SMPTE ST 2112-10:2020

SMPTE STANDARD

Open Binding of Content Identifiers (OBID)

Table of Contents

445 Hamilton Avenue., White Plains, NY 10601
(914) 761-1100

Approved July, 26 2020

Index of figures

Index of tables

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 24TB.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.
Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

This clause is entirely informative and does not form an integral part of this Engineering Document.

This document specifies a means of binding content identifiers (Ad-ID and EIDR) to audiovisual content in such a way that it survives processing encountered on the way to the viewer, allowing the content to be accurately identified, regardless of how it got to the viewer.

Scope

This document describes a method of binding content identifiers to media, utilizing audio watermarking, allowing the content to be identified both electronically and acoustically.

1 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any clause explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; then formal languages; then figures; and then any other language forms.

2 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

SMPTE RP 2092-1:2015 - SMPTE Recommended Practice - Advertising Digital Identifier (Ad-ID®) Representations

SMPTE RP 2079:2017 - SMPTE Recommended Practice - Digital Object Identifier (DOI) Name and Entertainment ID Registry (EIDR) Identifier Representations

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

3.1 Ad-ID

industry standard identifier for advertising content

3.2 acoustic path

transmission of sound from a loudspeaker through the air to a microphone.

3.3 audio signal

single channel (monophonic) of audio

3.4 audio program

set of audio signals that are intended to be rendered simultaneously

Note to entry: for example, a stereophonic audio program or 5.1 multichannel audio program

3.5 audio watermark

data that is embedded in an audio stream in such a way that it can be extracted by a watermark decoder

3.6 EIDR

Entertainment Industry Data Registry.

Note to entry: Industry standard identifier for program content

3.7 embed

modify the audio signal by adding the audio watermark

3.8 embedder

tool that is able to embed the audio watermark in the audio signal

3.9 decoder

tool that is able to detect and extract an audio watermark embedded in a marked audio signal

3.10 symbol

representation of binary information in the audio watermark

3.11 marked audio

audio that has an audio watermark embedded in it

3.12 OBID

content identifier bound to media using the audio watermarking process defined in this standard

3.13 packet

sequence of contiguous symbols that contains independently recoverable information data

3.14 Linear PCM

Pulse Code Modulation audio that has not been bit rate reduced

3.15 synchronization symbol

specific symbol that defines the start of a packet

3.16 whitening

linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables whose covariance is the identity matrix, meaning that they are uncorrelated and each have variance 1. The transformation is called "whitening" because it changes the input vector into a white noise vector.

4 System Overview (Informative)

Audio watermarking is a signal processing technology that enables the embedding of digital data information into the audio signal itself. It implements audio processing solutions that are resilient to most types of audio distribution systems including lossy bitrate compression systems.

Figure 1: Audio watermark system architecture shows a block diagram illustrating the basic audio watermarking process. The original unmarked audio signal is sent to the watermark embedder system. This system performs signal processing to embed the watermark data into the audio signal itself. The audio signal at the embedder output carries the watermark information and can be distributed through traditional broadcast and/or storage systems. Common distribution platforms include, but are not limited to, terrestrial broadcast, cable/satellite networks, IP networks, mobile networks as well as ad hoc distribution in consumer applications that access either file-based or streaming content. At the audio receiver and/or rendering system, the watermark data present in the audio signal can be recovered by the watermark decoder.

File-based watermarking applications are usually storage applications where the desire is to identify content as opposed to distribution. Some examples of these applications include identification of syndicated programming, identification of network content, identification of a cable network program episodes, identification of a commercial, VOD, etc. The content usually resides on storage medium on a server in the highest quality format possible. A software application either on the server or on a remote networked computer processes the audio from the files on the server. The original content may or may not be copied, depending upon whether the desire is to keep an unwatermarked original for possible future versions. Most file-based watermarking happens through a media asset management system or or a transcoder system. Additionally, several systems are cloud based, which makes content available anywhere on the web globally. File based watermarking is advantageous in the fact that it can happen faster than realtime. Typically, a one-hour asset can be watermarked in less than a minute, depending upon the system used and the complexity of the asset (i.e. multiple languages, etc).

The audio watermark is embedded in the linear PCM audio before its distribution. Thus, this embedding should be performed before any bit rate reduction.

The audio watermark data is decoded from the linear PCM audio. If audio is received in another format, it shall be decoded to PCM audio prior to audio watermark decoding.

The audio watermarking system is also designed to support watermark decoding on acoustically-captured audio signals.

An example application is the use of a mobile device such as a smartphone to detect watermarked information in an audio signal reproduced through a TV set or loudspeakers in a living room.

