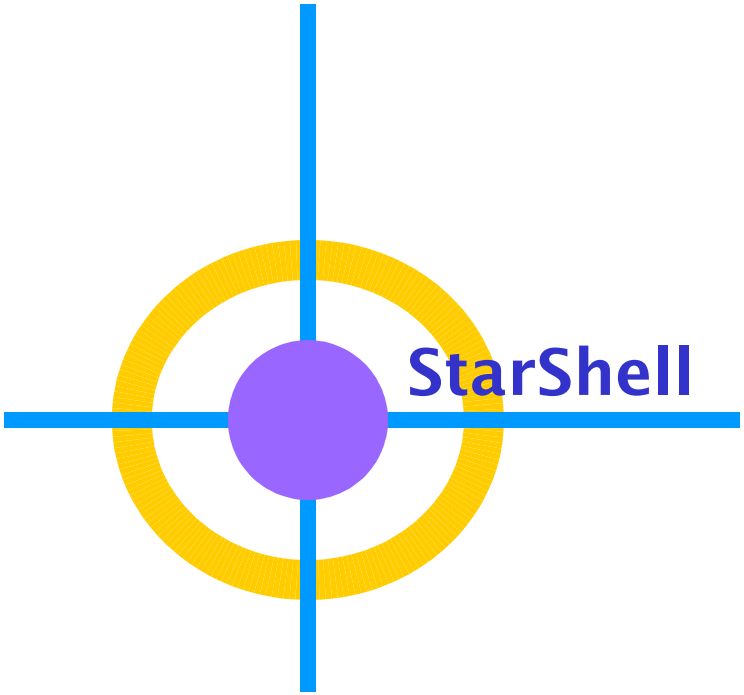


Steganografia

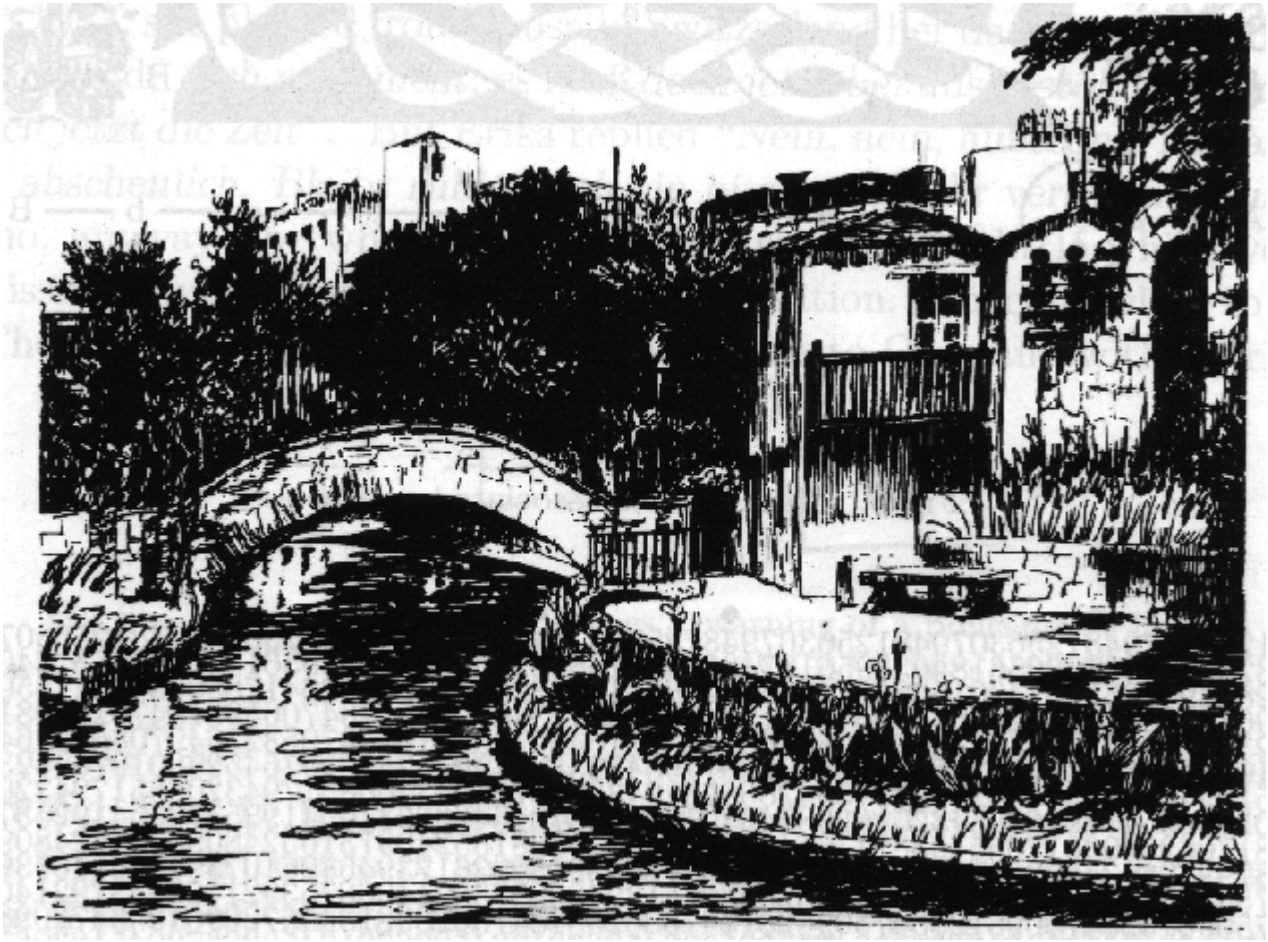
Etimologia

- Greco steganos: coperto
  - **Scrittura segreta coperta**
  - **I lettori terzi non si accorgono nemmeno della presenza di un messaggio nascosto**
- Greco kryptos: nascosto
  - **Scrittura segreta aperta**
  - **I lettori terzi sanno che esiste un messaggio ma non riescono a comprenderlo**
- Classi
  - **Steganografia Linguistica**
  - **Steganografia Tecnica**

Steganografia Tradizionale

- Steganografia Lingustica
  - Codice aperto: il messaggio sembra innocente
    - **Mascheratura: circostanze note solo ai recipienti – gergo, mots convenus**
