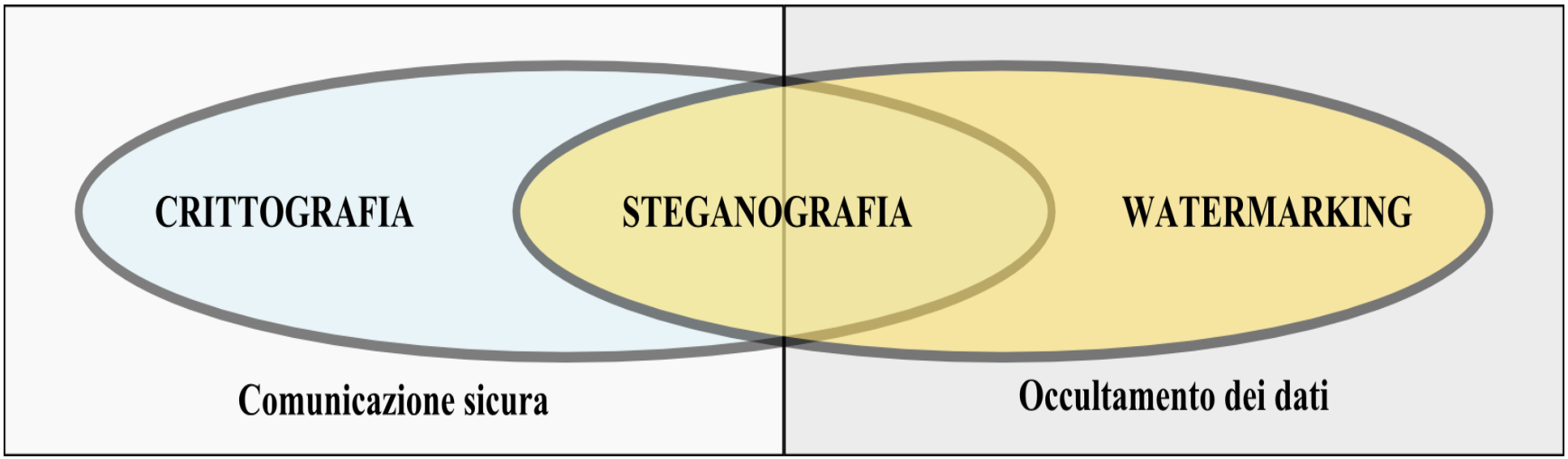
L'approccio più utilizzato per rendere **privata** la conversazione è quello di rendere il messaggio **incomprensibile** a chi non è a conoscenza delle tecniche necessarie a renderlo nuovamente leggibile.

La cosiddetta rivoluzione digitale ha portato anche nel settore della segretezza (o sicurezza) delle informazioni nuovi paradigmi di implementazione di teorie e tecniche già note.

# Introduzione (2/2)

- La scienza che si occupa di questo problema è la **crittografia**, il cui obiettivo principale è quello di garantire la segretezza attraverso la codifica del messaggio. Il messaggio è visibile, ma è **codificato** mediante appositi algoritmi di cifratura che lo rendono incomprensibile a chi non è a conoscenza dei relativi sistemi di decodifica.

- Il **watermarking** (letteralmente filigranatura) invece è una tecnologia grazie alla quale è possibile inserire opportune informazioni in un segnale, in particolare su file multimediali quali immagini e video, magari per segnalarne l'originalità o il titolare dei diritti di proprietà. Un **watermark**, proprio come le filigrane delle banconote, deve essere visibile solo in certe condizioni – per esempio dopo l'applicazione di opportuni algoritmi.




La **Steganografia** nasconde l'esistenza stessa del messaggio, includendolo in un mezzo che potremmo definire "neutrale" e garantendo quindi la segretezza della comunicazione.

# *Steganografia*

Il termine steganografia deriva dai vocaboli greci *stèganos* (nascosto) e *gràfein* (scrivere).

Insieme di tecniche che consentono di nascondere messaggi, che devono essere intelligibili al solo destinatario, inserendoli all'interno di un contesto del tutto estraneo, che funge da **contenitore**, in grado non tanto di nascondere il contenuto ma la stessa esistenza della comunicazione, agli occhi di un eventuale osservatore.

**La steganografia è la scienza (arte) di comunicare senza essere osservati**

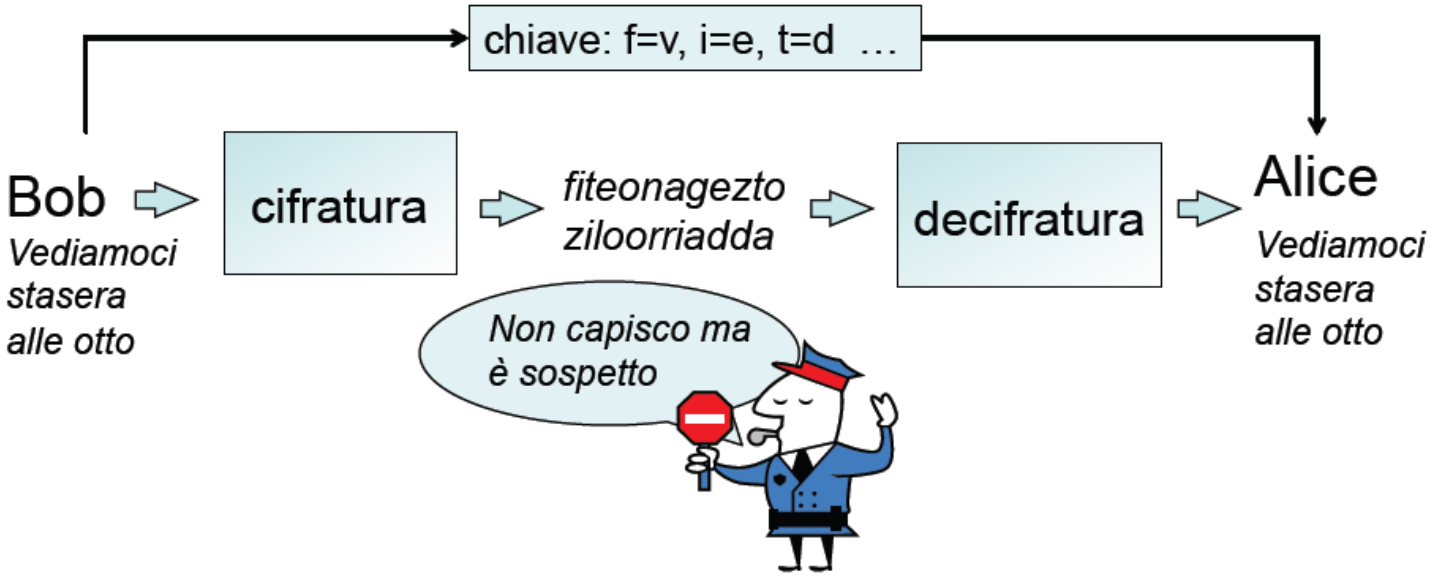
# *Steganografia*

A differenza della crittografia, dove l'avversario sa dell'esistenza della comunicazione, **l'obiettivo della steganografia è nascondere l'esistenza stessa della comunicazione nascondendo il vero messaggio all'interno di un messaggio dal significato innocuo.**

# *Steganalisi*

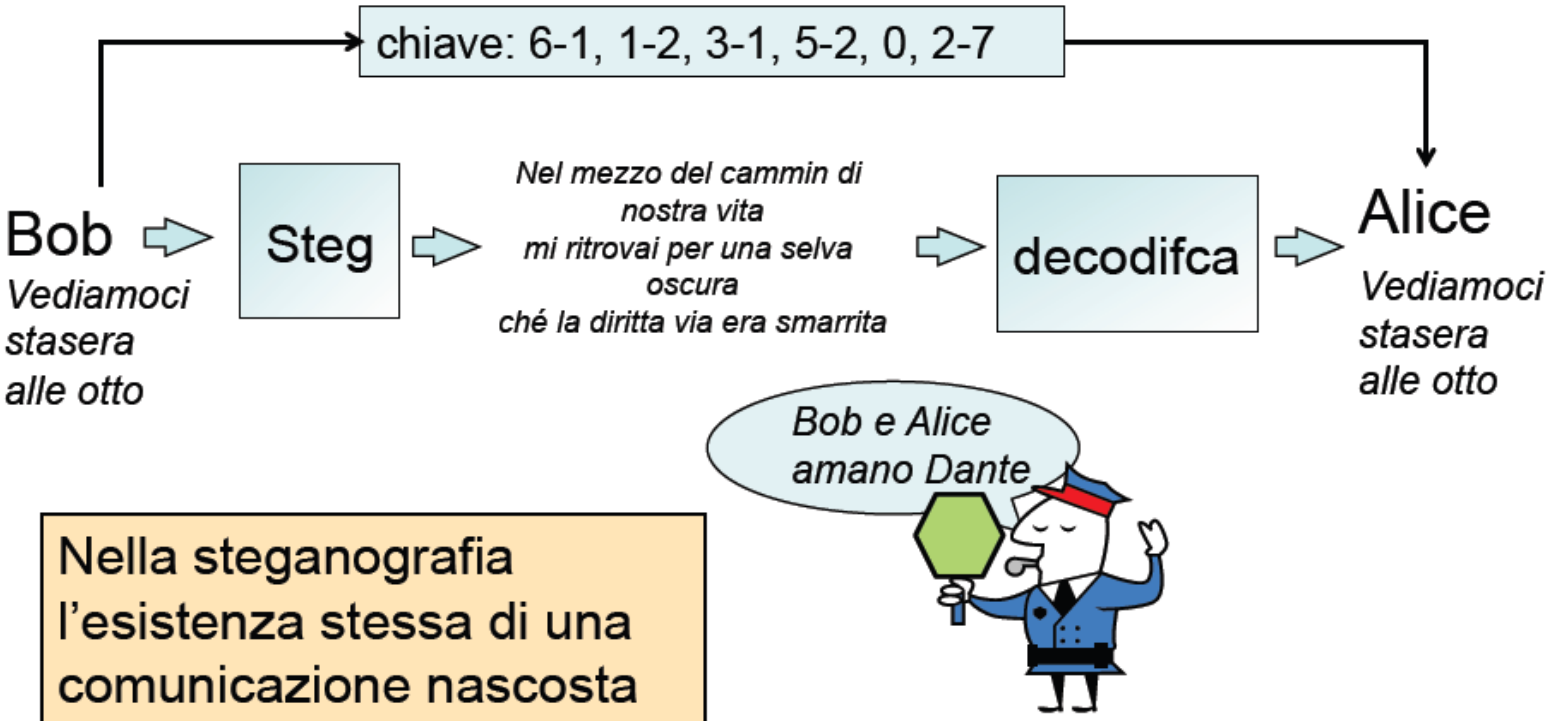
Tecniche di analisi per la rivelazione di messaggi nascosti (anche senza decifrarli) . Possibili Motivazioni: contro-spionaggio, anti-terrorismo, controllo dell'opinione pubblica in regimi totalitari, raccolta di dati (anche sensibili) per motivazioni commerciali o per fini illeciti. Indipendentemente dalle motivazioni lo sviluppo di tecniche di steganalisi è indispensabile allo studio delle stesse tecniche di steganografia.

# *Steganografia vs. Crittografia*




In certi scenari l'esistenza stessa di un messaggio cifrato può destare sospetti e non essere ammessa.

# ***Steganografia* vs. Crittografia**




Nella steganografia l'esistenza stessa di una comunicazione nascosta rimane segreta

# *Steganografia vs. Crittografia*

Importante non confondere la crittografia con la steganografia in quanto:

- Crittografia:
  - ✓ Ha lo scopo di nascondere il **contenuto** di un messaggio.

- Steganografia:
  - ✓ Ha lo scopo di nascondere l'**esistenza** del messaggio.

In molte situazione l'uso della sola crittografia o della sola steganografia non è sufficiente quindi le due tecniche vengono combinate.

# *Un po' di storia…*

La steganografia vanta origini molto antiche:

- **Erodoto(400 a.C.)**: Racconta la storia di un nobile persiano che fece tagliare a zero i capelli di uno schiavo fidato al fine di poter tatuare un messaggio riservato sul suo cranio. Una volta che i capelli furono ricresciuti, inviò lo schiavo alla sua destinazione con la sola istruzione di tagliarseli nuovamente.
- **Griglie di Cardano (XVI secolo)**:

Sir John regards you well and spekes again that
all as rightly 'vails him is yours now and ever.
May he 'tone for past d'lays with many charms.