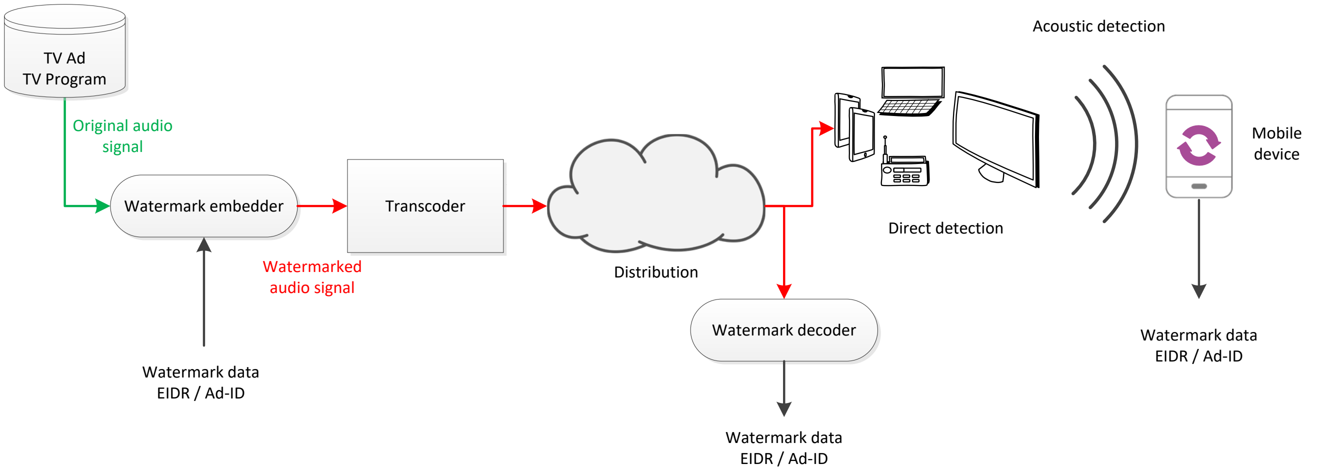

Figure 1: Audio watermark system architecture

When the audio program contains more than one mono audio channel, the same symbol shall be watermarked at the same time on all audio channels. This principle ensures that the watermarking information is recovered at the audio receiving side even if a channel downmix is performed between audio channels. This also ensures coherent watermarking information between channels at audio rendering, which is mandatory to decode the watermarking through the acoustic path.

Different content identifiers could be used for each language track in a multi-language transmission, enabling separate measurement of the viewership of the content by language.

5 Signal Processing Specifications

5.1 Baseband Layer

The baseband layer for the audio watermark shall be the linear PCM audio signal. The audio signal shall be split into consecutive audio blocks of equal duration (referred to as *T* below). Each audio block shall carry a single watermark symbol.

A watermark packet shall be defined as a symbol sequence carrying either Ad-ID or EIDR information.

The embedder shall embed the symbols corresponding to a watermark packet sequence continuously without gaps between packets. As a result, the periodicity of the packets (reciprocal of packet insertion rate) will be equal to the duration of a packet. The duration *T* of the audio block is defined in Table 1.

The embedding of the watermark symbol shall be achieved by phase modulating the audio signal.

The phase modulation shall be performed on the watermarking frequency band specified in Table 1.

Table 1: Watermarking parameters specification

Parameter description	**Parameter value**
Watermarking frequency band	4078.125 Hz - 8015.625 Hz
Sample Frequency: *Fs*	48 kHz
Block duration: *T*	16384 samples @ 48kHz

Note:

Several watermark technologies can coexist on the same frequency band.

The watermark technology described in this standard uses spread-spectrum modulation, Each symbol is spread over a wide range of frequencies (~4000 Hz), and a long time period (*T* ~ 0.34 s), and is therefore highly robust to interference from other watermark systems. It furthermore uses phase modulation embedding. It does not modify or use amplitude information, and therefore, by design, does not interfere with amplitude modulation watermark systems.

Non-interference with commonly used watermarks, including the watermarks specified in ATSC 3.0 A/334 and A/336, has been verified.

5.1.1 Symbol Definition

A watermark symbol shall be defined as a phase sequence corresponding to an audio block of *T* audio samples.

$\boldsymbol{N_S}$ = 272 symbols are defined. Those 272 symbols shall be divided in two groups:

- 256 data symbols: each one shall correspond to a byte value of watermarking information.

These shall be mapped to symbol indexes ranging from 0 to 255 in the symbol definition table.