    - **Velatura: solo alcuni caratteri nel testo – acrostici, cronogrammi, griglie**
  - Semagrammi: dettagli grafici visibili in uno scritto o disegno
    - **Lettere abbassate o spaziatura delle lettere**
    - **Gli elementi di un disegno contengono il messaggio**
- Sreganografia Tecnica
  - Inchiostri invisibili, micropunti
  - Trasmissioni a 'spurt', congegni meccanici

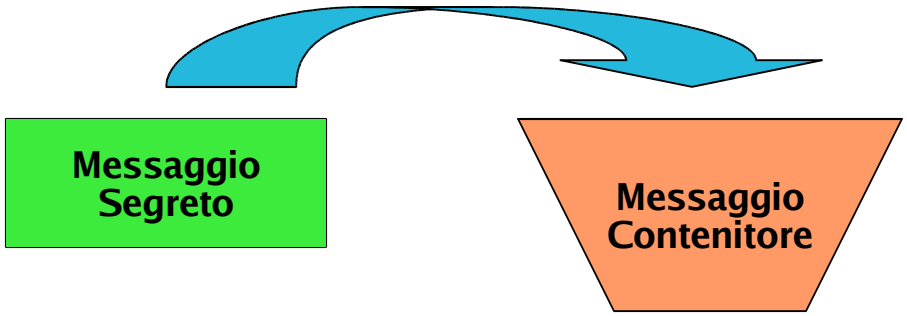
Esempio di Steganografia



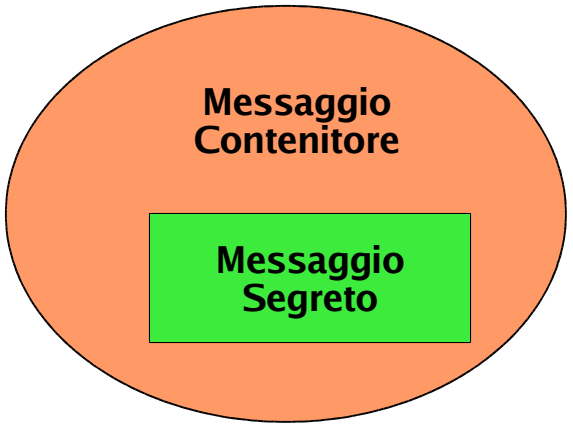
Modelli Steganografici

- Due messaggi:
  - **messaggio segreto**
  - **messaggio contenitore**

Steganografia Iniettiva



Steganografia generativa



Altri Modelli:
- **Steganografia sostitutiva**
- **Steganografia selettiva**
- **Steganografia costruttiva**

