

REALIZZARE UN PROGRAMMA CHE NASCONDE TESTI IN UN'IMMAGINE

# COME TI NASCONDO

## IL FILE

In passato, il fatto di nascondere un messaggio in un altro aveva un vago sentore di zolfo e magia, ma anche oggi non è chiaro a tutti come possa funzionare questo processo. Vediamolo insieme!

byte per ciascun pixel. Ogni byte rappresenta il livello del colore primario cui fa riferimento.

Quindi, un byte indicherà il livello della componente rossa, uno il verde e uno il blu i quali sono i colori luce fondamentali.

Dopo avere detto che ogni pixel ha 3 byte, che indicano ciascuno le componenti cromatiche, va anche detto che una variazione dei valori dei singoli byte corrisponde a una variazione dell'immagine. Se essa è minima, è impercettibile all'occhio umano.

Ecco perchè si modificherà solo l'ultimo bit di ciascun byte dell'immagine per inserirvi un bit del testo da nascondere.

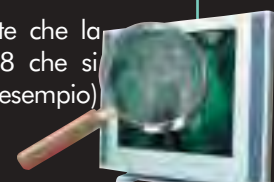
### Un esempio pratico

L'immagine ha vari byte che la compongono, i primi 8 che si analizzano hanno (per esempio) i seguenti valori:

[145] [211] [85] [99]  
[77] [177] [248] [218]

La prima lettera del testo da nascondere è la lettera "C" che ha un valore Ascii 67 che in binario corrisponde a 0100000111.

Scomponendo in bit i byte del-



1

Il termine Steganografia deriva dal greco e significa "scrittura nascosta".

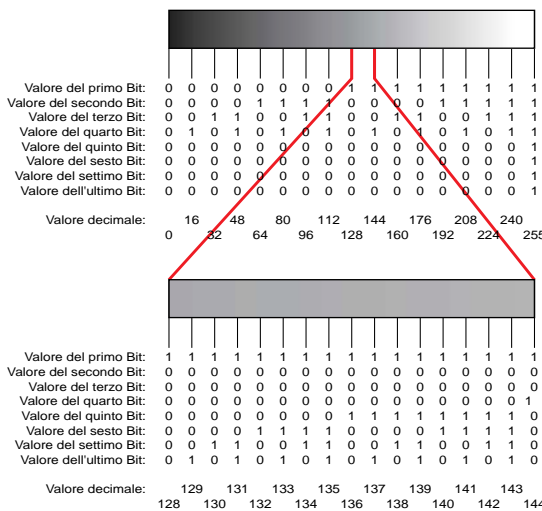
La steganografia ha varie forme; in informatica si utilizza in genere la steganografia sostitutiva, ovvero quella in cui si sostituiscono dei dati per nascondere altri.

Queste sono tecniche utili per **occultare testi o altro all'interno di documenti "insospettabili" quali, per esempio, immagini, audio o filmati**. Visti i sempre più frequenti discorsi sull'insicurezza in Internet, la steganografia consente di trasmettere documenti riservati o personali mantenendoli tali e non mettendoli alla mercé di chiunque.

Passando subito alla pratica (per la teoria della steganografia, consigliamo una ricerca sui vari motori di ricerca in Internet) si esamineranno in questo articolo i principi di base per realizzare un programma di steganografia.

esso un testo o un programma, altro non è che una sequenza di caratteri. Dato questo per scontato, si può partire dicendo che se si vuole celare un documento all'interno di un'altro si deve trovare un compromesso: **o si aggiunge qualcosa o si modifica qualcosa**.

Il sistema più efficiente e meno evidente consiste nel codificare un documento all'interno di una immagine. Anche questa è una sequenza di byte. Prendendo in esame una immagine a colori, generalmente a 24 bit per pixel, si hanno 3



Ogni byte (nelle colonne) indica il livello del colore primario cui fa riferimento. Sopra, i vari byte che indicano varie intensità di colore. Nella parte inferiore si nota che al variare dell'ultimo bit di una sola unità, corrispondono variazioni di tono impercettibili. Basandosi su questo fatto è possibile scomporre un carattere di testo nei bit corrispondenti secondo la codifica ASCII e quindi modificare l'ultimo bit dei byte relativi ai vari pixel dell'immagine per inserirvelo.