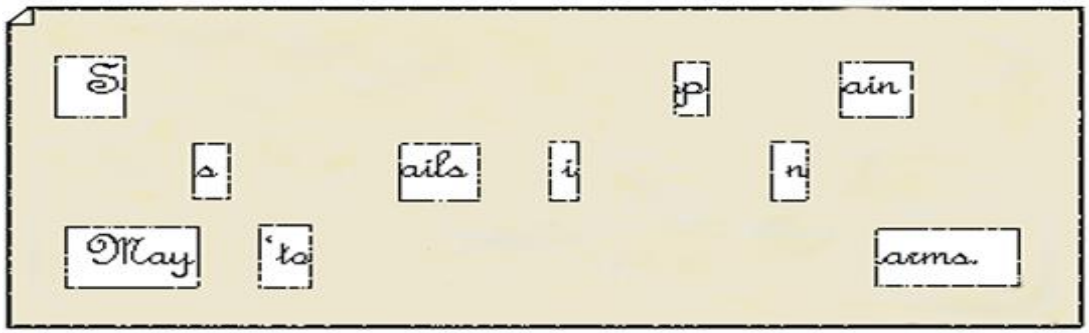



IMAGE PROCESSING LABORATORY

# *Un po' di storia…*

- **Cifre nulle**: Il messaggio trasmesso veniva composto intenzionalmente in modo tale che, unendo le prime lettere di ogni capoverso (o con altre tecniche), si otteneva un messaggio di senso compiuto. Il seguente è un testo realmente inviato da una spia tedesca durante la seconda guerra mondiale:

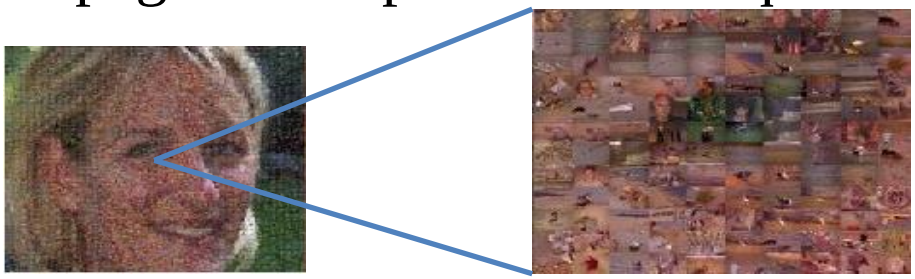
***Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.***

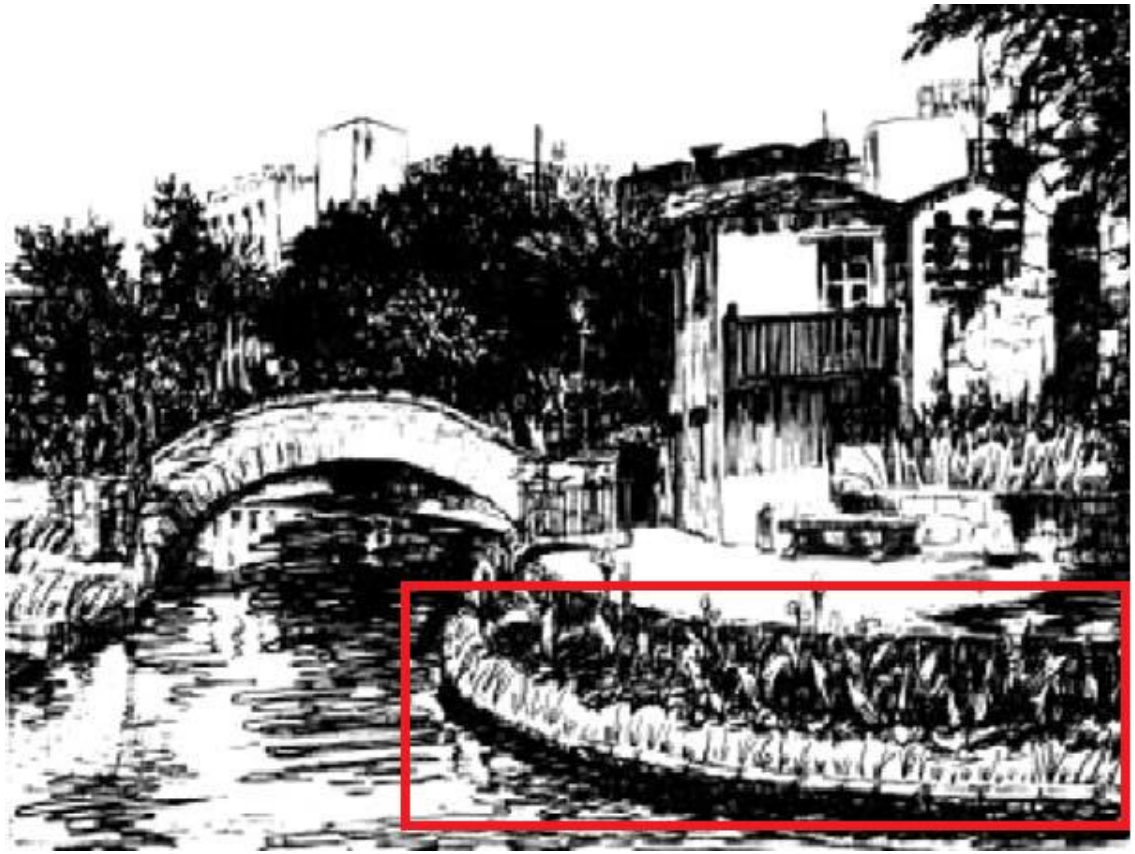
Considerando in sequenza la seconda lettera di ogni parola, si ottiene il messaggio: ***Pershing sails from NY June 1*** (anche se in realtà c'è una "r" di troppo e la "i" alla fine viene interpretata come 1).

# *Un po' di storia…*

- **Inchiostri invisibili**: Sostanze che in condizioni normali non lascino tracci visibili sul foglio di carta, ma che lo diventano sottoponendo il foglio una fonte di calore.

- **Micropunti fotografici**: Fotografie della dimensione di un punto dattiloscritto che, una volta sviluppate e ingrandite, possono diventare pagine stampate di buona qualità.





In questo esempio il messaggio è nascosto nelle diverse lunghezze dei fili d'erba sull'argine del fiume, interpretate secondo il codice MORSE: un filo d'erba lungo viene interpretato come una linea, un filo d'erba corto viene interpretato come un punto.

# Steganografia e Scienza

- Il primo a formalizzare in termini matematici la steganografia, separandola dalla crittografia, fu l'abate tedesco Tritemio, personaggio poliedrico del periodo rinascimentale. Egli scrisse, intorno al 1500, un trattato intitolato appunto "**Steganographia**".
- "*Il forte desiderio di Tritemio era quello di lasciare al mondo in eredità tutto il suo sapere riguardante la comunicazione possibile attraverso metodi non conosciuti. Non voleva però, allo stesso tempo, che queste informazioni cadessero in mani sbagliate o, peggio ancora, fossero vittime della mannaia della censura ecclesiastica.*" Tritemio nascose le sue conoscenze in manoscritti, sperando che non venissero messi all'Indice. Ciò nonostante, la Chiesa ritenne quei manoscritti troppo espliciti, e condannò le copie esistenti ad essere bruciate.

# Steganografia e Scienza

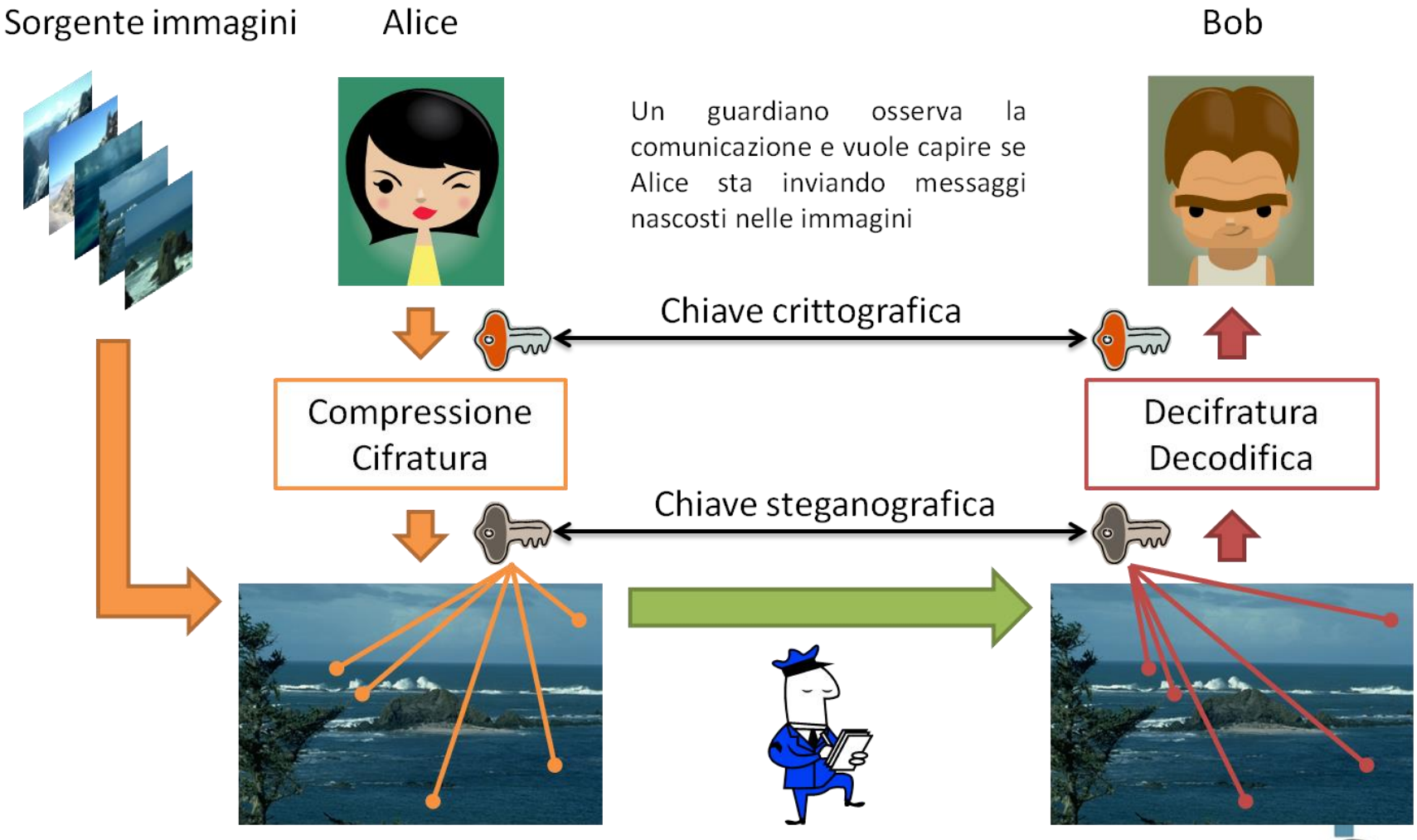
- La prima versione fu pubblicata solo nel 1606, ma il lungo periodo di tempo intercorso tra la scrittura e la pubblicazione fa supporre che ci possano essere stati errori nella trascrizione dell'opera originale, rendendo così il messaggio nascosto non più "estraibile". Altra opera rilevante di Tritemio è "Clavis steganographie" (Le chiavi della steganografia), in cui l'abate, come nell'altra opera, espone tecniche di occultamento di messaggi all'interno di testi in chiaro.

# *Il problema dei prigionieri*

Ai giorni nostri lo studio di questa materia nella letteratura scientifica si deve a Simmons che nel 1983 formulò il **problema dei prigionieri**.

In questo contesto **Alice** e **Bob** sono in prigione e devono escogitare un piano per fuggire. Tutti i loro messaggi vengono scambiati tramite un guardiano. Se quest'ultimo scopre che essi si **scambiano messaggi segreti** metterà uno di loro in isolamento ed il piano fallirà. Quindi essi devono trovare un metodo per nascondere il loro testo in un testo apparentemente innocuo.

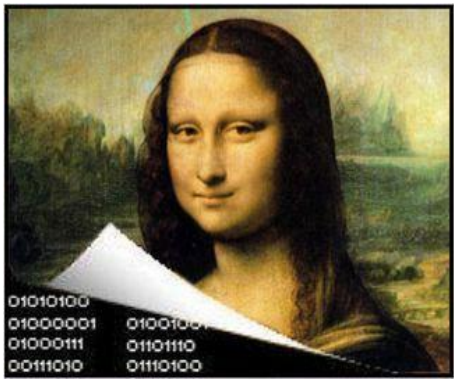
# *Il problema dei prigionieri*



# *Steganografia digitale*

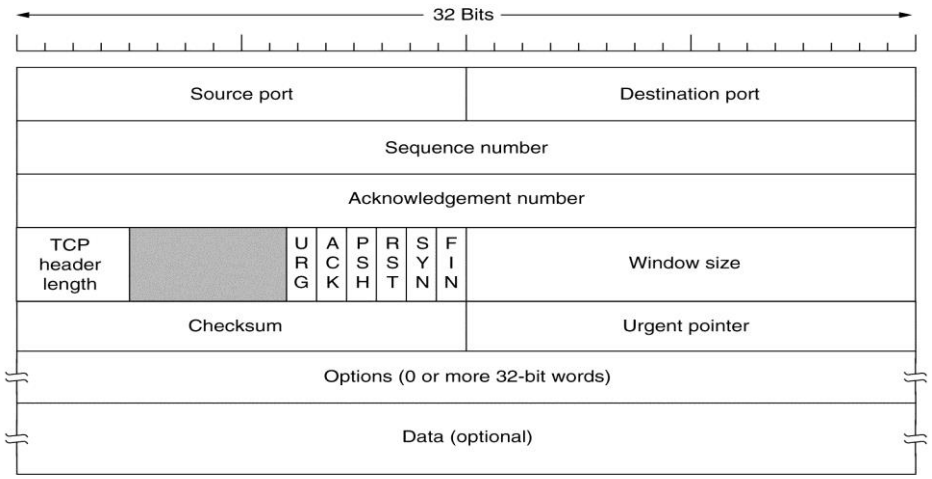
L’avvento dell’era digitale e della rete Internet ha permesso lo sviluppo di tecniche per la steganografia consentendo l’utilizzo dei seguenti supporti digitali:

- File di Immagini
- File Audio/Video




Ma anche:

- File System (Steganografici)
- Header pacchetti TCP/IP



# Formalismi

- Ci riferiremo all'immagine designata a contenere il messaggio con il nome di immagine **cover** o **contenitore**. Quando il messaggio, detto **payload** (letteralmente carico), viene inserito nella cover image, chiameremo il risultato **stego-image** o **immagine stego**. Quindi, in termini matematici, si avrà:

$$\text{stego-image} = F\big(G(\text{cover}), H(\text{payload})\big)$$

dove

- $F$ è la funzione steganografica, che prende in input un'immagine **cover** ed un messaggio e restituisce l'immagine **stego**;
- $G$ è una funzione che elabora l'immagine **cover**;
- $H$ è una funzione che elabora il messaggio da inserire, ad esempio una funzione criptografica.

