# Compressione Dati

## Data Audio Hiding

Andrea Di Pierno
Marco Russo
Hermann Senatore

01

## BASI TEORICHE

Introduzione alla Teoria dei Segnali

02

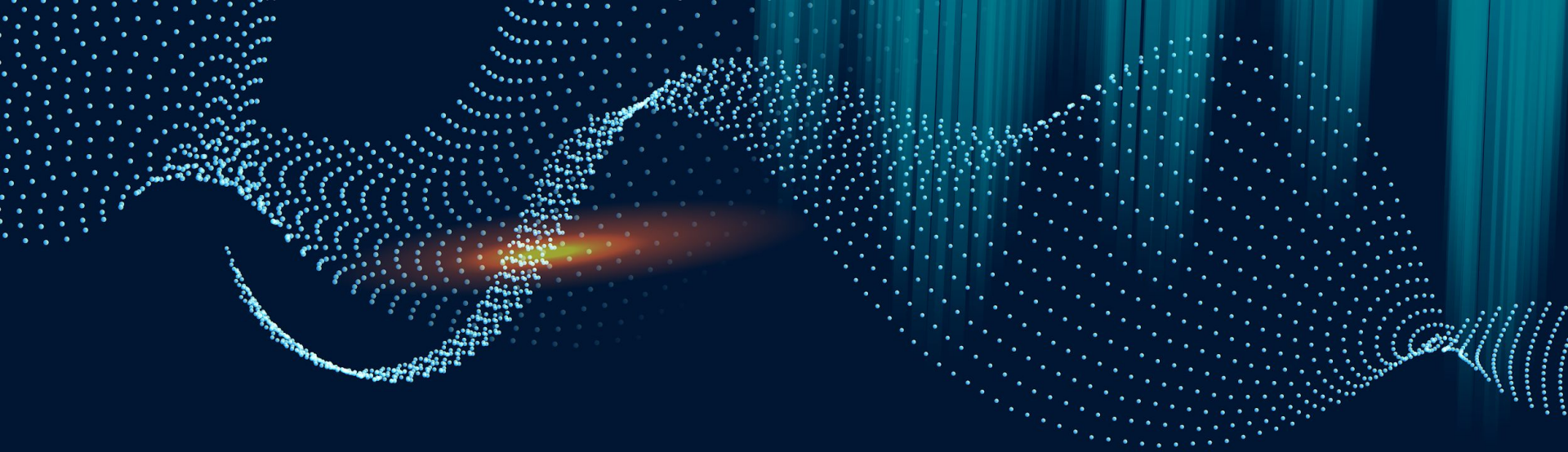
## MP3

Breve analisi del codec e dei file MP3

03

## DATA AUDIO HIDING

Stato dell’arte delle principali tecniche di steganografia audio

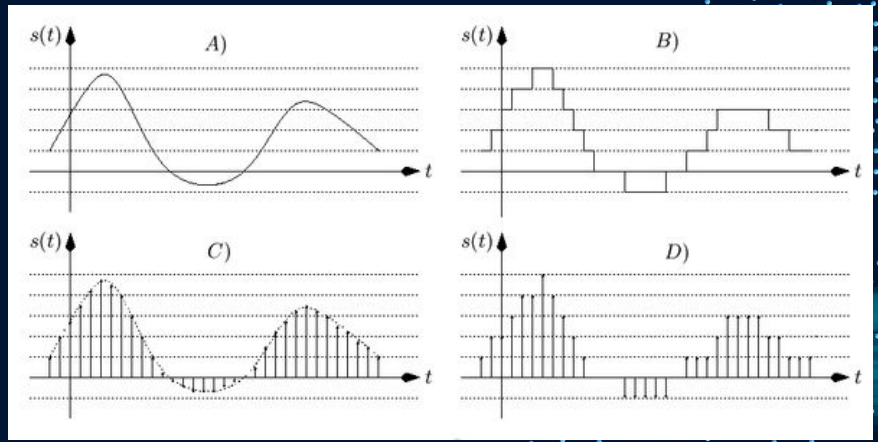


# 01 Basi Teoriche

Teoria dei Segnali

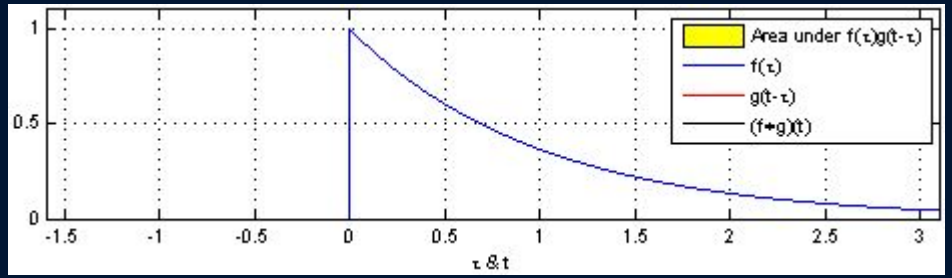
# COS'È UN SEGNALE?