Steganografia sostitutiva

Rumore
Sempre presente - conversione analogica-digitale
Influenza i bit meno significativi

Inserire il messaggio segreto nei bit meno significativi dopo la conversione
Simula il rumore

File Bitmap (.BMP)

Matrice di pixel in formato RGB, dimensione MxN
P. es. a 24 bit per pixel, con tre canali, servono 3xMxN bytes

Pixel originario trasmesso:

10011001 00110010 11000111

Sostituire i tre bit meno significativi dei canali con tre bit del messaggio:

10011001 00110011 11000110

La percezione visiva
- non vede molta differenza col colore originario
- non sa quale fosse esattamente il colore originario

Iniezione di Messaggio

- Dimensione del messaggio iniettato:
  - **(M x N x 3) / 8**
- Esempio: immagine di risoluzione 640 x 480 pixel
  - **(640 x 480 x 3) / 8 = 115200 bytes**

- E' possibile codificare il messaggio con più bit
  - **2, 3 o 4 bit meno significativi per canale**
  - **aumento dimensioni del messaggio steganografico**
  - **aumento del 'rumore' percepito**
    - **diminuzione della qualità dell'immagine**
    - **probabilità di insospettire il ricevente**

File sonoro (.WAV)

Formato 44100 Hz, 16 bit, stereo, durata 1 minuto

Generata una stringa di bit ogni 1/44100 secondi

La stringa è di 16 bit

Sono generate due stringhe di bit

Dimensione file contenitore:
44100 x 16 x 2 x 60 = 84762000 bit
= 10366 kB

Usando i due bit meno significativi,
la dimensione massima del messaggio nascosto è:
84762000 x 2 / 16 = 10595250 = 1293 kB

File JPEG

Iniettare le informazioni
in un file BMP e poi convertirlo
in JPEG?