The symbol index $\boldsymbol{i_S}$ in the table shall correspond to the data byte value the symbol is carrying.

- 16 synchronization dedicated symbols: used to define the start and the type of watermarked packets.

Each such dedicated symbol shall correspond to a *MediaID_header* value. They shall be mapped to the indexes ranging from 256 to 271 in the symbol definition table.

The mapping of the 16 synchronization symbols to the corresponding *MediaID_header* values is given in Table 2. Reserved values are intended to be used for future identifiers.

Table 2: Synchronization symbol to *MediaID_header* mapping

Synchronization symbol index $\boldsymbol{i_S}$	Corresponding *MediaID_header* value in binary format	Type of watermarked packet
256	0000	Ad-ID
257	0001	EIDR
258-271	0010-1111	Reserved

The complete Symbol Table (ST2112-10a) is provided as a normative non-prose element of this Standard.

This table contains 272 symbols, with each symbol having T float values (T=16384), as specified in clause 5.1, Physical Layer.

This results in a total of 4,456,448 values in the table, each delimited by a carriage return.

The mapping between the Symbol Table file values and symbolTable is the following;

$\forall i_S$ in [0,271], $\forall$ n in [0,T-1],symbolTable[i_S](n) shall correspond to value index i_S*T+n from (2112-20a) symbol table (where index starts from 0).

5.1.2 Watermarking of Multichannel Audio Program

Each full bandwidth audio signal in a program shall have the same symbol value embedded at the same instant in the program (at the same audio sample time). Band-limited signals such as low-frequency effects (LFE) signals are an exception; these need not have symbols embedded. All signals within an audio program, whether having symbols embedded or not, shall be kept in time synchronization. This may require buffering of any signals that have not had symbols embedded to compensate for the latency of the embedding process.

5.2 Watermark Embedding

5.2.1 Watermark Embedder Process

Ad-ID or EIDR data shall be converted into a symbol sequence according to the “Data Link Layer Architecture” section (below). After conversion, these symbols shall be embedded into the audio stream using a methodology that employs a pair of phase modulation steps.

First, the watermark information to be embedded shall be converted into a symbol sequence following clause 5.4 and clause 5.5 specifications. Then, the symbol sequence shall be embedded in the audio stream by the hereinafter-defined phase modulation process.

The process of embedding an OBID watermark is illustrated in Figure 2.