Un segnale è una variazione temporale dello stato fisico di un sistema (o di una grandezza fisica), come la tensione o l'intensità di corrente per i segnali o i parametri di campo elettromagnetico per i segnali radio, che serve per rappresentare e/o trasmettere messaggi ed informazioni



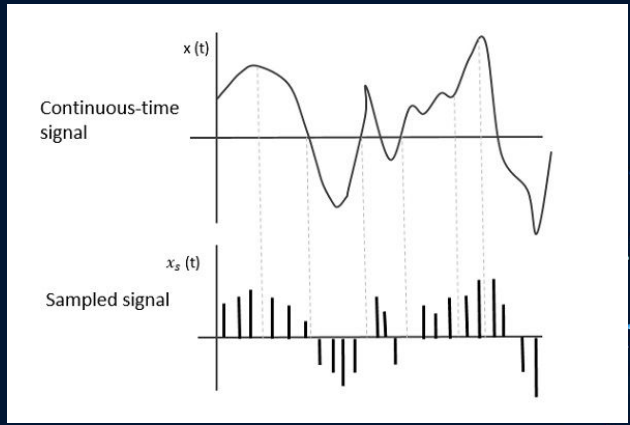
# CONVOLUZIONE

È un’operazione tra due funzioni (in questo caso segnali) che consiste nell’integrare il prodotto tra il primo ed il secondo segnale traslati di un certo valore



# CAMPIONAMENTO

Il campionamento è una tecnica che permette di convertire un segnale continuo nel tempo in un segnale discreto, valutandone l'ampiezza ad intervalli temporali o spaziali regolari.



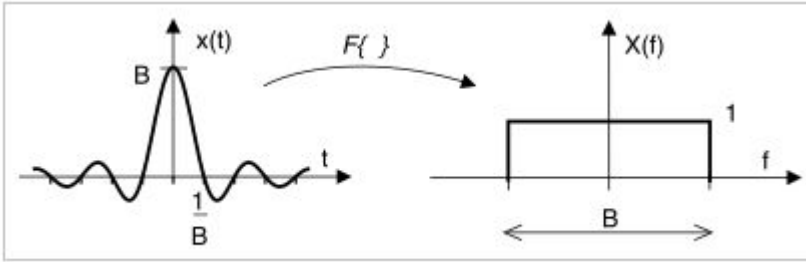
# TEOREMA DEL CAMPIONAMENTO

Il teorema che stabilisce quale sia la frequenza di campionamento con una determinata caratterizzazione in frequenza affinché il segnale analogico possa essere ricostruito a valle a partire da quello discreto in input è il teorema di Shannon-Nyquist (teorema del campionamento), ovvero, la frequenza di campionamento deve essere maggiore a 2 volte la frequenza dello spettro del segnale da campionare.

# TRASFORMATA DI FOURIER

La trasformata di fourier è un operatore che permette di rappresentare nel dominio delle frequenze di un segnale nel dominio del tempo e viceversa.

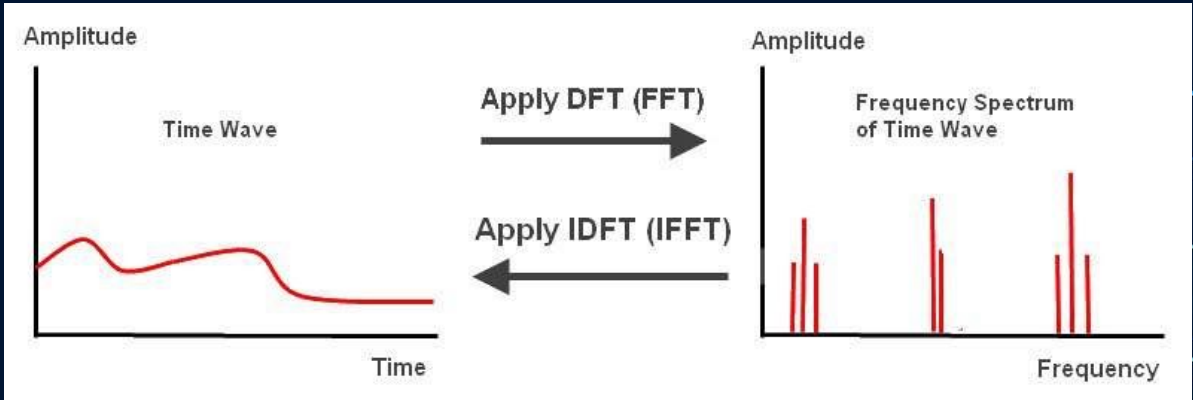
Viene utilizzata per poter calcolare in maniera efficiente la convoluzione di un segnale