# *Modelli steganografici*

Lo schema di base della steganografia presuppone l'esistenza di due messaggi:

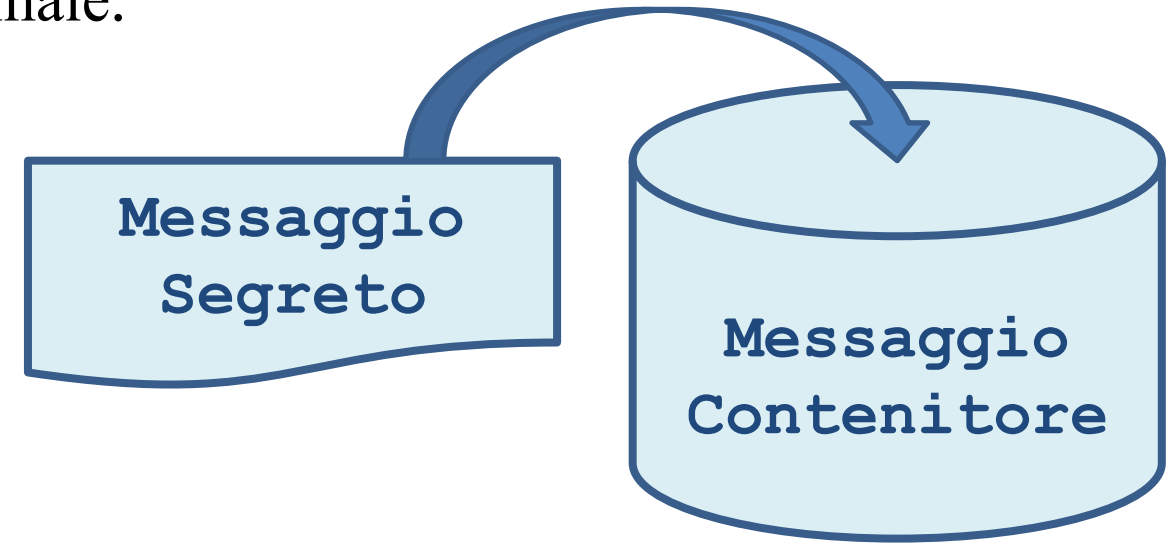
- Messaggio **segreto**
- Messaggio **contenitore**

In base all'origine del contenitore è possibile distinguere:

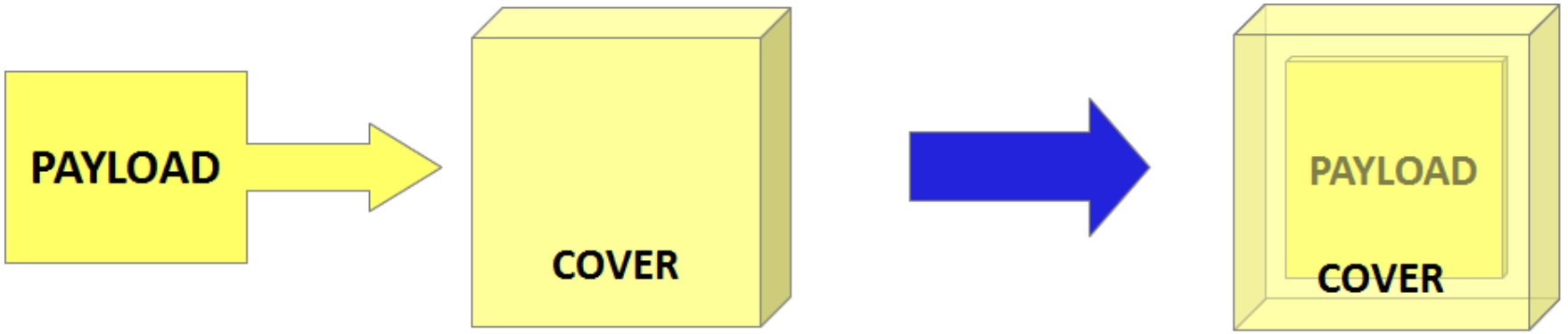
- Steganografia **iniettiva**
- Steganografia **generativa**

# *Steganografia iniettiva*

La steganografia **iniettiva** è la più utilizzata e consente di inserire il messaggio segreto all'interno di un messaggio contenitore già esistente modificandolo in modo tale sia da contenere il messaggio segreto, sia da risultare, al livello al quale viene percepito dai sensi umani, praticamente indistinguibile dall'originale.

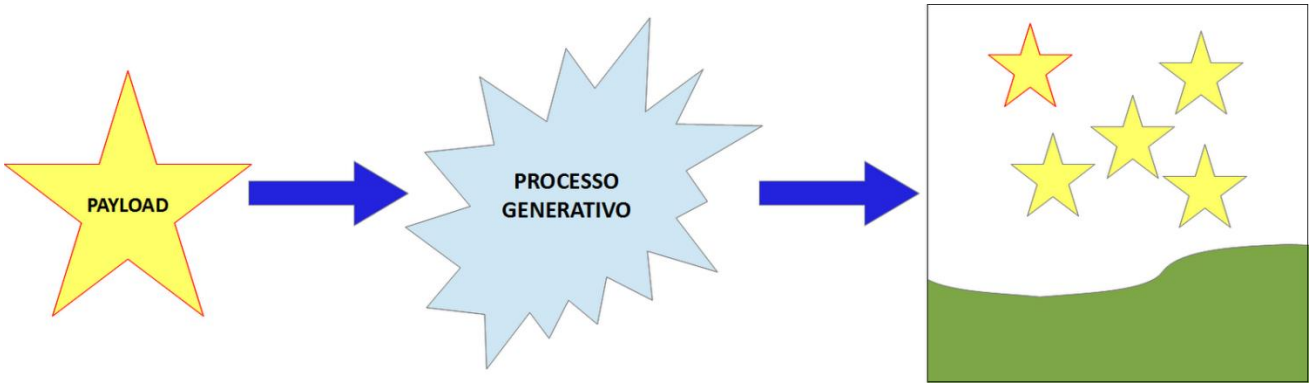


# *Steganografia iniettiva*



# *Steganografia generativa*

La steganografia **generativa** consente di generare, a partire dal messaggio segreto, un messaggio contenitore atto a nascondere nel migliore dei modi quel messaggio segreto.



# *Un'altra classificazione*

Oltre alla classificazione precedente, di carattere prettamente concettuale, ne esiste un'altra che caratterizza le tecniche steganografiche a livello pratico:

- Steganografia **sostitutiva**
- Steganografia **selettiva**
- Steganografia **costruttiva**

# *Steganografia sostitutiva*

È la tecnica steganografica più diffusa, tanto che spesso quando si parla di steganografia ci si riferisce implicitamente a quella di questo tipo.

Tale tecnica si basa sull'osservazione che la maggior parte dei canali di comunicazione (linee telefoniche, trasmissioni radio, ecc.) trasmettono segnali che sono sempre accompagnati da qualche tipo di **rumore**.

Questo rumore può essere sostituito da un **segnale** (il **messaggio segreto**) che è stato trasformato in modo tale che, a meno di conoscere una chiave segreta, è indistinguibile dal rumore vero e proprio, e quindi può essere trasmesso senza destare sospetti.

# *Steganografia sostitutiva (cont.)*

La tecnica impiegata è concettualmente molto semplice, consiste nel sostituire i **bit meno significativi** (LSB least significant bit) dei file digitalizzati con i bit che costituiscono il messaggio segreto.

Quello che succede quindi è che il file contenitore risultante, dopo un'iniezione steganografica, si presenta in tutto e per tutto **simile** all'originale, con differenze difficilmente percettibili e quindi, a meno di confronti approfonditi con il file originale (non effettuabili ad occhio nudo) è difficile dire se le eventuali perdite di qualità siano da imputare al rumore od alla presenza di un messaggio segreto.

# *Steganografia selettiva*

Ha un valore puramente teorico e non viene realmente utilizzata nella pratica. Ne fanno parte le tecniche che mirano a scegliere il supporto a seconda del messaggio da occultare. Viene effettuata una selezione dei supporti a disposizione o si usano semplici algoritmi per generarli, procedendo per tentativi finché non vengono rispettate particolari condizioni.

Un esempio di tecnica selettiva è quella di impostare una funzione *hash* che controlli la parità dei bit del file digitale contenitore, assumendo il valore 1 se il cover contiene un numero di bit uguali ad 1 dispari, e il valore 0 se ne contiene un numero pari. A questo punto, volendo codificare il bit 0, acquisendo il file binario si controlla se il numero di bit uguali ad 1 sia pari, in caso affermativo si è trovato un file adatto a contenere l'informazione codificata, altrimenti si procede con l'acquisire un altro file digitale.

# *Steganografia selettiva (cont.)*

Questa tecnica ha il pregio di risultare praticamente impossibile da identificare, in quanto il supporto, nonostante contenga effettivamente un messaggio segreto, non risulta assolutamente modificato.

Purtroppo ha il difetto di rivelarsi una soluzione alquanto dispendiosa in termini di tempo e insoddisfacente dal punto di vista dell'esiguo quantitativo di dati segreti che permette di celare.

Infatti, dovendo aumentare il numero di bit da nascondere, aumenta anche il tempo per il reperimento o la generazione del supporto, il quale è costretto a soddisfare più vincoli contemporaneamente.

# *Steganografia costruttiva*

Opera più o meno come la steganografia sostitutiva, con la differenza che nel modificare il file contenitore si tiene conto di un modello di rumore, nel senso che si tenta di sostituire il rumore presente con il messaggio segreto nel rispetto delle caratteristiche statistiche del rumore originale.

Secondo questa concezione, un buon sistema steganografico dovrebbe basarsi su un modello del rumore e adattare i parametri dei suoi algoritmi di codifica in modo tale che il falso rumore contenente il messaggio segreto sia il più possibile conforme al modello di partenza.

Questo approccio sembra la soluzione migliore, ma in realtà anch'esso non è esente da difetti.

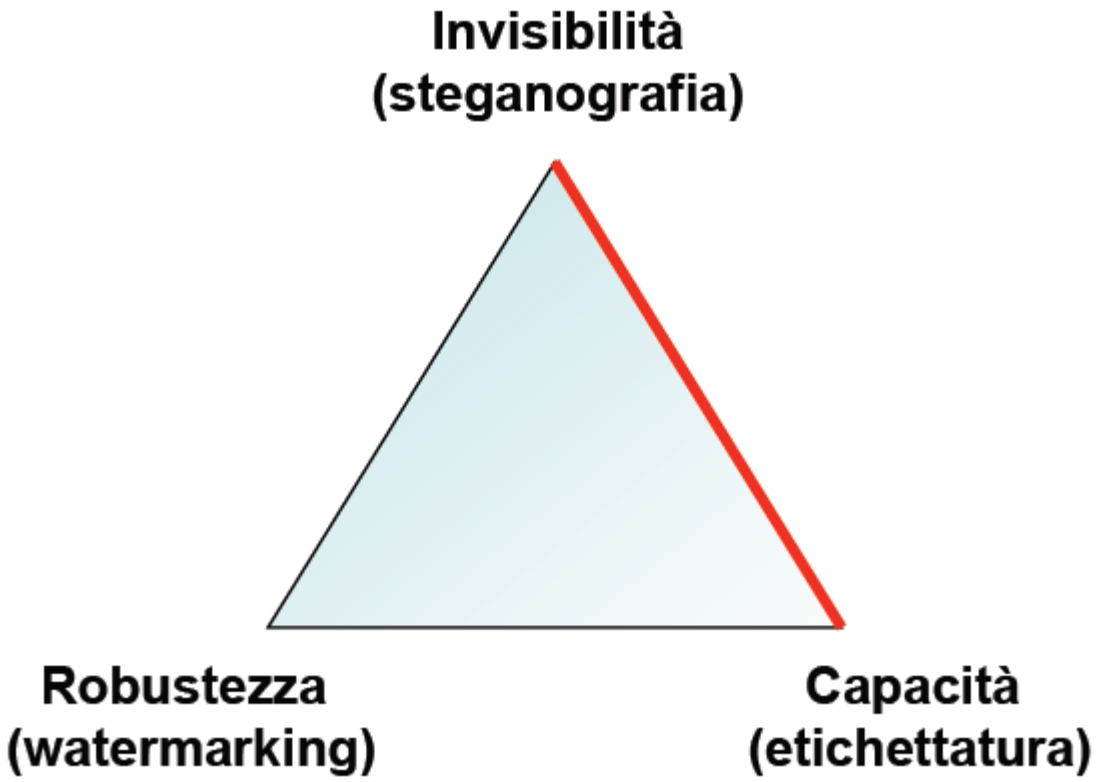
# *Steganografia costruttiva (cont.)*

Innanzitutto non è facile costruire un modello accurato del rumore. La costruzione di un modello del genere richiede grossi sforzi ed è probabile che qualcuno, in grado di disporre di maggior tempo e di risorse migliori, riesca a costruire un modello più accurato riuscendo ancora a distinguere tra il rumore originale e un sostituto.

Inoltre, se il modello del rumore utilizzato dal metodo steganografico dovesse cadere nelle mani del "nemico" egli lo potrebbe analizzare per cercarne possibili difetti e quindi utilizzare proprio il modello stesso per controllare che un messaggio sia conforme ad esso. Così, il modello, oltre ad essere parte integrante del sistema steganografico, fornirebbe involontariamente uno strumento di attacco estremamente efficace proprio contro lo stesso sistema.