### >> Nascondere i dati

Qualsiasi documento informatico su qualsiasi piattaforma, appare al computer come sequenza di byte. Sia

REALIZZARE UN PROGRAMMA CHE NASCONDE TESTI IN UN'IMMAGINE

## Il programma di codifica

```
Definisci la variabile "Testo"
Definisci la variabile "Immagine"
Definisci la variabile "NuovaImmagine"
Definisci la variabile "Header" ##Conterrà la parte
di Header dell'immagine originale##
Leggi il contenuto del file di testo mettilo nella
variabile "Testo";
Leggi il contenuto del file di immagine, tralascia la
parte di Header, metti la parte dati nella variabile
"Immagine"
In questa fase, occorre conoscere la struttura del formato imma-
gine usato. Per semplificarci la vita, si può lavorare solo su for-
mati "grezzi" come per esempio il formato "RAW" di Photoshop,
che non ha Header.
```

```
Se la lunghezza del file di testo è maggiore di (lun-
ghezza del file di immagine / 8) avvisa che il file
di immagine è troppo piccolo quindi esci dal program-
ma.
```

```
Metti "1" nella variabile "ContatoreTesto"
Metti "1" nella variabile "ContatoreImmagine"
Ripeti per il numero di caratteri della variabile
"Testo"
    Metti il valore numerico del carattere N°
    ("ContatoreTesto") della variabile "Testo" nella
    variabile "BitCar"
```

```
    converti il valore di "BitCar" in binario
    metti "1" nella variabile "ContaBit"
Ripeti 8 volte
```

```
    se il valore del carattere N° ("ContaBit") della
    variabile "BitCar" è uguale a "0"
```

```
        metti il valore del carattere N°
        "ContatoreImmagine" nella variabile "ByteImmagine"
        metti (tronca ("ByteImmagine"/2)*2) nella varia-
        bile "ByteImmagine"
```

```
        converti la variabile "ByteImmagine" nel caratte-
        re corrispondente al suo valore numerico Sarà necessa-
        ria una funzione esterna
```

```
altrimenti
```

```
    metti il valore del carattere N°
    "ContatoreImmagine" nella variabile "ByteImmagine"
    metti (tronca ("ByteImmagine"/2)*2)+1 nella
    variabile "ByteImmagine"
```

```
Qui in realtà occorrerebbe verificare se il valore risultante è
superiore a 255, in tal caso avremmo un errore siccome non
posiamo avere un carattere con valore superiore a 255!
```

```
    converti la variabile "ByteImmagine" nel
    carattere corrispondente al suo valore numerico Sarà
    necessaria una funzione esterna
    fine "se"
```

```
    Aggiungi 1 alla variabile "ContatoreImmagine"
    ##in seguito uso il successivo carattere grazie
    all'impiego di questo contatore##
```

```
    Aggiungi 1 alla variabile "ContaBit"
    ##in seguito uso il successivo carattere grazie
    all'impiego di questo contatore##
```

```
    Fine ripeti ##Era: Ripeti 8 volte##
    Fine ripeti ##Era: Ripeti per il numero di caratteri
    della variabile "Testo"##
```

```
Giunti a questo punto, la variabile "Header" contiene ancora
la parte di Header dell'immagine originale; la variabile
"NuovalImmagine" contiene parte dei dati immagine già codi-
ficati per contenere il testo; la variabile "Immagine" contiene i
dati immagine originali. Ora si procede a scrivere il file della
nuova immagine col testo codificato al suo interno.
```

```
Apri il file (nome del file) in scrittura
Scrivi nel file (nome del file) la variabile "Header"
Scrivi nel file (nome del file) la variabile
"NuovalImmagine"
Scrivi nel file (nome del file) la variabile "immagi-
ne" a partire da (carattere (lunghezza della variabi-
le "NuovalImmagine") + 1)
Chiudi il file (nome del file)
```

### La composizione delle immagini

I colori della luce sono il rosso, il verde e il blu; sommandoli in egual misura si ottiene luce bianca. Questo metodo, detto sintesi additiva, è quello usato per generare colori in tutti i dispositivi che emettono luce come i monitor e le televisioni. Se si guarda con una lente d'ingrandimento una televisione, si vedono distintamente le singole parti rosse, verdi e blu che opportunamente comandate generano le immagini.

### >> Questioni generali

Se avete avuto la pazienza di leggere tutta la sezione di programmazione, **siete pronti per realizzare il vostro programma di Steganografia col linguaggio che preferite.**

Come suggerito nel codice, occorrono delle funzioni esterne per vari scopi, quali la verifica dell'esattezza del nome del file secondo gli standard della piattaforma adottata, la conversione da numero decimale a bi-

nario e viceversa.

L'unico problema che si può incontrare sta nella codifica dei vari formati di file i quali possono avere degli Header che vanno lasciati inalterati se si vuole potere visualizzare l'immagine.