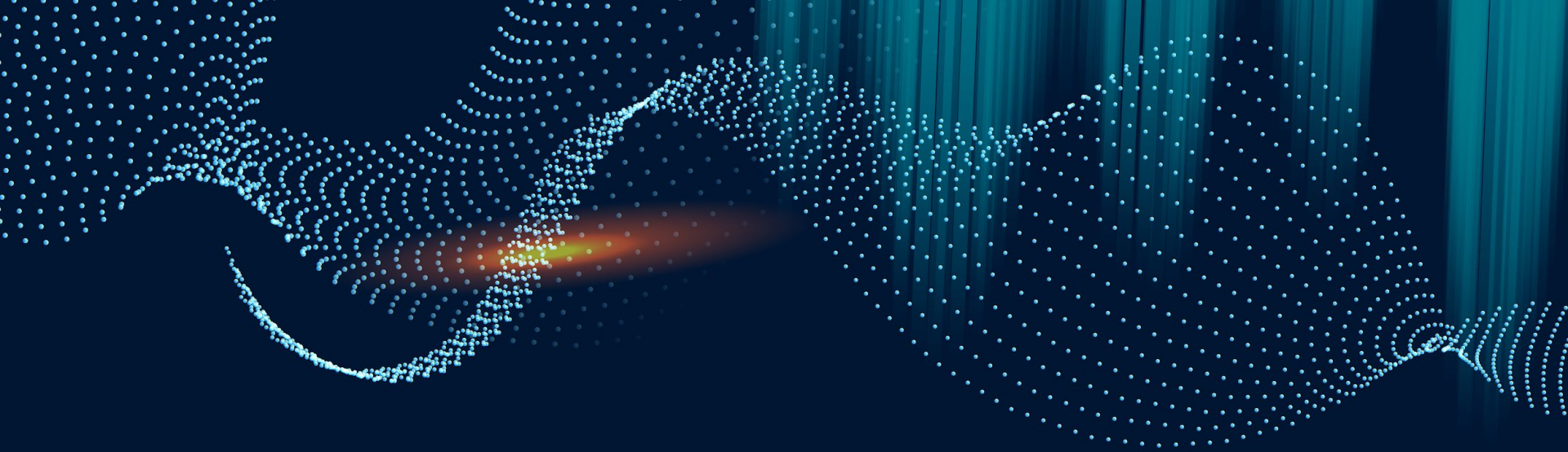


# DISCRETE FAST FOURIER TRANSOFRM

È un algoritmo utilizzato per calcolare in maniera efficiente la trasformata discreta di fourier e la sua inversa.

È utilizzata per l'elaborazione di segnali digitali poichè ha un basso costo computazionale





# 02 MP3

Audio Layer per MPEG

# INTRODUZIONE AD MP3

## ORIGINI

L’acronimo MP3 nasce nel 1997 dalle email di un gruppo di esperti MPEG.

Formalmente conosciuto come MPEG-1 Audio Layer III, rivoluzionò il modo di poter ordinare le tracce audio grazie all’introduzione delle playlist.

## MP3 OGGI

Benché MP3 sia ancora largamente impiegato nella codifica audio e supportato dalla maggior parte dei dispositivi in commercio, lo standard MPEG ha adottato AAC come suo successore.

Degno di nota è anche il codec Ogg Vorbis, famoso per essere open source ed impiegato in applicazioni largamente utilizzate come WhatsApp.

# TECNICHE DI CODIFICA

## STANDARD STEREO

Codifica indipendente dei canali L ed R. Migliore resa qualitativa.

## FORCE STEREO

Codifica di un unico canale, sdoppiato durante la riproduzione. Elevata perdita qualitativa.

## JOINT-STEREO (Mid/Side)

Codifica di un unico canale, con aggiunta di informazioni sulle differenze tra L ed R.

## JOINT-STEREO (Intensity)

Codifica basata sul principio di localizzazione sonora che impiega tecniche di modulazione dell'ampiezza inter-aurale

# STRUTTURA DI UN FILE MP3

Un file MP3 è suddiviso in frame da 1152 samples ciascuno.
Ogni frame ha una durata di 26ms, quindi avremo 38 fps.