**JPEG è un formato 'lossy':
il messaggio nascosto
viene perduto**

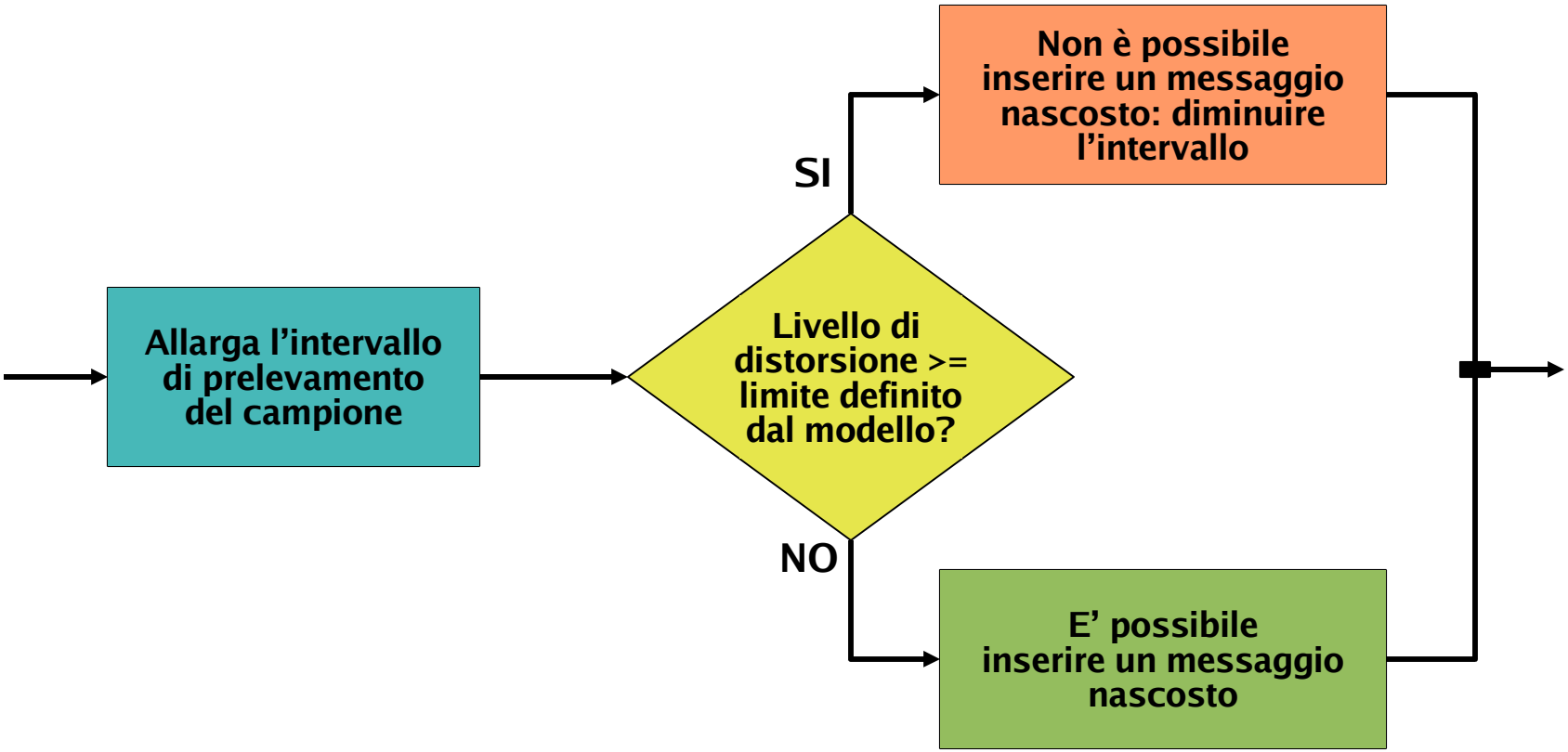
Non è possibile usare una semplice tecnica sostitutiva

**La compressione JPG intende preservare le caratteristiche visive dell'immagine e non la sequenza di pixel originaria.
Nella compressione JPG viene compiuta una trasformata di Fourier dell'immagine originaria e vengono tagliate le frequenze più basse prima di compiere una trasformata di Fourier inversa.**