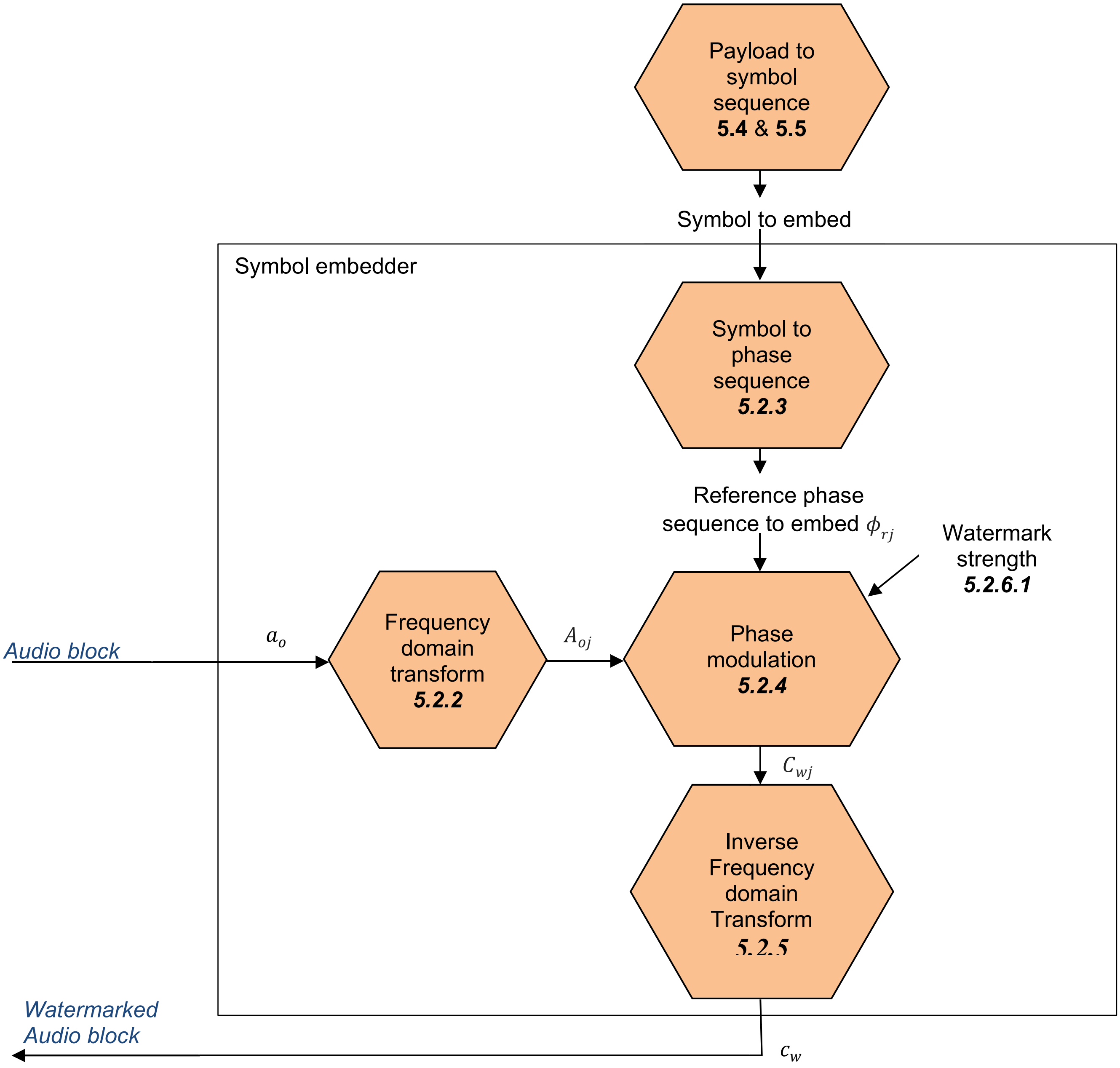

Figure 2: Watermark embedder block diagram

5.2.2 Embedder Frequency Domain Transform

The original audio block of *T* samples a_o shall be transformed into the frequency domain using discrete Fourier transformation:

> The audio block a_o shall be subdivided in N_B sub-blocks of *B* samples. Let a_{oj} designate a resulting sub-block.
>
> Each sub-block of original audio signal a_{oj} shall be transformed into the frequency domain using discrete Fourier transformation, such that it carries the phase ϕ_{oj} and magnitude A_{oj} of each Fourier coefficient.
>
> $$A_{oj} = DFT(\, a_{oj}) = |A_{oj}| \times e^{i\,\phi_{oj}}$$

B shall be equal or higher than 1024 to ensure a sufficient frequency resolution.

5.2.3 Symbols to Phase Sequence Mapping

The symbol S shall be embedded in the current audio block. Let i_S be the index in the symbol table of symbol S.

S is either a data symbol ($i_S < 256$) or a synchronization symbol ($i_S \geq 256$).

The temporal reference signal corresponding to S shall be retrieved from the symbol definition table:

> For $n \; in \; [0, T-1], r(n) = \; symbolTable[i_S](n)$

To generate the phase sequence of this symbol, this temporal signal shall be subdivided in N_B sub-blocks of *B* samples. This decomposition into sub-blocks shall be identical to the one applied to the original signal, i.e. time blocks shall have the same *B* samples, and there shall be the same number of blocks N_B as a result of this processing.

Each sub-block j resulting of this decomposition shall then be transformed into the frequency domain using Fourier transformation, such that it carries the phase ϕ_{rj} of each Fourier coefficient.

5.2.4 Phase Modulation

Let C_{wj} be the watermarked signal in the frequency domain, ϕ_{rj} the angle to watermark, and *F* the maximum phase deviation (the watermark's strength).

For all frequencies in the marking frequency band, the phase of the audio signal shall be modified in the direction of the selected symbol phase sequence with respect to the maximum deviation constraint *F*.

$$\Delta\phi_{oj} \; = sign\,(\phi_{rj} - \; \phi_{oj}\,) * \min\,(\,F, |\phi_{rj} - \; \phi_{oj}|)$$

$$C_{wj} \; = |A_{oj}| \; \times \; e^{i\,(\phi_{oj} + \Delta\phi_{oj}\,)}$$

Note: The amplitude of the audio signal is not modified as part of this process.

5.2.5 Embedder Inverse Frequency Domain Transform

The marked sub-block shall be transformed back to time domain using inverse discrete Fourier transformation on the modified Fourier coefficients:

$$c_{wj} = IDFT(\ C_{wj})$$

All sub-blocks shall be recomposed to create the watermarked signal c_w.

5.2.6 Symbol Embedding Recommendations (Informative)

5.2.6.1 Watermarking Strength

The strength employed for symbol embedding determines the content-dependent tradeoff between imperceptibility of the audio watermark to listeners versus the robustness of the audio watermark to distortions introduced by audio processing. This value is not normatively specified.