## FRAME MP3

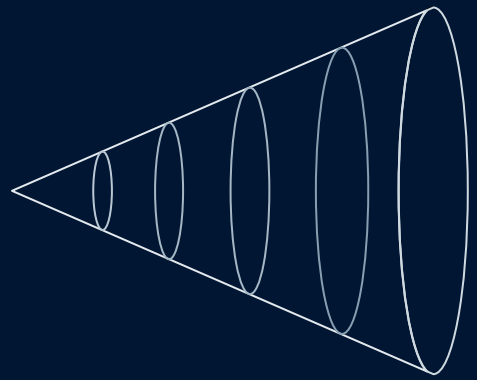
| Header | CRC | Side informations | Main data | Accessory Data |
|---|---|---|---|---|

I frame sono a loro volta suddivisi in granules da 576 samples ciascuno.

A seconda del bitrate e della frequenza di campionamento, avremo samples più o meno grandi.

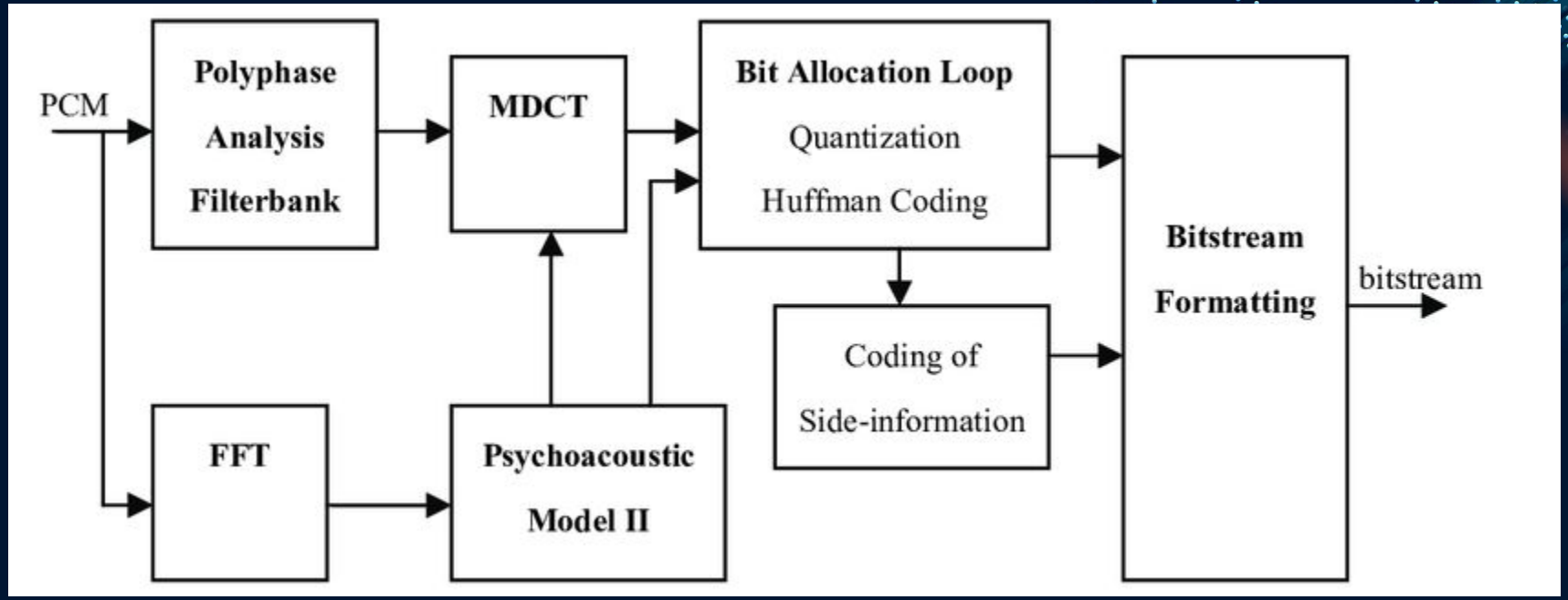