# *Analisi*

- In una qualsiasi applicazione di data-hiding si devono fare i conti con tre requisiti in contrasto tra loro



# Invisibilità Percettiva

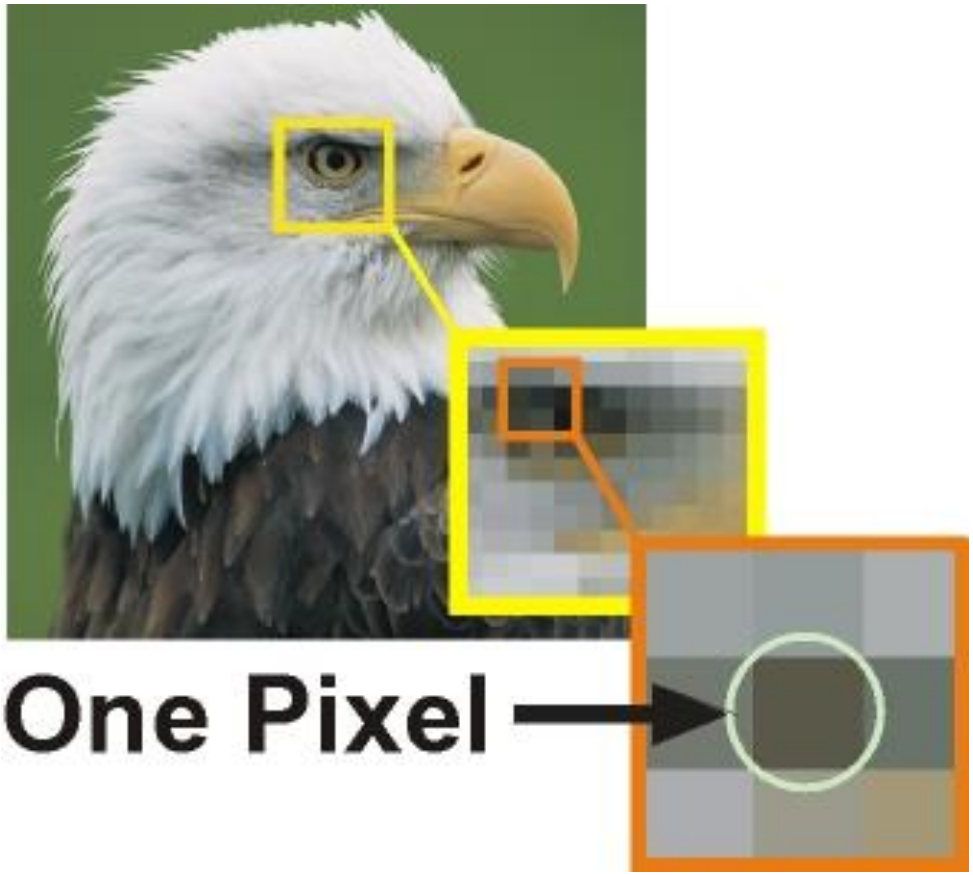
La tecnica di steganografia iniettiva su immagini sicuramente più diffusa anche grazie alla sua relativa facilità di implementazione è quella che si basa sulla modifica del bit meno significativo (LSB).


Immagine cover

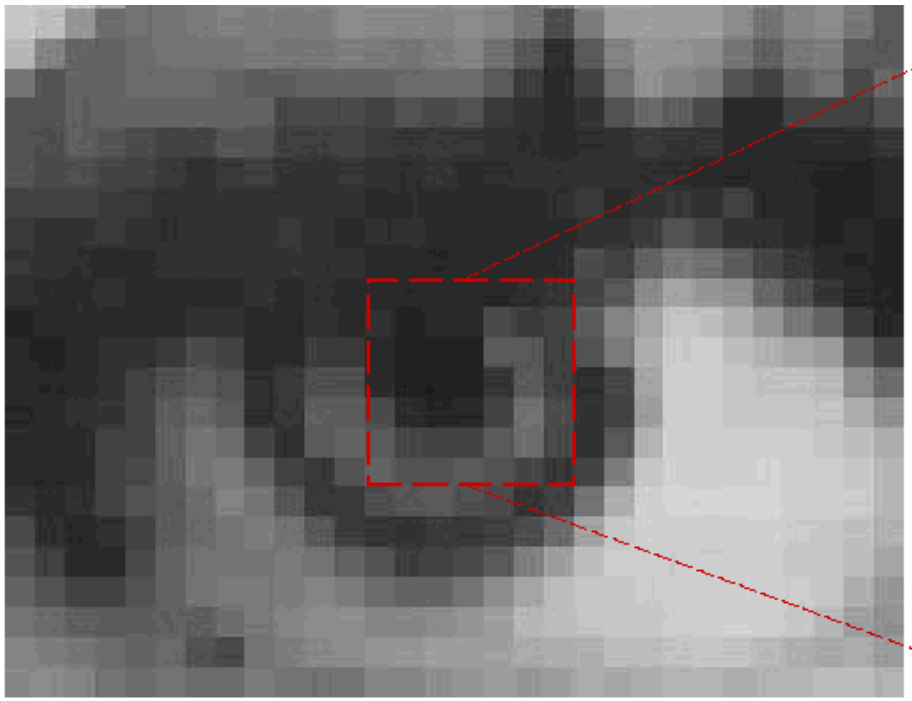

LSB del canale verde (originale)


LSB del canale verde (stego-immagine)

# *Immagini Digitali*



# *Dal pixel al BIT*



| | | | | | | |
|---|---|---|---|---|---|---|
| 46 | 44 | 46 | 47 | 44 | 50 | 56 |
| 50 | 40 | 46 | 48 | 76 | 62 | 69 |
| 45 | 39 | 36 | 40 | 88 | 87 | 65 |
| 48 | 38 | 36 | ? | 56 | 90 | 65 |
| 76 | 48 | 41 | 43 | 69 | 112 | 77 |
| 90 | 72 | 66 | 66 | 90 | 108 | 74 |
| 98 | 86 | 86 | 91 | 83 | 72 | 57 |

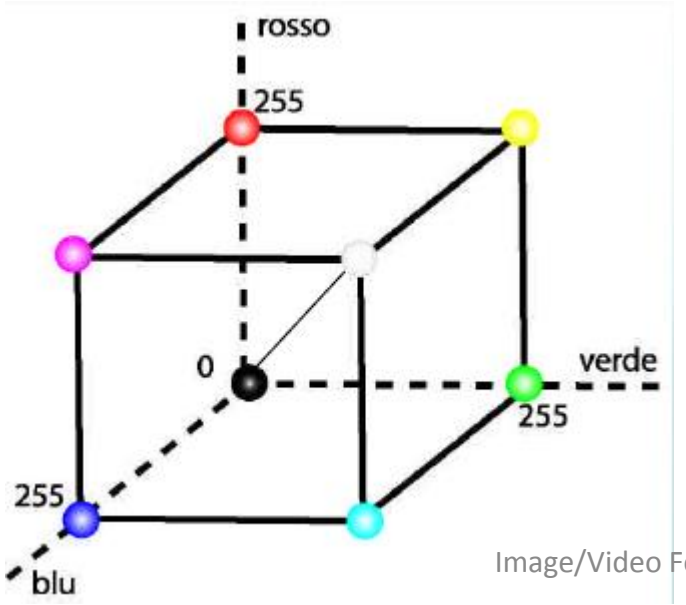
Immagini: Matrici di valori interi compresi tra  0 e 255

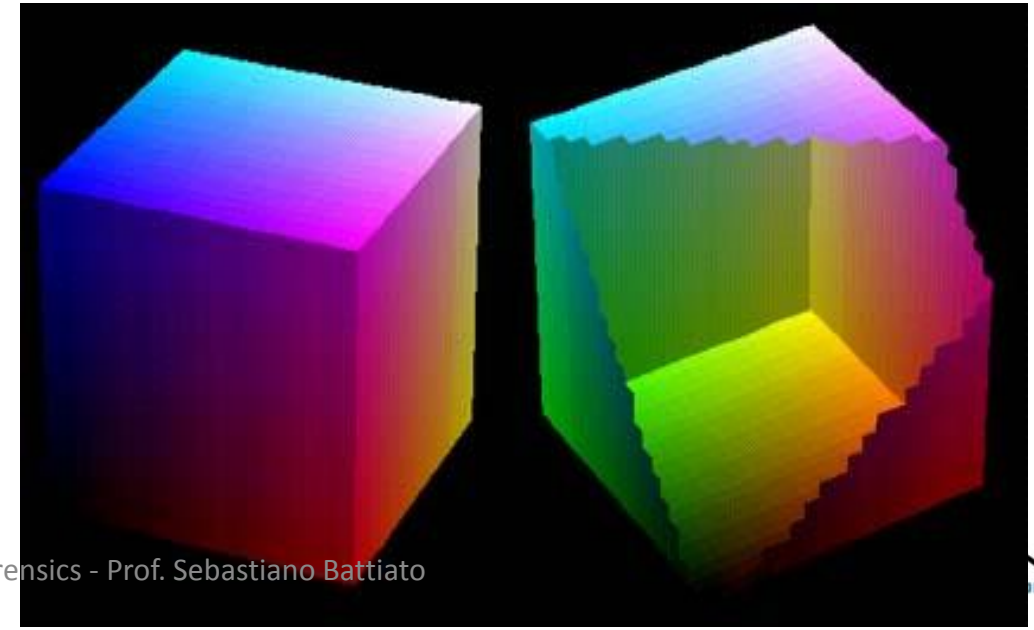
# *Livelli di grigio*

# *RGB*

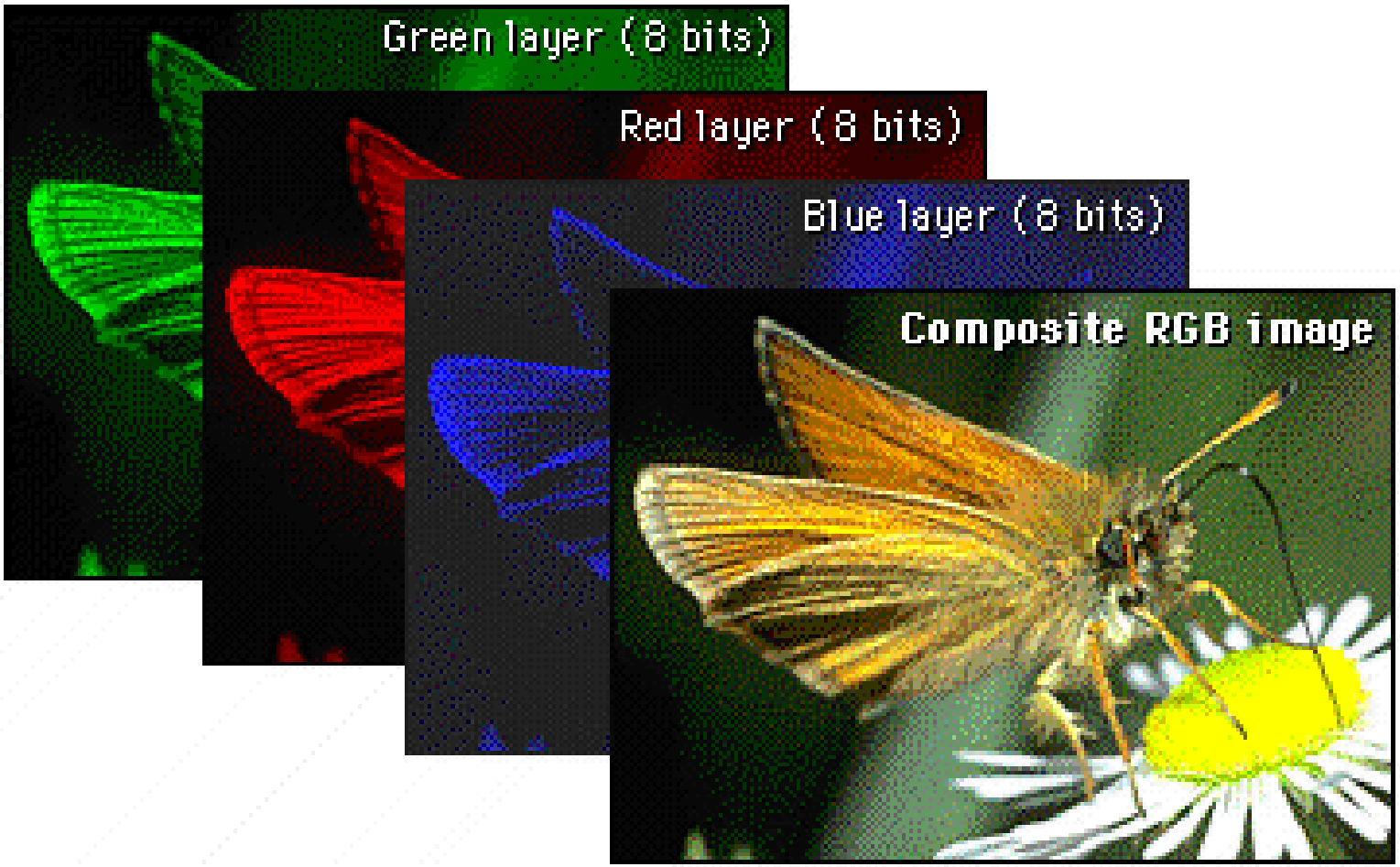
E’ molto comune descrivere i colori riferendosi allo spazio colore RGB (red, green, e blue). Lo spazio RGB è basato sul fatto che ogni colore possa essere rappresentato da una “miscela” dei tre colori primari red, green, e blue. I vari contributi sono assunti indipendenti l’uno dall’altro (e quindi rappresentanti da direzioni perpendicolari tra loro). La retta che congiunge nero e bianco è la retta dei grigi.

RGB è molto usato nelle videocamere e nei monito dato che risulta essere lo spazio colore più semplice per registrare e visualizzare immagini digitali a colori.