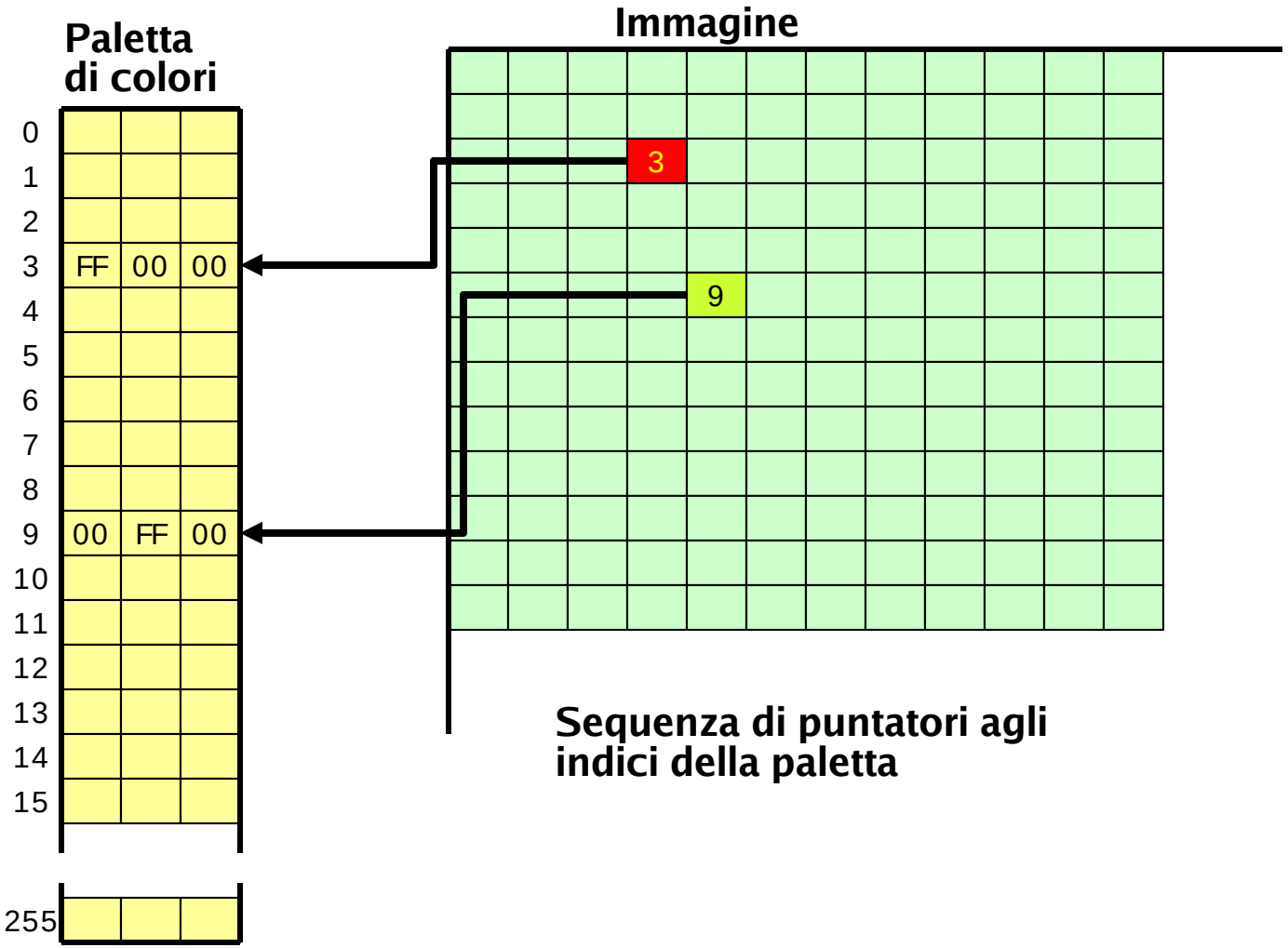
Soluzione: iniettare le informazioni nei coefficienti di Fourier ottenuti nella prima fase della compressione

File MP3

Si inserisce il messaggio segreto nella fase di ‘inner loop’:



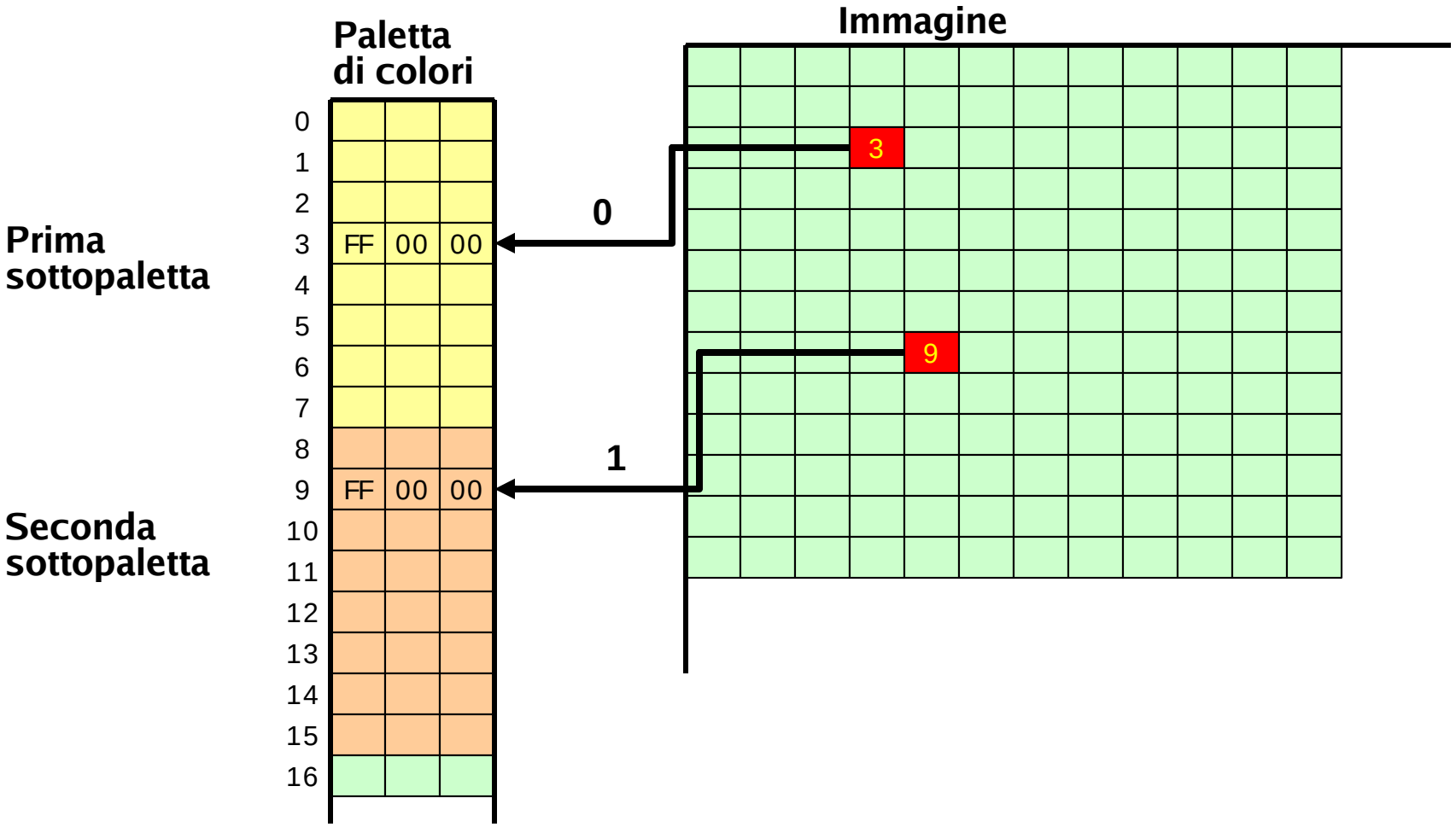
File GIF


Paletta
di colori
Immagine
0
1
2
3
FF
00
00
4
5
6
7
8
9
00
FF
00
10
11
12
13
14
15
255
3
9
Sequenza di puntatori agli
indici della paletta

File GIF

- Decrementare il numero di colori della paletta
  - **potenza del 2, inferiore a 256**
    - **risultato: 1/2 o 1/4 della paletta originale**
  - **limitare la perdità di qualità**
  - **usare algoritmi specifici**
- Duplicare (o quadruplicare) la paletta risultante
  - **la nuova paletta è di 256 colori**
  - **ogni colore ha 2 o 4 entries nella paletta**
- La scelta della semipaletta determina il bit del messaggio nascosto
  - **P. es. 0 per paletta originaria, 1 per paletta copia**
- Immagine M x N pixel
  - **messaggio nascosto di almeno M x N / 8 bytes**

File GIF



StarShell

Osservazioni

- E’ molto facile scoprire la presenza di sottopalette
- Altro metodo di staganografia GIF:
  - **le 256 entries della paletta si possono disporre in 256! permutazioni**
  - **variando l’ordine delle permutazioni si può codificare un messaggio**
  - **dimensione massima:**
    - **$\log_2 256! = 1683$ bit = 210 bytes**
- Anche altri formati sono adatti come contenitori
  - **Postscript, PDF, HTML, ecc.**

Difetti

- Esposizione del contenitore
  - **Non utilizzare file contenitori il cui originale è accessibile**
  - **Non riutilizzare lo stesso contenitore per più messaggi**
  - **Distruggere il contenitore dopo l'uso**
- Modelli di rumore
  - **Il messaggio cambia le caratteristiche statistiche del rumore, sostituendosi ad esso**
  - **Chi possieda un modello statistico del rumore può accorgersene**
  - **Lo scopo è che l'opponente non sospetti neppure la presenza del messaggio nascost**
    - **Steganalisi**