# HEADER MP3

Header

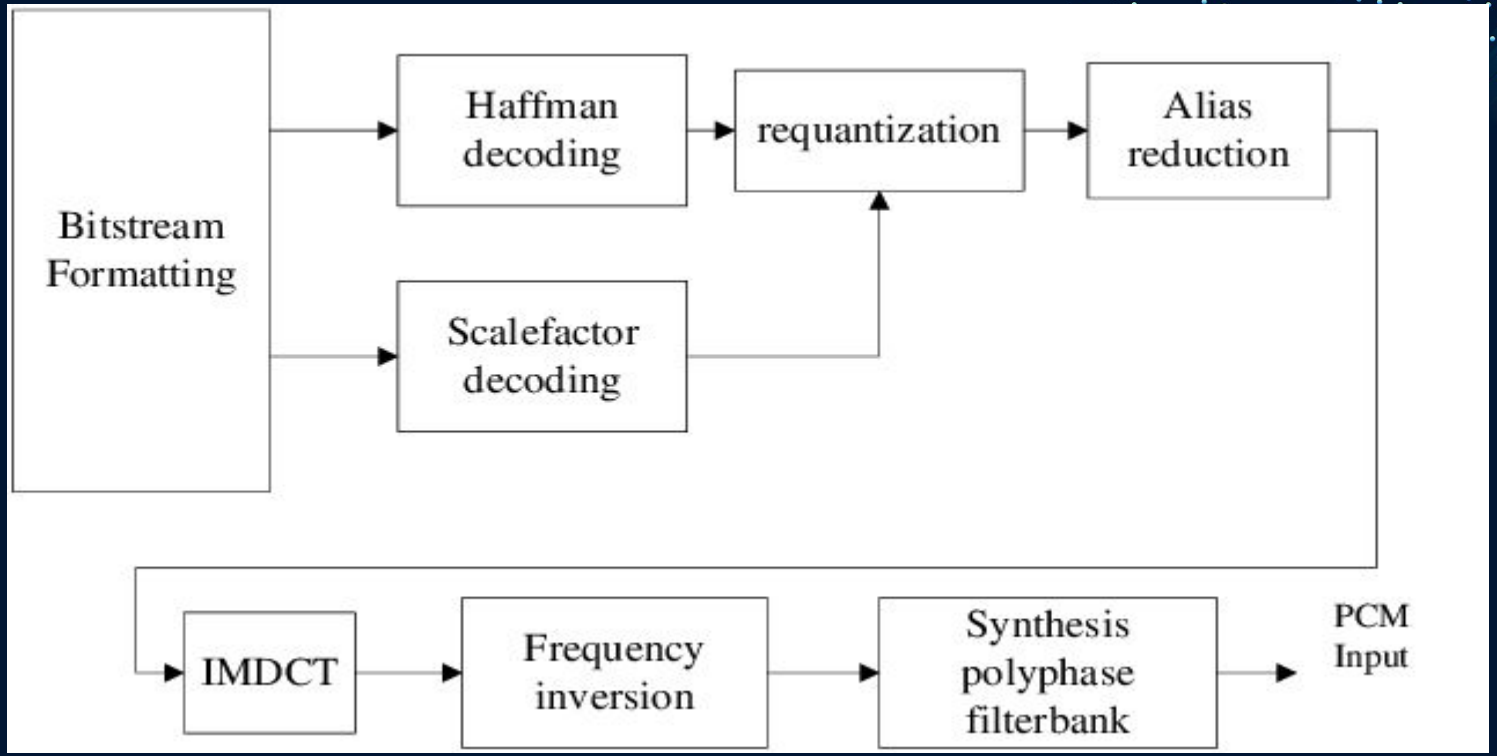


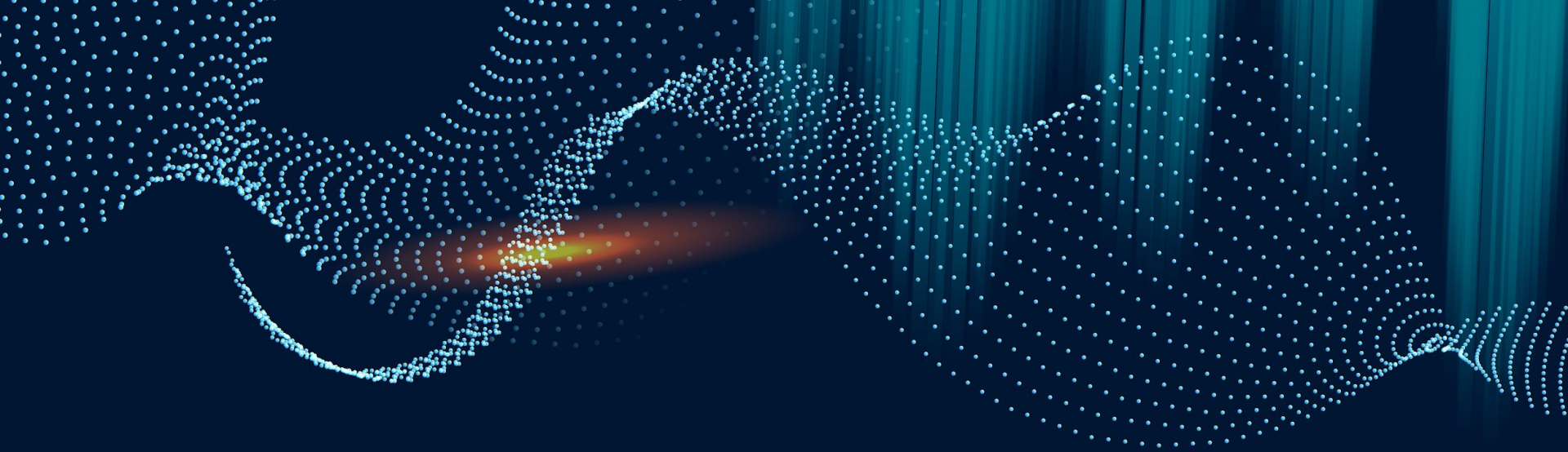
| | | |
|---|---|---|
| Sync (12 bit) | | |
| ID | Layer (2 bit) | Protection |
| Bitrate (4 bit) | | |
| Frequency (2 bit) | Padding | Private |
| Mode (2 bit) | Mode extension (2 bit) | |
| © | Home | Emphasis (2 bit) |