# *TrueColor (24 bit)*



Esempi di codifica digitale di immagini a colori:
(R->rosso, G->verde, B->Blu)

| | | |
|---|---|---|
| | colore bianco | (R=255, G=255, B=255) |
| | colore grigio | (R=200, G=200, B=200) |
| | colore grigio scuro | (R=100, G=100, B=100) |
| | colore nero | (R=0, G=0, B=0) |
| | colore rosso | (R=255, G=0, B=0) |
| | colore giallo | (R=255, G=255, B=0) |
| | colore azzurro | (R=0, G=255, B=255) |
| | colore verde scuro | (R=0, G=200, B=0) |

# *Depth Resolution*

8 bit – 256 Gray levels

4 bit – 16 colors

…

24 bit True colors

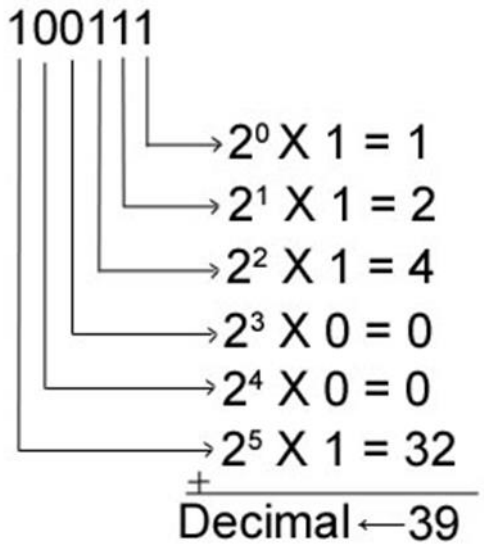
# *How many colors?*

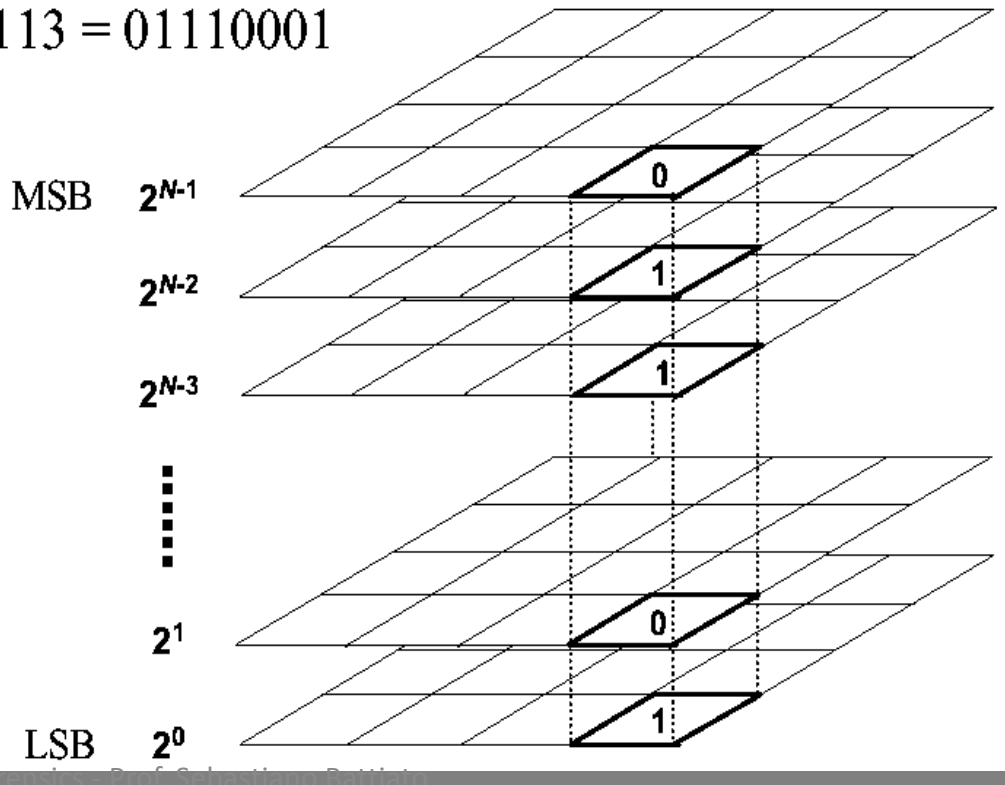
| Bit | # colors | |
|---|---|---|
| 1 | $2^1$ | 2 |
| 2 | $2^2$ | 4 |
| 3 | $2^3$ | 8 |
| 4 | $2^4$ | 16 |
| 5 | $2^5$ | 32 |
| 6 | $2^6$ | 64 |
| 7 | $2^7$ | 128 |
| 8 | $2^8$ | 256 |
| 16 | $2^{16}$ | 65.536 (16 bit True Color) |
| 24 | $2^{24}$ | 16.777.216 (True Color) |
| 32 | $2^{32}$ | 24 bit True-Color + 8 bit Alpha Channel |

# *Bit-planes*

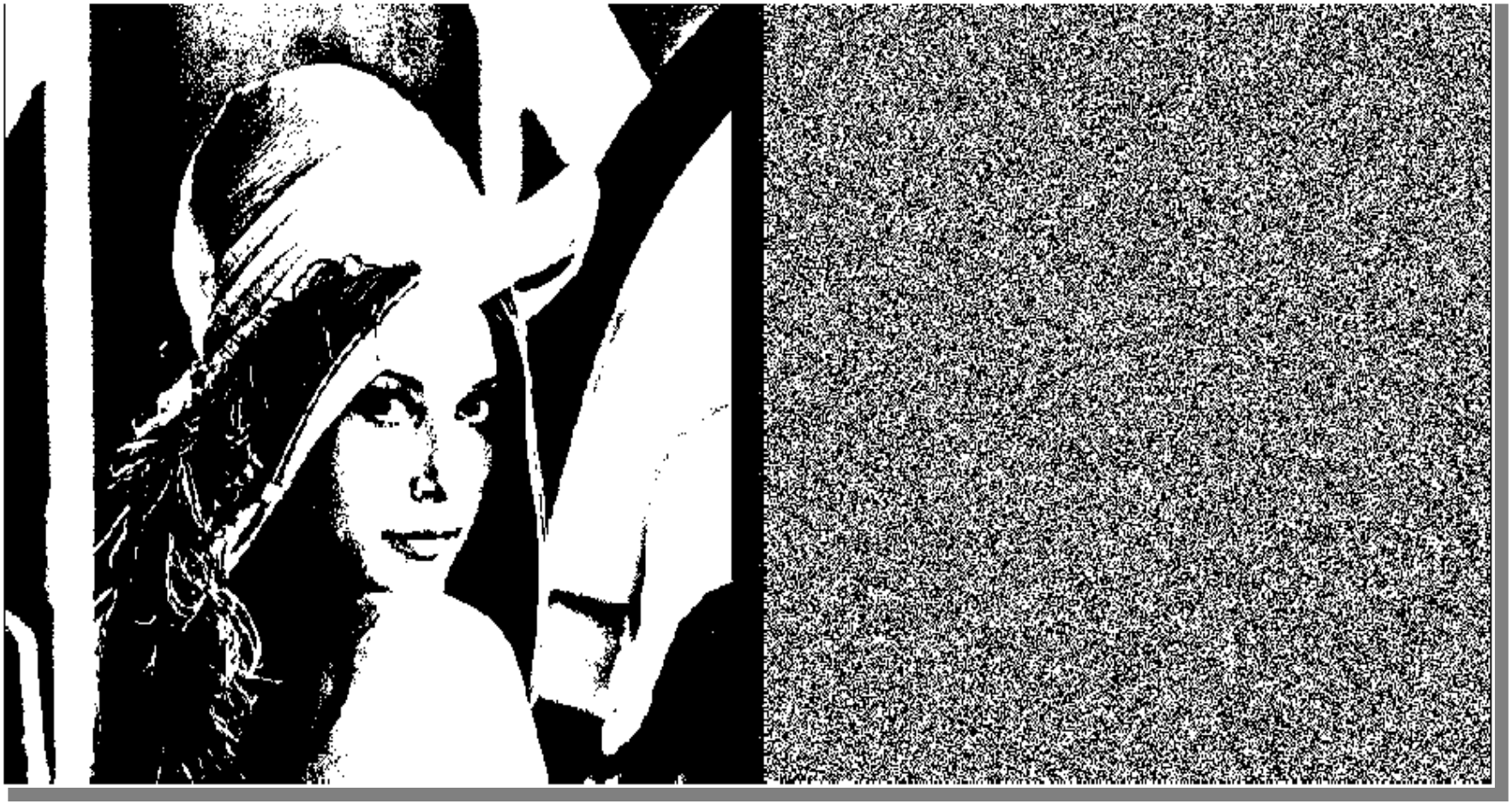
Un'immagine con una profondità colore di N bit può essere rappresentata da N piani di bit (bit-planes), ciascuno dei quali può essere vista come una singola immagine binaria. In particolare si può indurre un ordine che varia dal **Most Significant Bit** (**MSB**) fino al **Least Significant Bit** (LSB).

$$x(k,l) = 113 = 01110001$$



# *Lena: Bit-planes*

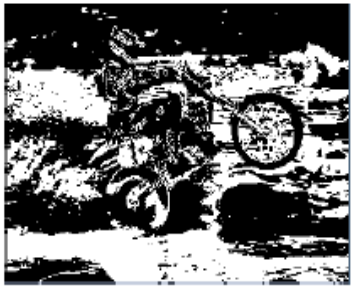


Most Significant bit (**MSB**)

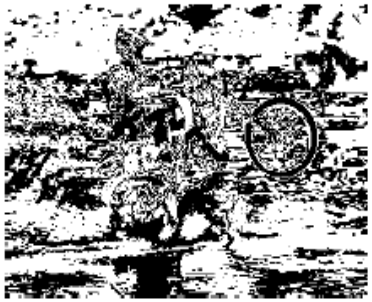
Least Significant bit (**LSB**)

# *Bit-planes*

7

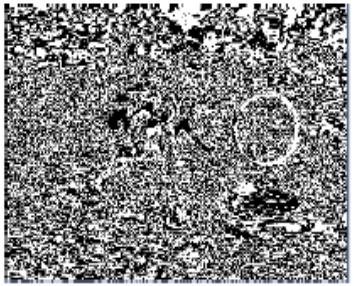


6

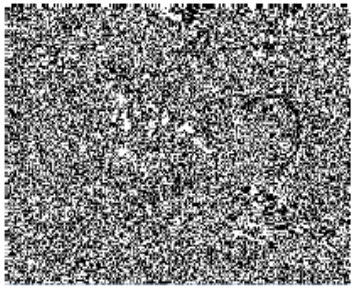


5

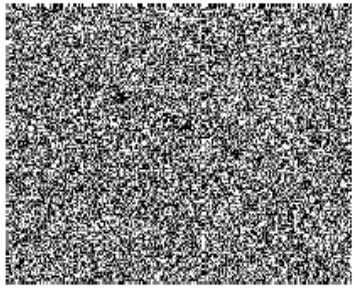
4



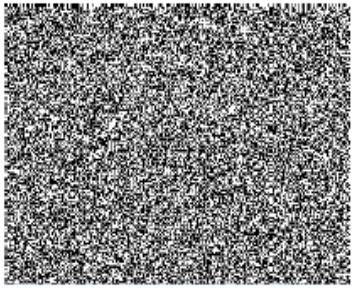
3



2



1



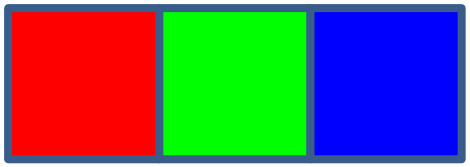
0

# *Steganografia nelle immagini digitali (.bmp)*

Supponiamo di voler utilizzare come contenitore un file di tipo bitmap con una profondità di colore a 24 bit, quindi una matrice MxN di pixel codificata in modalità RGB.

Quindi per esempio un file bitmap a 24 bit di dimensione 640x480 occuperà uno spazio di 640x480x3=921600 byte.

**Una possibile operazione di steganografia sostitutiva su questi file consiste nel sostituire i bit meno significativi dei singoli byte (LSB).**



# *Steganografia nelle immagini digitali (.bmp)*

Se , ad esempio, avessimo un pixel codificato nel modo seguente:

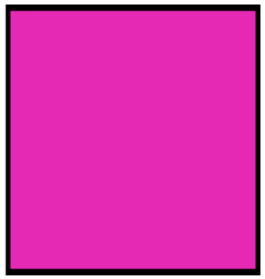
**11100001 00000100 00010111**

E volessimo codificare: **110** Il pixel diventerà:

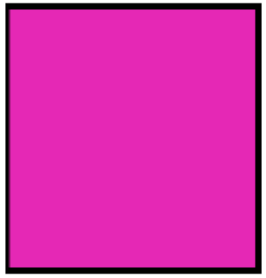
**11100001 00000101 00010110**

Queste semplici operazioni fanno si che le variazioni di colore siano praticamente impercettibili ad occhio nudo.