Steganografia Selettiva

- Selezionare solo file contenitori che già possiedano una certa proprietà, tra tutti i possibili
  - **P. es. file di dimensione pari codificano uno 0, dispari un 1**
  - **Meglio se molti contenitori piccoli**
- Il file contiene il messaggio segreto senza essere stato modificato
- La banda passante è molto bassa
- Poco usata in pratica
  - **Ma: utile per invio password, chiavi, 'chirp'**

Steganografia Costruttiva

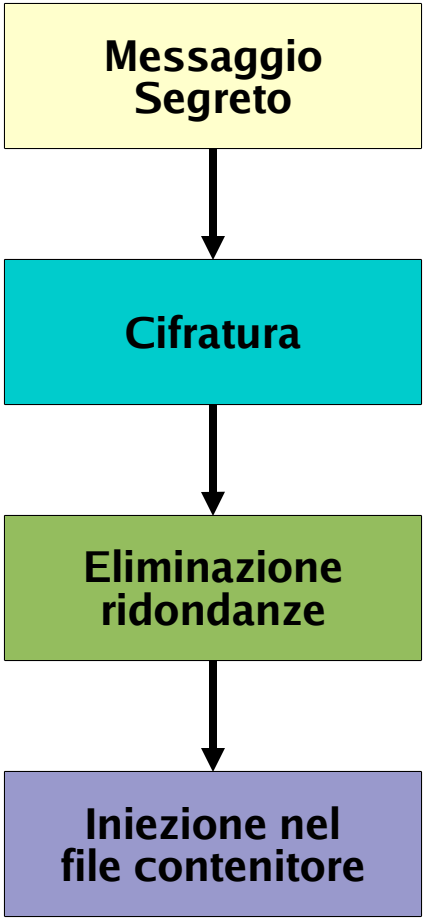
- Tiene conto di un Modello di Rumore
- Il falso rumore introdotto è statisticamente compatibile col modello
- Problemi
  - **difficile realizzazione di un modello di rumore**
  - **può sempre essere soprasseduto da un modello più avanzato in mano nemiche**
  - **se il modello cade in mano nemica, si possono forse trovare debolezze del modello e sfruttarle**

Principio di Kerchoff

- Il nemico conosce appieno il progetto e l'implementazione del modello di rumore
- Il nemico non dispone solo di una chiave segreta
  - **sequenza limitata di bit**
- Senza la chiave segreta non è deducibile che un canale di comunicazione contenga o meno un messaggio nascosto

- Soluzioni:
  - Cifrare il messaggio segreto prima di iniettarlo nel contenitore
  - Eliminare ogni ridondanza prima dell'iniezione
    - **basso profilo contro metodi di crittanalisi**

Sistema Steganografico Ideale



Software

- Steganos 3 Security Suite
- Steganos File Manager
- S-tools
- Jsteg Shell
- Mp3 Stego
- Gifitup

File System Steganografico

- I file sono organizzati in insiemi: livelli di sicurezza
- Ad ogni livello corrisponde una passphrase
- Gerarchia lineare
  - **l'accesso al livello n permette l'accesso anche a tutti i livelli inferiori**
- Senza la passphrase non è nota neppure l'esistenza del livello di sicurezza
- I livelli di sicurezza massimi sono prefissati
- I livelli utilizzati possono essere inferiori al massimo
  - **L'attaccante non sa quanti livelli sono utilizzati**

Diniego Plausibile

- Dato un sistema di sicurezza con parametri $\{p_1,p_2,p_3,...,p_n\}$ è possibile negare in modo convincente ogni conoscenza di un parametro p_j

- P. es. mentire sul numero di livelli di sicurezza effettivamente utilizzati

Primo Metodo

- Anderson, Needham, Shamir - 1998
- Architettura:
  - **File mascherati con contenuto casuale iniziale**
  - **Memorizzazione come modifica di più file mascherati**
  - **Estrazione: combinazione XOR dei file mascherati**
  - **La passphrase di accesso identifica l'insieme dei file mascherati**
- Usa solo algebra lineare, non crittografia
- Difetti
  - **scarse prestazioni - elevato numero di file mascherati**
  - **manca supporto a file lunghi e directories**
  - **l'attaccante deve ignorare qualsiasi parte del testo in chiaro**