# CODIFICA MP3



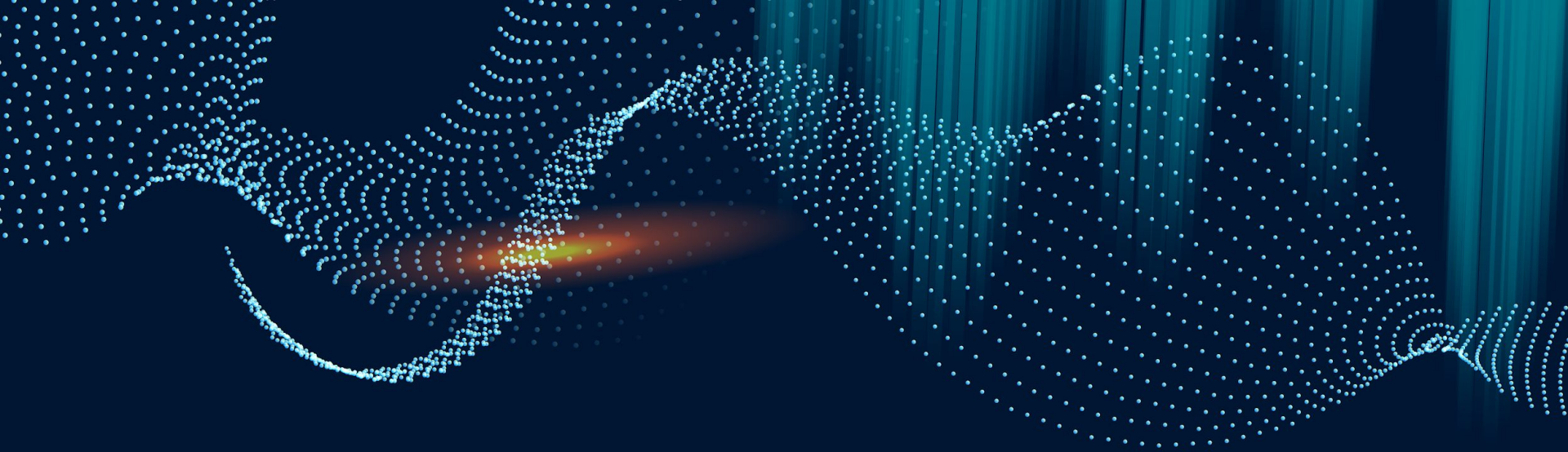
# DECODIFICA MP3





# 03 Data Audio Hiding

Stato dell’arte



# STEGANOGRAFIA AUDIO

*“Nascondere informazioni*
*all’interno di una traccia audio”*

# TOOL PER STEGANOGRAFIA AUDIO

1. DeepStego
2. Mp3Stego
3. StegHide
4. QuickStego
5. Audio Stego

# TECNICHE DI STEGANOGRAFIA AUDIO - STATO DELL'ARTE

## Echo Hiding

H. B. Dieu

ICIEIS 2013

## Amplitude Hiding

M. Wen-Nung Lie,
L. - C. Chang

IEEE Transaction on Multimedia 2006

# ECHO HIDING - DIEU, 2013

Strategia semplice ed immediata;

Embedding delle informazioni tramite inserimento di eco nella traccia;

L’algoritmo fa uso di una chiave condivisa.

# ECHO HIDING - EMBEDDING

- Viene utilizzata una chiave k = *(int seed, int a);*
- Mediante k viene generata una sequenza binaria R della stessa lunghezza del messaggio da nascondere;
- Si divide la traccia in frame di 1024 samples ciascuno;
- Se ci sono meno frame che bit da nascondere, l'algoritmo termina;
- Ad ogni frame i corrisponde un bit da nascondere:
  - Se $R_i$ è 1, allora per codificare il bit 1 non viene effettuata alcuna operazione, per codificare il bit 0 si aggiunge dell'eco all'audio;
  - Se $R_i$ è 0, allora per codificare il bit 0 non viene effettuata alcuna operazione, per codificare il bit 1 si aggiunge dell'eco all'audio
- Il resto dei frame viene lasciato inalterato

# ECHO HIDING - RETRIEVAL

- Si genera la sequenza R a partire da k;
- Si divide la traccia audio modificata in frame da 1024 samples;
- Viene effettuato un confronto tra la traccia audio originale e la traccia audio modificata;
- Per ogni frame i:
  - Se i differisce nella traccia modificata, basandosi su $R_i$ viene estratto 0 o 1;
  - Ripeti;
- Viene restituito il messaggio binario.