# *Steganografia nelle immagini digitali (.bmp)*



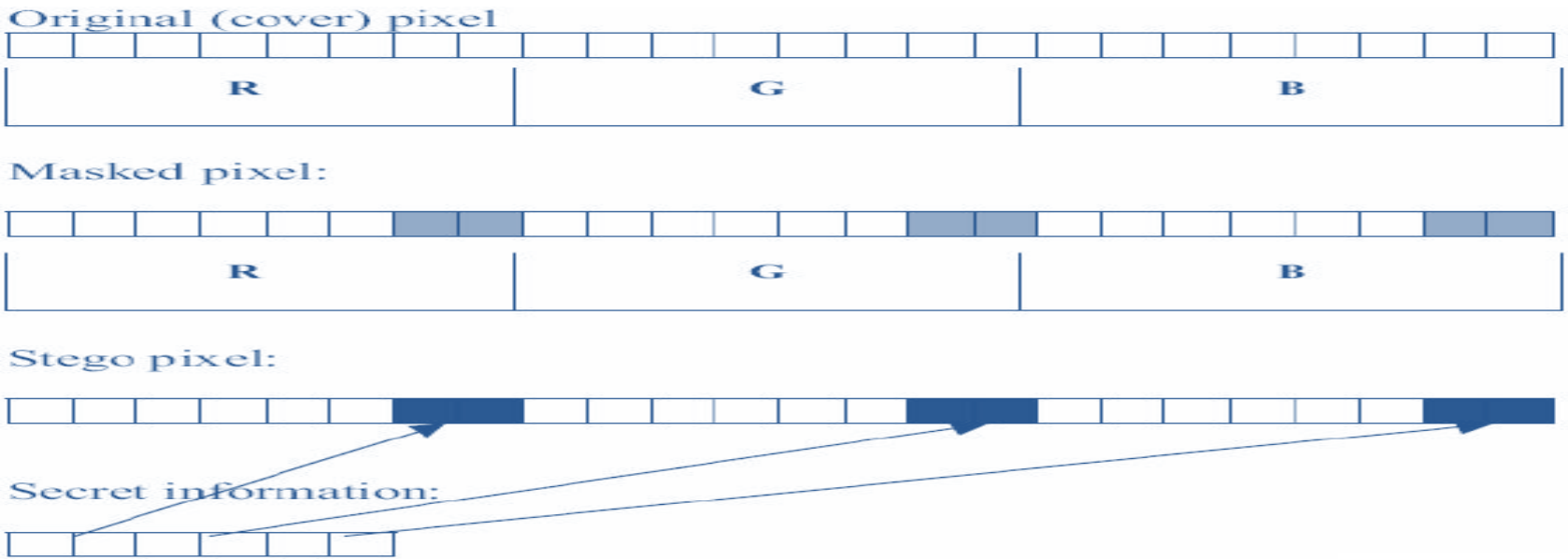
R: 11100101
G: 00101001
B: 10110100



R: 11100101
G: 00101000
B: 10110101

# *Steganografia nelle immagini digitali (dimensione del messaggio)*

Quindi dato che un solo pixel può contenere un'informazione segreta di 3 bit, un'immagine di dimensione MxN può contenere un messaggio segreto lungo fino a (MxNx3)/8 byte.



# *Steganografia nelle immagini digitali (esempio)*

Pixel originale:

**11100001 10011001 10110001**

Carattere “**A**”: **01000001**

Pixel sostituito:

**11100001 10011000 10110000**

In questo caso si potranno inserire: (640x480x3x2)/8 = 230400 byte (cioè 230400 caratteri)

N.B: Gli ultimi due bit del carattere “A” (**01**) verranno inseriti nel pixel successivo.

# *Steganografia nelle immagini digitali (esempio .bmp)*

In questo esempio e stato inserito in un'immagine BMP un file di testo TXT contenente il primo canto dell'inferno della divina commedia.

Immagine originale .bmp

Dimensione:1,19 MB
(1.254.214 byte)

Immagine steganografata .bmp

Dimensione:1,19 MB
(1.254.214 byte)

# *Steganografia nelle immagini digitali (.gif)*


Il formato GIF è un formato molto utilizzato per i siti web perché è poco ingombrante. Si basa su una palette di 256 colori. Un file GIF è una sequenza di puntatori alla palette (uno per ogni pixel).

Per iniettare un messaggio segreto in un file GIF, acquisita l'immagine, bisogna:

1. Decrementare il numero di colori ad un numero inferiore a 256 con un opportuno algoritmo che limita la perdita di qualità;
2. Convertire in GIF riempiendo la palette con colori molto simili a quelli rimasti.

# *Steganografia nelle immagini digitali (.gif)*

Dopo un'operazione di questo tipo ogni pixel potrà essere rappresentato alternativamente con il colore originale o con il relativo colore simile aggiunto.

Quindi in presenza di alternative possiamo nascondere un'informazione. Per esempio se le alternative sono due potremo nascondere un bit (se il bit è 0, scegliamo la prima alternativa, se è 1 la seconda). Se le alternative sono quattro potremo nascondere due bit (00 per la prima, 01 per la seconda, 10 per la terza, 11 per la quarta) e così via.

Ad esempio se per rappresentare il pixel *p(i,j)* si hanno due alternative $C_1$ e $C_2$ ,si potrà nascondere un bit, associando il valore 0 a $C_1$ ed il valore 1 a $C_2$. Il numero di bit rappresentabili aumenterà di uno al raddoppiare dei colori sostituiti.

# *Steganografia nelle immagini digitali (altra soluzione .gif)*

La soluzione esposta è senz'altro molto ingegnosa, ma presenta il problema che è molto semplice scrivere un programma che analizzi la palette ed individui sottoinsiemi di colori simili e quindi la probabile presenza di un messaggio steganografato. In effetti, questo tipo di attacco è stato portato a termine con pieno successo da diversi steganalisti, tanto che alcuni di loro hanno sostenuto che il formato GIF non fosse adatto alla steganografia.

In realtà esiste un altro metodo per steganografare con GIF che si basa sulla seguente osservazione: un immagine GIF può essere rappresentata in 256! modi diversi.

# *Steganografia nelle immagini digitali (altra soluzione .gif)*

La palette di una GIF si compone di 256 colori, tuttavia non è importante **l'ordine** in cui i colori compaiono nella palette e quindi i 256 colori di una palette possono essere permutati in 256! modi, ciò vuol dire che una stessa immagine GIF può essere rappresentata in 256! modi diversi, a patto di cambiare opportunamente la sequenza dei puntatori.

La teoria dell'informazione afferma e dimostra che l'informazione è una quantità misurabile, ed è direttamente proporzionale al numero di simboli dell'alfabeto del linguaggio che viene usato per comunicare. Nel nostro caso, permutando opportunamente la palette, si potranno avere 256! rappresentazioni - cioè simboli - alternative della stessa immagine, per un totale di log(256!) = 1683 bit (circa 200 byte) disponibili per la codifica di un messaggio nascosto, indipendentemente dalla dimensione dell'immagine

# *Steganografia nelle immagini digitali (esempio .gif)*

Immagine mappa segreta .gif

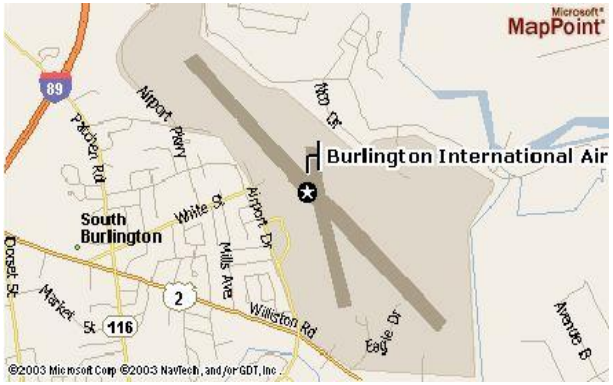
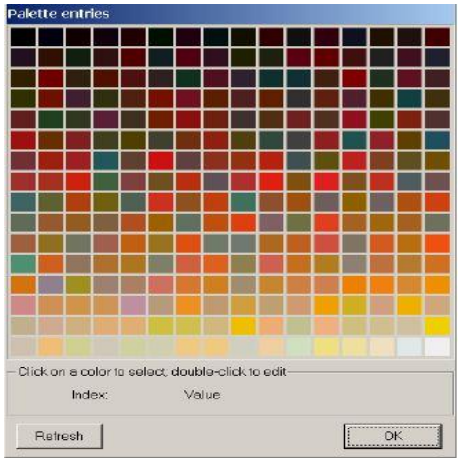



Immagine contenitore .gif

Palette immagine contenitore prima




Palette immagine contenitore dopo

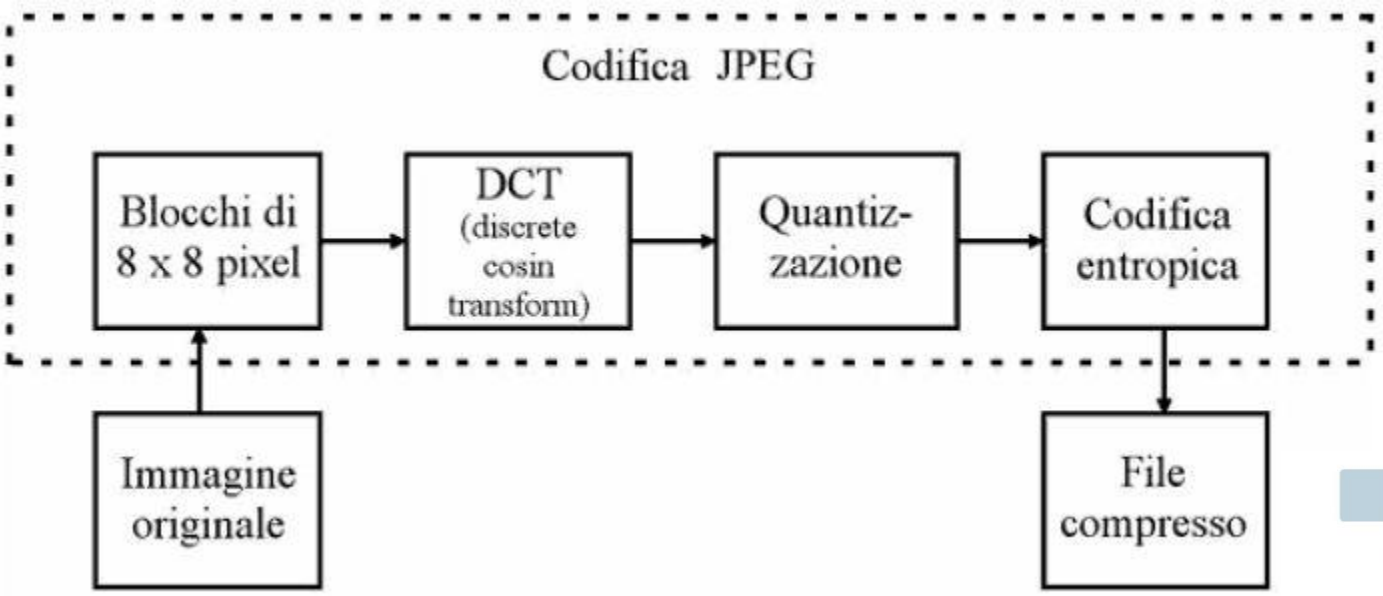
# *Steganografia nelle immagini digitali (Lossless vs. Lossy)*

I formati considerati fino ad adesso sono tutti formati **Lossless** (compressione dati senza perdita) nel caso si dovessero considerare i formati **Lossy** (compressione con perdita) non è possibile operare come sinora descritto. In particolare, se iniettassimo delle informazioni in un file bitmap e dopo lo convertissimo ad esempio, in JPEG,, le informazioni andrebbero inevitabilmente perse.

La compressione JPEG, infatti, ha la tendenza a preservare le caratteristiche visive dell'immagine piuttosto che l'esatta informazione contenuta nella sequenza di pixel, di conseguenza sarebbe impossibile risalire al file bitmap originario.

# *Steganografia nelle immagini digitali (.jpg)*

In questi casi si opera ad un livello di rappresentazione intermedio. Per poter utilizzare anche le immagini JPEG come contenitori, è possibile iniettare le informazioni nei **coefficienti di Fourier** ottenuti dalla prima fase di compressione.



# *Steganografia nelle immagini digitali (.jpg)*

Le tecniche steganografiche solitamente vengono applicate dopo la fase di quantizzazione e sono caratterizzare dall'eseguire un'alterazione dei coefficienti DCT al fine di occultare informazioni segrete.

Bisogna innanzitutto tenere presente che la modifica di un singolo coefficiente DCT in un blocco ha effetto su tutti e 64 pixel dell'immagine appartenenti ad esso. Inoltre, la scelta di quali coefficiente modificare deve essere ponderata in funzione del tipo di protezione necessaria.

- Coeff. alte frequenze: + invisibilità - robustezza;
- Coeff. basse frequenze: + robustezza - invisibilità;
- Coeff. medie frequenze:  invisibilità = robustezza. (solitamente)

# *Steganografia nelle immagini digitali (.jpg)*

La principale modalità di occultamento prevede la modifica dei bit meno significativi (LSB) dei coefficienti DCT per inserirvi i bit del messaggio segreto.

Un'altra procedura alquanto efficace sfrutta e pilota l'operazione di arrotondamento all'intero dei coefficienti DCT, nella fase di quantizzazione, per occultare informazione. L'inserimento dei dati segreti viene attuato scegliendo opportunamente di arrotondare i coefficienti all'intero superiore o inferiore.

In un sistema più complesso, invece, l'occultamento dei bit del file segreto viene codificato nella differenza relativa tra coefficiente che corrispondono a locazioni di uguale valore nella tabella di quantizzazione. Se tale differenza non eguaglia il bit da nascondere allora i coefficienti vengono scambiati tra loro.

# *Steganografia nelle immagini digitali (esempio .jpg)*

In questo esempio e stato inserito in un'immagine JPEG in un'altra immagine JPEG.

Immagine segreta .jpg

Immagine contenitore .jpg

Immagine steganografata .jpg

# *Steganografia nelle immagini digitali (BPCS)*

Invece di considerare sempre e solo i bit meno significativi come bit da sostituire, è possibile analizzare l'immagine e scegliere delle regioni nelle quali effettuare una modifica non significa alterare in modo significativo l'immagine nel complesso. Una tecnica che effettua questo lavoro è la BPCS Steganography sviluppata da Eiji Kawaguchi nel 1997.

Per determinare tali regioni sostituibili l'immagine viene divisa in blocchi, per ogni blocco di 8x8 pixel viene effettuato un test che determina la complessità di immagine contenuta in questo blocco. Se tale complessità è minore di una determinata soglia (parametro variabile dipendente dall'immagine) allora un messaggio segreto (eventualmente cifrato) può essere nascosto in questo blocco senza alterare significativamente l'immagine.