Secondo Metodo

- Anderson, Needham, Shamir - 1998
  - **File system inizialmente riempito da blocchi con contenuto casuale**
  - **I blocchi dei file cifrati sono nascosti tra i blocchi casuali in locazioni pseudo-casuali**
    - **derivata p. es. da hash del percorso e nome file**
  - **I blocchi cifrati sono indistinguibili da quelli casuali**
- Difetti
  - **L'algoritmo di hash può generare collisioni e sovrascritture**
    - **ogni blocco cifrato viene duplicato n volte**
  - **Allocazione molto inefficiente**
  - **Overhead di cifratura e duplicazione in lettura e scrittura**

Terzo Metodo

- Van Schaik, Smeddle
  - **Nascondere l'informazione con combinazioni lineari di sottoinsiemi di blocchi**
  - **Marcatura dei blocchi con livelli di sicurezza accettabili**
- Difetti
  - **Mancanza di effettivo diniego plausibile**
  - **Possibilità di ricavare un limite superiore sulla quantità di blocchi nascosti**

Linux StegFS

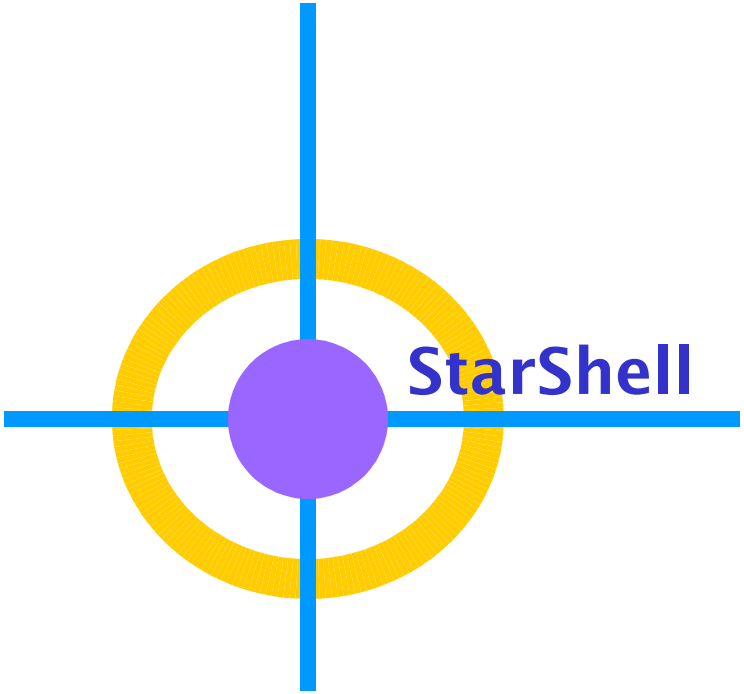
- Andrew D. McDonald e Markus G. Kuhn, 1999
  - **Licenza GNU**
  - **Implementazione del secondo metodo steganografico**
- Proprietà:
  - **15 livelli di sicurezza**
  - **Contesto di Sicurezza: insieme di livelli accedibili tramite singola chiave**
  - **Contesto di sicurezza N dà accesso a tutti i livelli di sicurezza da 1 a N (accesso lineare - default)**
  - **Possibile costruire gerarchie complesse e non lineari**

Gestione delle Chiavi

- Matrice di Sicurezza, 15 x 15
  - **righe: livelli di sicurezza**
  - **colonne: contesti di sicurezza**
  - **elemento: chiave di livello cifrata con lo hash della chiave di contesto**
- Viene decifrata l'intera colonna delle chiavi appartenenti al contesto
- Le chiavi di livello permettono l'accesso ai blocchi dei file appartenenti a tale livello

Allocazione dei Blocchi

- I-node posti in blocchi a posizione casuale
  - **hash della combinazione di i-number e chiave di livello**
- Blocchi dati in posizioni completamente casuali
  - **/dev/urandom genera i numeri casuali**
- E' mantenuta una Tabella di Allocazione dei blocchi
  - **una entry per blocco disco**
  - **segna i blocchi in uso e loro proprietà**


StarShell