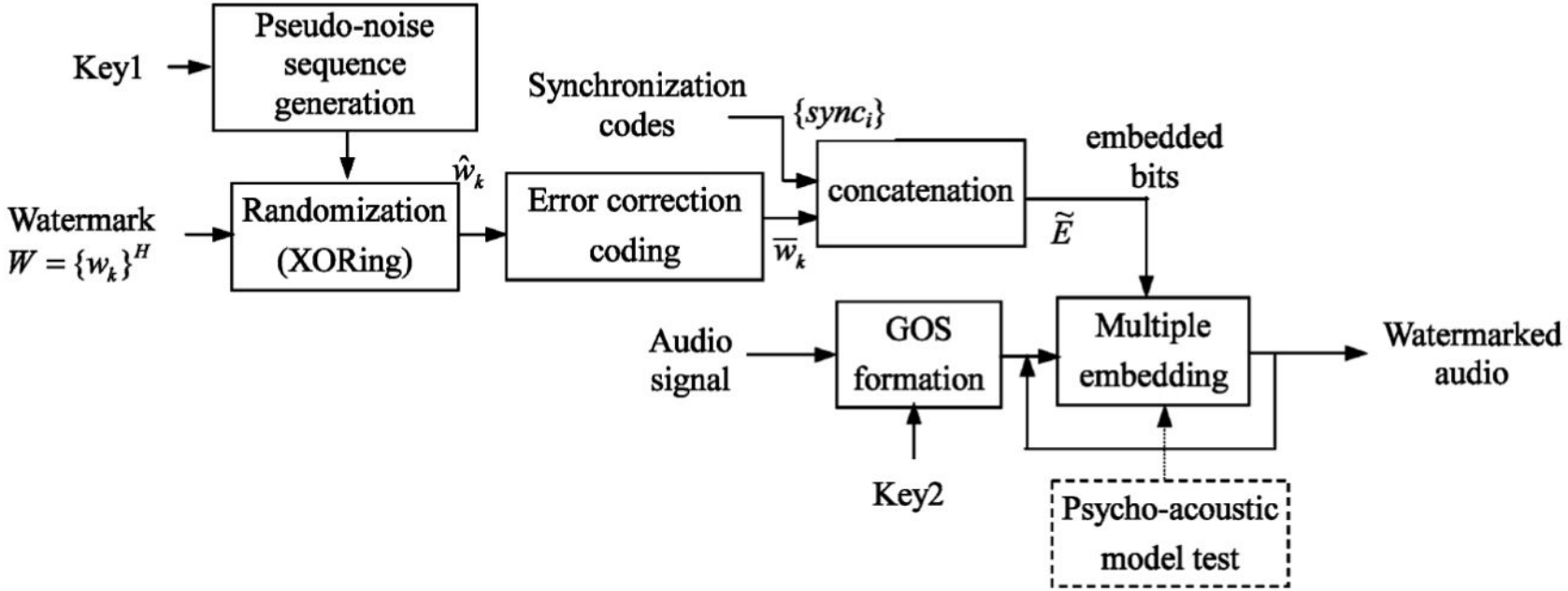
# AMPLITUDE HIDING - LIE, CHANG, 2006

La strategia è leggermente più complessa rispetto alla precedente.

Invece di inserire eco all'interno della traccia, si modifica l'*amplitude* dello spettro delle frequenze.

L'algoritmo di embedding lavora su **GOS** (Group of Samples) e tiene in considerazione una feature chiamata **AOAA** (Average of Absolute Amplitudes).