# *Steganografia nelle immagini digitali (BPCS)*

Un'immagine **P** costituita da pixel ad n-bit può essere decomposta in un insieme di n immagini binarie.

Per esempio, se l'immagine è una immagine n-bit gray, la possiamo descrivere come: **P=(P1,P2,...,Pn)**

Un'immagine RGB P, invece, può essere vista come:

**P=(PR1,PR2,..,PRn; PG1,PG2,..,PGn; PB1,PB2,..,PBn)**

Dove **PR1, PG1, PB1** è la bit-plane più significativa (immagine formata dai bit più significativi di tutti i pixel dell'immagine) e **PRn, PGn, PBn** è la bit-plane meno significativa.

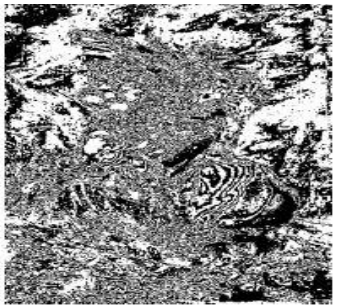
Analizzando ogni singola bit-plane, quello che accade molto di frequente è che più ci si allontana dalla bit-plane più significativa, più aumenta la complessità (intesa come confusione dell'immagine) della bit-plane stessa.

# *Steganografia nelle immagini digitali (BPCS)*

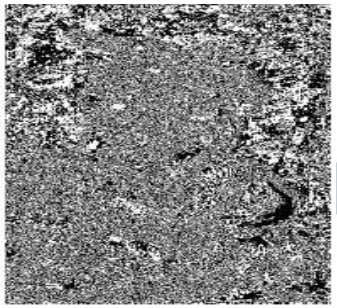
Ogni bit-plane può essere segmentata in regioni *shape-informative* (forma informativa, cioè parti dell'immagine che sono significative sotto il punto di vista visivo) e *noise-looking* (disturbo visivo, cioè poco rilevanti dal punto di vista visivo). Le regioni semplici (cioè ben distinguibili visivamente) sono delle *shape-informative* e pertanto non possono essere modificate, quelle complesse, invece, rappresentano delle *noise-looking* che possono essere rimpiazzate senza deteriorare la qualità dell'immagine nel complesso.

**PR3** contiene più zone *shape-informative* e **PR5** più *noise-looking*. **PR4**, invece, è un mix di regioni *shape-informative* e *noise-looking*.

**PR3**



**PR4**



**PR5**

# *Steganografia nelle immagini digitali (esempio)*

In quest'esempio viene illustrato il risultato ottenuto con la tecnica BPCS impiegando un'immagine contenitore, a 24 bit di dimensione 617x504 pixel pari a 932 KB di spazio in memoria, e l'informazione segreta costituita da un'immagine e vari documenti di testo. Nonostante la considerevole quantità celata, l'immagine contenitore e l'immagine steganografata risultano praticamente indistinguibili l'una dall'altra.

Immagine contenitore

Immagine steganografata

# Cenni Sulla Steganografia Su File Video

- Un file video è un file multimediale in senso stretto. Infatti esso contiene il video (cioè una sequenza di immagini), audio (sia dialoghi che musica) e testo (sottotitoli, titoli di testa e di coda, scritte in sovraimpressione). Inoltre i file video vengono compressi a causa delle grandi dimensioni, utilizzando numerosi schemi di codifica/decodifica (CODEC).
- Sfruttando la poliedricità dei file video, è possibile inserire messaggi nascosti all'interno di una qualunque delle sue parti (flusso di immagini statiche, audio, testi), utilizzando gli efficienti algoritmi già noti per ciascuna parte. L'uso dei file video come cover consente di avere più bit a disposizione per il messaggio nascosto, e rende più complicato l'attacco: infatti il messaggio potrebbe essere nascosto in uno qualunque tra gli elementi del video, o addirittura suddiviso tra di essi.

# Steganalisi

- Le tecniche che si prefiggono di individuare la presenza di un messaggio occultato attraverso l'uso di tecniche steganografiche rientrano sotto il nome di **steganalisi**. L'obiettivo almeno in prima istanza non è quello di individuare e decodificare il contenuto del messaggio segreto, ma semplicemente determinare se un mezzo contiene un messaggio oppure no.

# Steganalisi

- Pertanto la steganalisi può essere formulata come un test sull'ipotesi che il mezzo contenga un messaggio segreto. Formalmente, dato un insieme di osservazioni Y={y1,y2,…,yn}, o di loro funzioni dette *feature o statistiche* si formulano due ipotesi alternative:
  - **Il file** non contiene un messaggio segreto;
  - **Il file** contiene un messaggio segreto.
- La decisione tra le due ipotesi viene presa in base ad un **criterio di ottimalità**.

# Attacchi

- Il tentativo di determinare la presenza di un messaggio segreto è detto **attacco**. Riprendendo l'Equazione 1, è possibile distinguere gli attacchi a seconda delle parti dell'equazione note all'analista. Si avrà quindi:
  - **stego-only-attack**: l'attaccante ha intercettato il frammento stego ed è in grado di analizzarlo. È il più importante tipo di attacco contro il sistema steganografico perché è quello che occorre più di frequente nella pratica;
  - **stego-attack**: il mittente ha usato lo stesso cover ripetutamente per nascondere dati. L'attaccante possiede un frammento stego diverso ma originato dallo stesso cover. In ognuno di questi frammenti stego è nascosto un diverso messaggio segreto;

# Attacchi

- **cover-stego-attack**: l'attaccante ha intercettato il frammento stego e sa quale cover è stato usato per crearlo. Ciò fornisce abbastanza informazioni all'attaccante per poter risalire al messaggio segreto;
- **cover-emb-stego-attack**: l'attaccante ha "tutto": ha intercettato il frammento stego, conosce il cover usato e il messaggio segreto nascosto nel frammento stego;
- **manipulating the stego data**: l'attaccante è in grado di manipolare i frammenti stego. Il che significa che l'attaccante può togliere il messaggio segreto dal frammento stego (inibendo la comunicazione segreta);
- **manipulating the cover data**: l'attaccante può manipolare il cover e intercettare il frammento stego. Questo può significare che con un processo più o meno complesso l'attaccante può risalire al messaggio nascosto.

# *Steganografia nelle immagini digitali (Attacchi visuali e statistici)*

È possibile effettuare degli attacchi alle immagini digitali con lo scopo di rilevare la presenza di dati segreti:

- **Attacchi visuali**: che sono in correlazione con le capacità visive umane. Si sfruttano cioè le capacità dell'occhio umano per individuare artefatti introdotti da tecniche steganografiche;
- **Attacchi statistici**: che effettuano test statistici sui file steganografati.

# *Steganografia nelle immagini digitali (Attacchi visuali)*

Fasi attacco visuale:

- Il file steganografato viene filtrato con un algoritmo di filtering dipendente dalla funzione utilizzata per nascondere il messaggio
- L'immagine filtrata viene osservata per determinare se è stato nascosto un messaggio o meno

L'operazione risulta lenta se la mole di immagini da analizzare è considerevole. Gli algoritmi di filtering effettuano un'operazione sull'immagine steganografata, facendo risaltare visivamente i bit che contengono il messaggio nascosto.

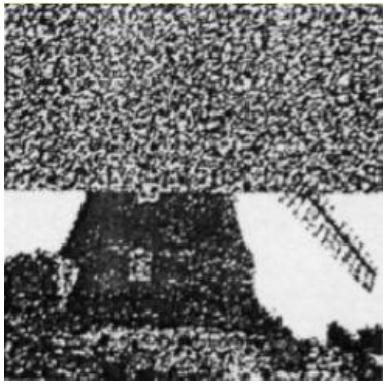
# *Steganografia nelle immagini digitali (Attacchi visuali)*

Immagine contenitore

Immagine steganografata al 50%

Immagine contenitore filtrata

Immagine steganografata filtrata

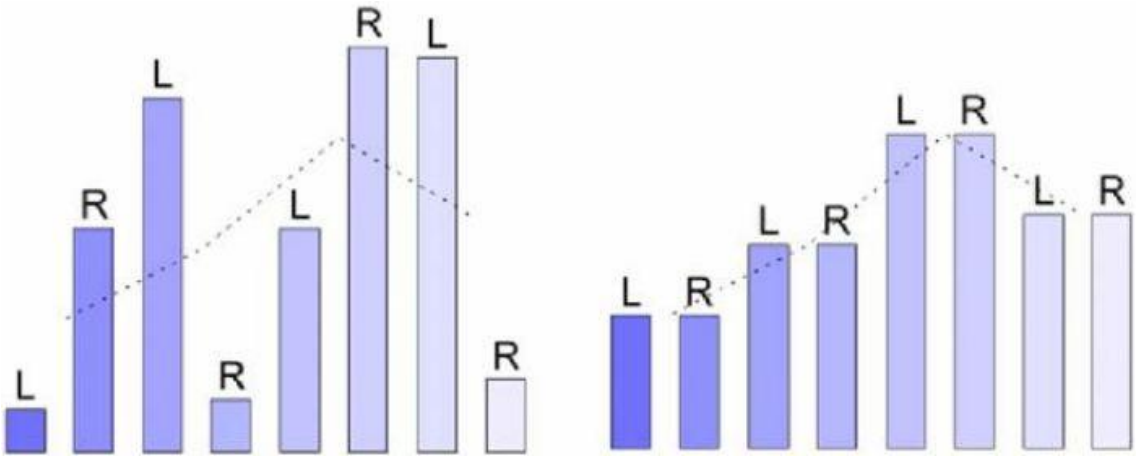


# *Steganografia nelle immagini digitali (Attacchi statistici)*

- L'informazione statistica estratta da un'immagine più nota ed utilizzata, è sicuramente l'istogramma che rappresenta la distribuzione del colore nell'immagine stessa. Si tratta di un semplice grafico a barre, in cui sull'asse delle ascisse ci sono i valori di intensità del colore e sull'asse delle ordinate il numero di pixel che hanno quel valore di intensità. Si può quindi immaginare di utilizzare queste informazioni per definire un modello statistico della distribuzione dei valori attesi di intensità in una immagine stego. Confrontando i valori dell'immagine candidata con il modello statistico teorico, si riesce ad ottenere la probabilità che nell'immagine sia presente un messaggio nascosto.

# *Steganografia nelle immagini digitali (Attacchi statistici)*

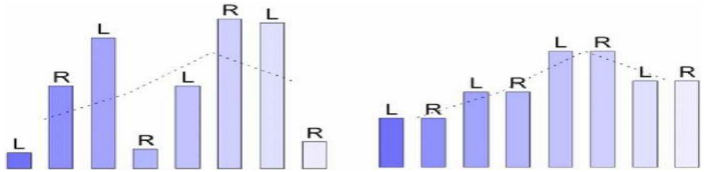
L'idea dell'attacco statistico è di confrontare la distribuzione di frequenza dei colori di un potenziale file steganografato con la distribuzione di frequenza teoricamente attesa per un file steganografato.




Freq. colore prima  Freq. colore dopo

In questo caso dopo l'inserimento le frequenze si eguagliano a due a due: L ed R stanno ad indicare, rispettivamente, l'elemento a sinistra e a destra della coppia dei colori adiacenti considerati.

# *Steganografia nelle immagini digitali (Attacchi statistici)*



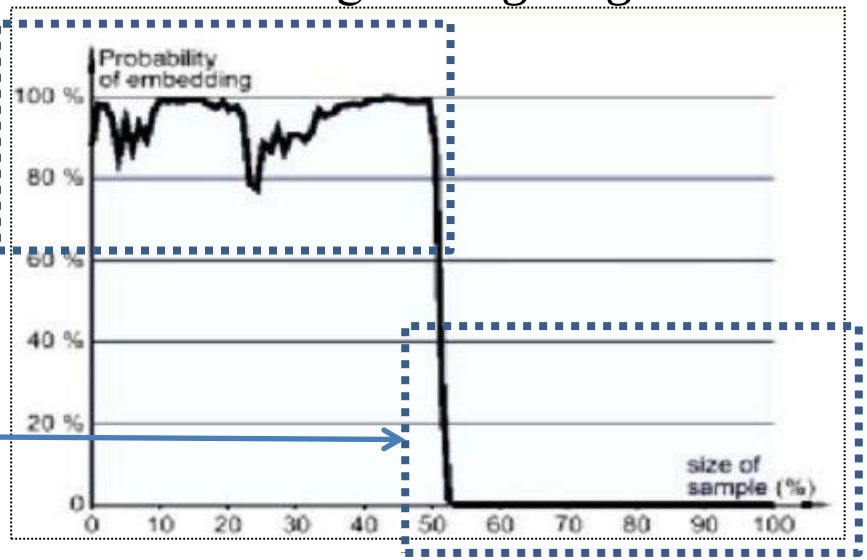

Se la funzione utilizzata è la sovrascrittura del bit meno significativo (LSB), il valore del pixel diventerà uguale a tutti gli altri pixel il cui valore differisce solo per il bit meno significativo. Se i bit da sovrascrivere (cioè il messaggio nascosto) sono equamente distribuiti, le frequenze nell'istogramma relative al valore originale e al valore con il LSB sovrascritto diventeranno uguali

# *Steganografia nelle immagini digitali (Attacchi statistici)*

La figura seguente mostra l'output di un attacco statistico effettuato. L'asse delle ascisse indica la dimensione del campione che è stato analizzato in percentuale. L'asse delle ordinate indica la probabilità di inserimento corrispondente all'area analizzata. È semplice notare che il risultato è molto alto nella prima metà dell'immagine e molto basso per la seconda metà. Questo risultato corrisponde all'attacco visuale terminato con successo nell'esempio visto in precedenza sull'immagine steganografata al 50%.