Tra i formati di file da usare, sono sconsigliati i formati che usano compressione (per esempio TIFF compresso in LZW, JPG) in quanto **il file sarebbe da interpretare prima di poterlo usare, e poi producono file molto piccoli**, che permettono di nascondere meno dati. Con-

## Il programma di decodifica

```

Definisci la variabile "Testo"
Definisci la variabile "Immagine"
Leggi il contenuto del file di immagine, tralascia
la parte di Header, metti la parte dati nella
variabile "Immagine"
Metti "1" nella variabile "ContatoreImmagine"
##tiene conto di quale byte di immagine sto elabo-
rando sto esaminando##

```

```

Ripeti per il numero di caratteri della variabile
"Immagine"

```

```

Metti 1 nella variabile "ContaTesto" ## Quando
questa variabile è uguale a 8 significa che ho
letto un carattere del testo ##

```

```

Metti "" nella variabile "CarattereLetto" ##
Inizializzo o svuoto la variabile che contiene il
carattere letto ##

```

```

Ripeti finchè ("ContaTesto" = 8)
metti il valore numerico del carattere n°
"ContatoreImmagine" della variabile "Immagine"
nella variabile "NumCar"
Se (tronca ("NumCar" / 2)) * 2 è uguale a
"NumCar" ## significa che l'ultimo bit era "0" ##
metti "0" dopo la variabile "CarattereLetto"
altrimenti
metti "1" dopo la variabile "CarattereLetto"
fine se

```

```

Aggiungi "1" alla variabile "ContatoreImmagine"
Aggiungi "1" alla variabile "ContaTesto"
Fine ripeti ## Ripeti finchè (ContaTesto = 8) ##

```

```

Converti "CarattereLetto" da binario al carattere
corrispondente ## sarà necessaria una funzione
esterna ##

```

```

Metti "CarattereLetto" dopo la variabile "Testo"
## la variabile "CarattereLetto" contiene un carat-
tere ascii, lo vado a scrivere dopo il contenuto
della variabile "Testo" che già contiene il testo
finora letto ##

```

```

Fine ripeti ##Era: Ripeti per il numero di caratte-
ri della variabile "Immagine"##

```

```

Apri il file (nome del file di testo) in scrittura
## Il nome del file potrebbe essere un nome stan-
dard, il nome dell'originale più un testo per iden-
tificarlo o un nome definito dall'utente. ##

```

```

Scrivi nel file (nome del file di testo) la
variabile "Testo"
Chiudi il file (nome del file)

```

viene usare formati come il TIFF non compresso e il BMP i quali (dopo la sezione di header) scrivono i dati byte per byte. Per tutti coloro che non hanno familiarità con i formati grafici, il formato consigliato è il formato RAW (disponibile da diversi programmi quali per esempio Photoshop anche nella versione economica Photoshop Elements) dato che questo **scrive nel file solo i dati immagine**. Usando questo formato, occorre ricordarsi però le dimensioni in pixel dell'immagine, dato che esse non sono scritte nel file!

### >> Conclusioni

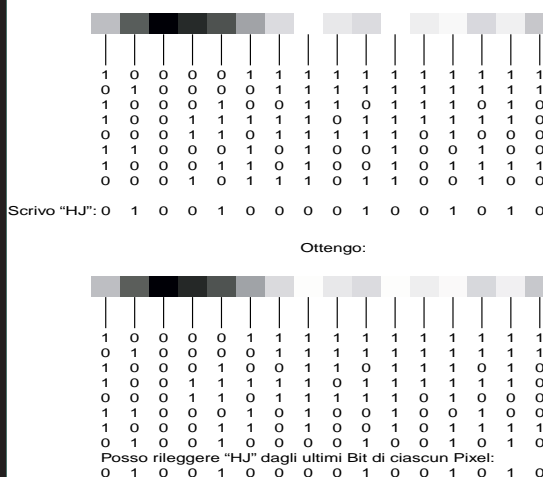
Un programma di steganografia è un ottimo strumento per tenere nascosto ciò che abbiamo di più privato, dalle lettere della morosa fino ai documenti aziendali più riservati. Va comunque detto che i dati scritti,

una volta estratti, sono leggibili da chiunque anche con altri programmi di steganografia. Se si teme di avere vicino qualche curioso, può essere conveniente cifrare i testi usando un programma tipo PGP prima di inserirli nell'immagine. In questo modo, anche estraendo i dati, si dovrà lottare per tentare di

leggerli e si avrà una ancor maggiore – per non dire totale – riservatezza.

Volendo, si può inserire nell'immagine e quindi nascondere, non solo un testo ma anche un qualsiasi file non TXT.

**Enzo Borri**



Sono stati presi in esame 16 Pixel di una immagine e viene mostrato il valore dei primi 16 Pixel. Per semplicità, è stata presa una immagine in scala di grigi. Nella parte superiore i dati contenuti nell'immagine originale cui, mediante modifica, sono inserite le lettere "HJ" che hanno rispettivamente i valori "0 1 0 0 1 0 0 0" e "0 1 0 0 1 0 1 0". L'ultimo Bite di ciascun pixel viene così modificato se necessario. Nonostante le modifiche, l'aspetto estetico dell'immagine resta invariato.