# AMPLITUDE HIDING, EMBEDDING




Robust and High-Quality Time-Domain Audio Watermarking Based on Low-Frequency Amplitude Modification, pag. 5

# EMBEDDING SCHEME

- La traccia viene partizionata in GOS;
- Ogni GOS viene partizionato in tre sezioni ($sec_1$, $sec_2$, $sec_3$) le cui lunghezze ($L_1$, $L_2$, $L_3$) possono essere uguali o differire;
- In funzione di $L_1$, $L_2$ ed $L_3$ vengono calcolati per ogni GOS i valori $E_1$, $E_2$ ed $E_3$, gli “item” della AOAA;
- Questi valori vengono ordinati in maniera crescente e rietichettati come $E_{min}$, $E_{mid}$, $E_{max}$

# EMBEDDING SCHEME

- Vengono poi calcolati i seguenti valori:
  - $A = E_{max} - E_{mid}$
  - $B = E_{mid} - E_{min}$
- La relazione tra A e B definisce gli “stati” del segnale:
  - Se $A \geq B$, allora ci troviamo nello stato “1”
  - Se $A < B$, allora ci troviamo nello stato “0”
- La definizione degli stati del segnale permette la procedura di *embedding*.

# EMBEDDING SCHEME

- Per inserire 1:
  - Se A - B ≥ di una soglia *(Thd1),* non viene effettuata alcuna operazione;
  - Altrimenti, si incrementa $E_{max}$ e si decrementa $E_{mid}$ di una quantità **δ***;*
- Per inserire 0:
  - Se B - A ≥ *Thd1,* non viene effettuata alcuna operazione;
  - Altrimenti, si incrementa $E_{mid}$ e si decrementa $E_{min}$ della stessa quantità **δ**.

# δ ED ω

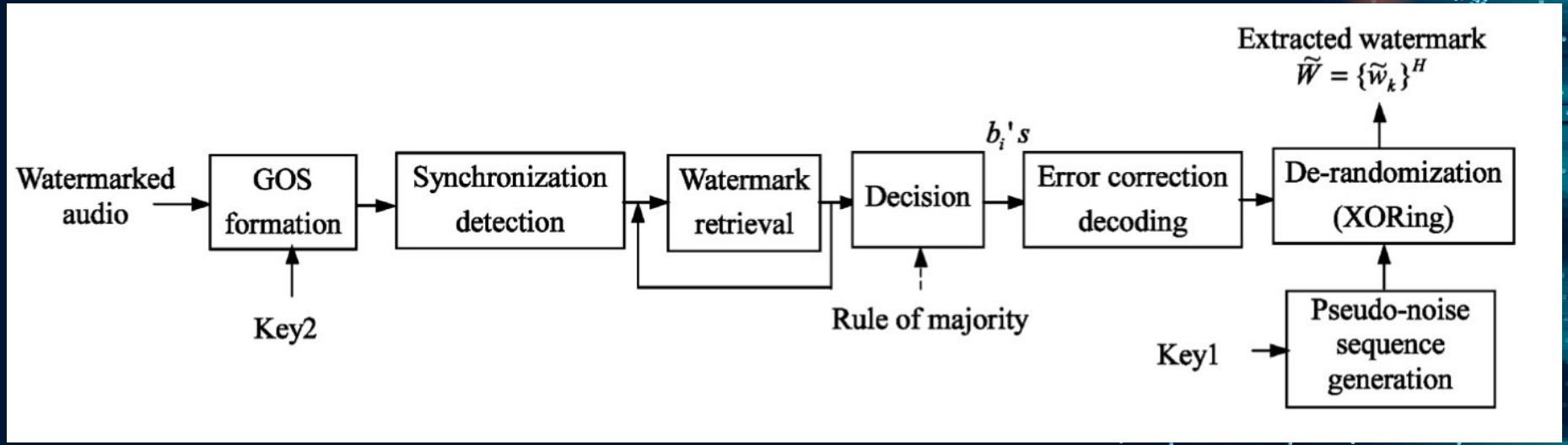
δ è una costante non negativa;

δ è ottenuto come combinazione tra A e B e *Thd1*.

A partire da δ si ottiene un altro parametro, ω, che rappresenta la variazione di amplitude dei samples

- Quando è necessario aumentare una componente dell'AOAA, ω assume il valore 1 + $\delta/E_{\{min, mid, max\}}$;
- Quando è necessario diminuire una componente dell'AOAA,, ω assume il valore 1 - $\delta/E_{\{min, mid, max\}}$.

# WATERMARK RETRIEVAL




Robust and High-Quality Time-Domain Audio Watermarking Based on Low-Frequency Amplitude Modification, pag. 5

# WATERMARK RETRIEVAL

L’estrazione dei dati embeddati è molto semplice e ricalca il procedimento dell’algoritmo precedente.

Si assume di conoscere il punto di partenza della modifica e le lunghezze delle sezioni $L_1$, $L_2$ ed $L_3$ di ogni GOS.

- Si raggruppano i sample della traccia audio modificata in GOS;
- Si calcolano i valori A e B come nell’algoritmo precedente;
- Per ogni GOS:
  - Se $A \geq B$ allora viene estratto 0;
  - Se $A < B$ allora viene estratto 1.

# Grazie per l’attenzione!