La prima parte dell'immagine nasconde dati segreti

La seconda parte dell'immagine non nasconde dati segreti



# *Steganografia nell'audio digitale (.wav)*

Un file WAV mono, campionato a 44100 Hz a 16 bit, per esempio, indica un file che è stato costruito ottenendo 44100 stringhe di 16 bit al secondo nella fase di digitalizzazione del suono, ossia è stata generata una stringa di 16 bit ogni 1/44100 di secondo. Nel caso di un wav stereo, le stringhe di 16 bit ottenute sono due, una per il canale destro ed una per il sinistro .

Anche in questo caso si possono sostituire i bit meno significativi allo scopo di steganografare un messaggio.

Esempio: File WAV [44100 Hz, 16 bit, stereo] di un minuto.

Dimensione file =

$$16bit \times 44100Hz \times 60\,\mathrm{sec} \times 2 = 8476200bit \approx 10366Kb$$

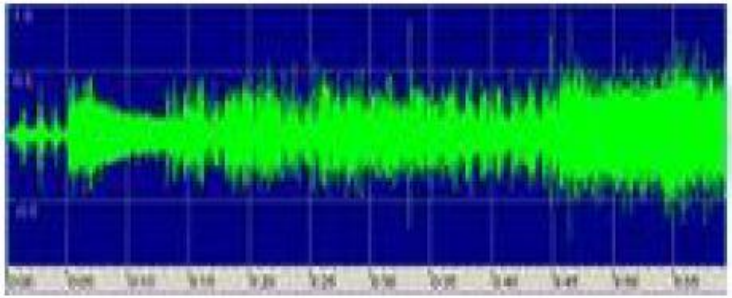
Spazio per file nascosto (utilizzano 2 bit meno significativi) =

$$8476200bit \div 16bit \times 2 = 1059525bit \approx 1293Kb$$

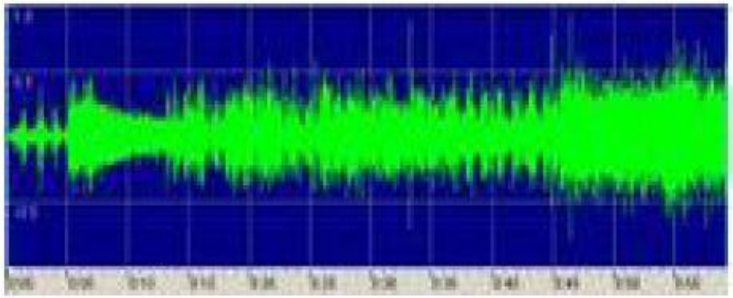
# *Steganografia nell'audio digitale (esempio .wav)*

In questo esempio viene illustrato un file WAV con occultato un file di testo di 128 KB.

File .wav originale



File .wav steganografato



Un punto debole della tecnica LSB sui file audio è rappresentato dal fatto di introdurre solitamente un fastidioso rumore di fondo (noise), avvertibile dall'orecchio umano.

Inoltre, le considerazioni fatte per la tecnica LSB applicate alle immagini valgono anche per i file audio, con la sola differenza che invece di operare sui pixel si agisce sui campioni dell'onda sonora.

# *Steganografia nell'audio digitale (Echo Data Hiding)*

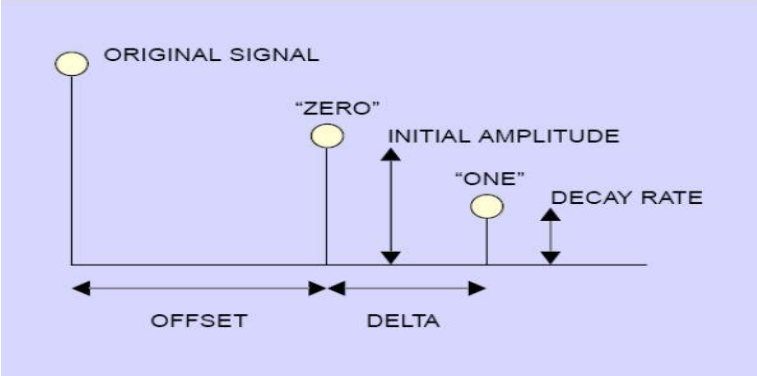
L'approccio visto in precedenza modifica il file aggiungendo un forte rumore di fondo (noise) facilmente avvertibile.

L'**Echo data hiding** è una tecnica che evita questo inconveniente.

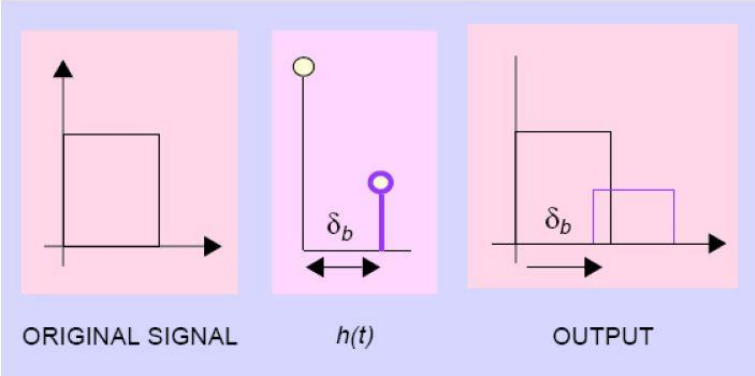
Se il suono originale e la sua eco sono divisi da uno spazio di tempo piccolo abbastanza, l'orecchio umano non riesce a distinguere i due suoni .

I dati vengono codificati in questi eco rappresentando gli 0 e gli 1 come due offsets differenti di eco.

Parametri dell'eco modificabili



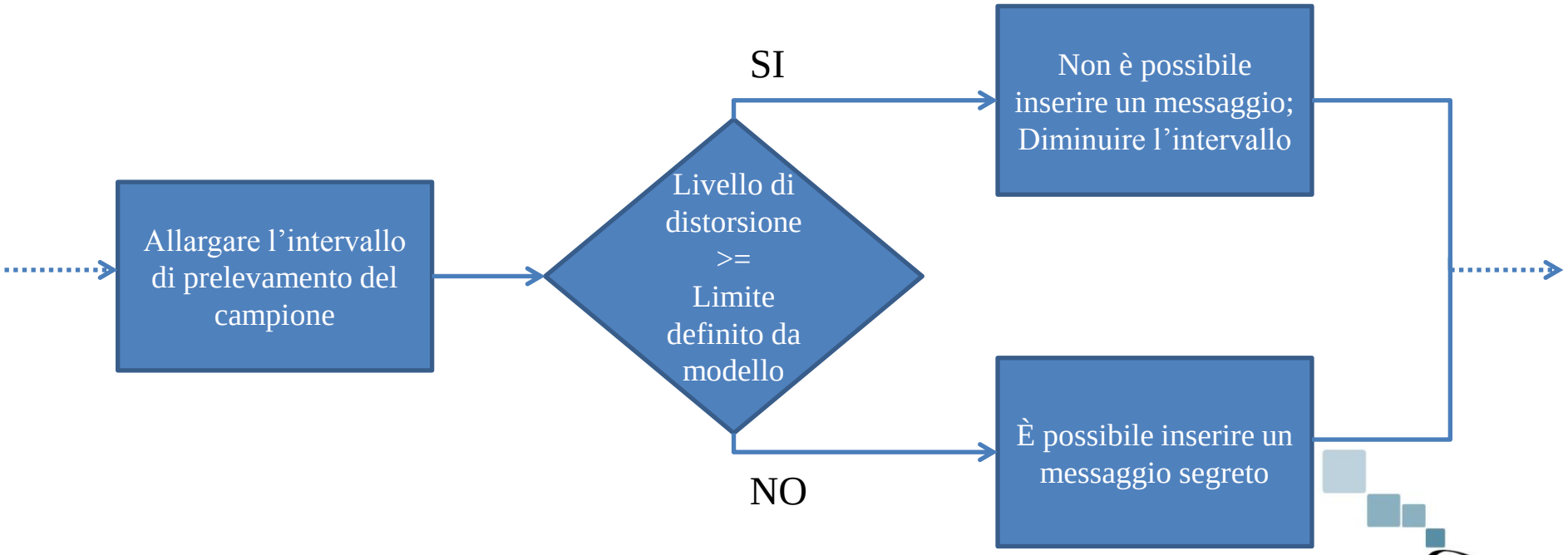

Esempio applicazione eco



# *Steganografia nell'audio digitale (.mp3)*

Anche nel caso dei file audio compressi Lossy, come il formato MP3, non è possibile iniettare il messaggio segreto operando come nel caso dei file WAV.

In questo caso si inserisce il messaggio segreto nella fase di *Inner loop:*



# Quali programmi usare?

- Hide’nSend
- Steghide UI
- Isteg
- Jsteg
- S-Tools 4
- Steganos 3 Security **Suite**
- Gif-it-up 1.0


http://www.jjtc.com/Steganography/tools.html

# Ancora Software

- **F5 :** software open source, che opera su immagini JPEG, BMP e GIF, utilizzando le tecniche di manipolazione dei coefficienti nel dominio trasformato attraverso la DCT. http://www.htw-dresden.de/westfeld/publikationen/f5r11.zip
- **Gif Shuffle:** software open source, che opera su immagini GIF, permutando i colori all'interno della palette. http://www.darkside.com.au/gifshuffle/
- **StegHide:** software open source, che opera su immagini BMP, implementando le tecniche di sostituzione del LSB. http://steghide.sourceforge.net/
- **OutGuess:** software open source, che opera su file JPEG e PNG utilizzando le tecniche di manipolazione dei coefficienti nel dominio trasformato attraverso la DCT. http://www.outguess.org/
- **Steganos Security Suite:** software commerciale, orientato alla protezione di documenti e password. http://www.steganos.com/
- **S-Tools:** software freeware, ma non opensource, che lavora su vari tipi di file (testo, audio, immagini). Il software combina tecniche steganografiche e criptografiche, codificando il messaggio con algoritmi di cifratura prima di inserirlo nel cover. http://www.spychecker.com/program/stools.html
- **MSU StegoVideo**: sviluppato in Russia nell'ambito del "compression project", consente di nascondere attraverso tecniche steganografiche qualsiasi file all'interno di un video. http://compression.ru/video/stego_video/index_en.html

# Secretbook


- È un progetto della Oxford University per gli utenti che utilizzano il browser Chrome e una sua estensione, Secretbook.

- Maggiori dettagli su:
- https://dl.dropboxusercontent.com/u/6853624/secretbook-draft-1.pdf

# Steganalisi

Esistono anche diversi tools steganalitici, che consentono di individuare, estrarre e/o distruggere messaggi nascosti all'interno di cover sospetti.

- **2Mosaic, StirMark Benchmark:** rimuovono messaggi steganografati da qualunque immagine.
  - http://www.petitcolas.net/fabien/watermarking/2mosaic/index.html
  - http://www.petitcolas.net/fabien/watermarking/stirmark/

- **StegDetect:** attacca con successo file steganografati attraverso i programmi *Jsteg*, *JPhide*, *Invisible Secrets*, *Outguess 01.3b*, *F5*, *appendX*, *Camouflage*, utilizzando gli attacchi statistici. http://www.outguess.org/detection.php

- **StegBreak:** individua ed estrae messaggi segreti inseriti dai software Jsteg-shell, JPhide, and Outguess 0.13b, utilizzando una variazione degli attacchi statistici, basata su un dizionario di mezzi cover tipici. http://manpages.ubuntu.com/manpages/hardy/man1/stegbreak.1.html

# Steganografia e Indagini

- Messaggi nascosti in file caricate su siti web pubblici (es. ebay, ecc.)
- La potenza della steganografia sta nel fatto che non c'è nessuna cifratura visibile ad un primo controllo, contrariamente a quanto avviene per le tecniche criptografiche; quindi la presenza di eventuali messaggi segreti che potrebbero essere utilizzati come evidenze va “scovata” caso per caso.
- La molteplicità di tecniche esistenti, e le grandi differenze che sussistono tra loro, rendono praticamente impossibile determinare la presenza di un messaggio nascosto, a meno di non avere indizi supplementari che l'analista può interpretare, supponendone la presenza o da ulteriori informazioni a corredo ottenute mediante metodi di investigazione tradizionale.

# Conclusioni

Esistono numerose tecniche steganografiche e una gran quantità di software (di validità dubbia e non documentata)

- sicurezza sembra banale ma non lo è
- conoscenza di principi base
- occhio agli attacchi di sistema

Steganalisi

- possibile e affidabile in certi casi
- molto difficile in altri
- dipendente dallo scenario
- …

# Bibliografia


- H. Delfs and H. Knebl, *Introduction to cryptography: principles and applications*. Springer-Verlag New York Inc, 2007.
- N. Amato, *La steganografia da Erodoto a Bin Laden: viaggio attraverso le tecniche elusive della comunicazione*. Italian University Press, 2009.
- G. J. Simmons, "The prisoners problem and the subliminal channel," in *Ádvances in Cryptology. Proc. of Crypto*, vol. 83, 1984, pp. 51–67.
- P. Hayati, V. Potdar, and E. Chang, "A survey of steganographic and steganalytic tools for the digital forensic investigator," in *Workshop of Information Hiding and Digital Watermarking to be held in conjunction with IFIPTM, Moncton, New Brunswick, Canada*, 2007.
- K. Choudhary, "Image steganography and global terrorism," *IOSR Journal of Computer Engineering, Volum*, vol. 1, pp. 34–48.

# Contatti

Per ulteriori dettagli o info si visiti il sito

**Image Processing Lab**

Università di Catania

www.dmi.unict.it/~iplab

Oppure email

battiato@dmi.unict.it



IMAGE PROCESSING LABORATORY