Implementers can compute a psycho-acoustic mask, as typically used in audio compression, to determine the maximum phase deviation applicable without being perceptible by human ears, for every frequency in the watermarking frequency band.

Acceptable results have been demonstrated by an implementation that employed on average a maximum phase deviation for symbols $F = 0.4 * \pi$

5.3 Watermark Decoding Process

5.3.1 Watermark Decoder Block Diagram

This clause describes how the watermarked information embedded in an audio stream shall be recovered and decoded, and is illustrated in Figure 3.

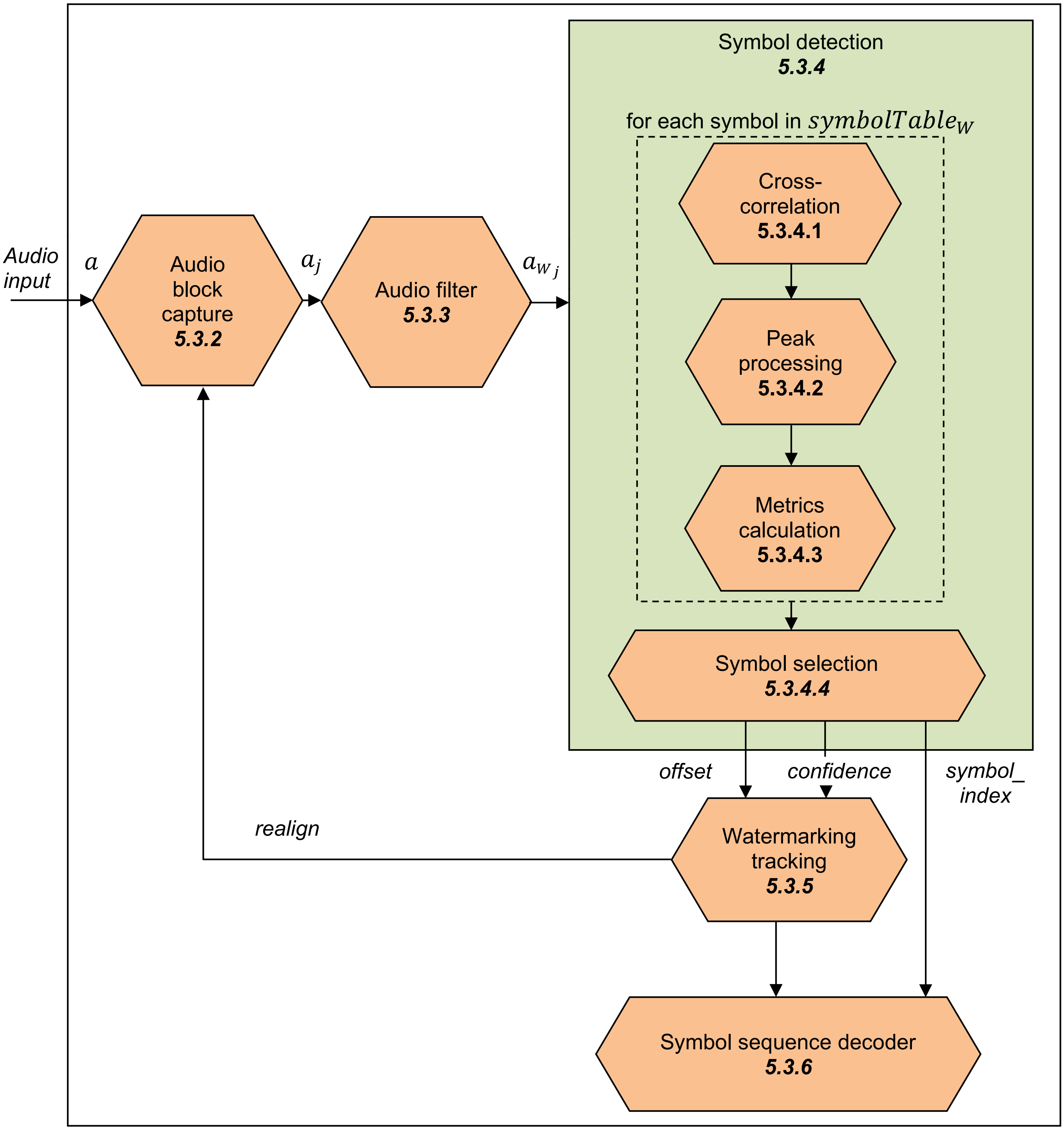

Figure 3: Watermark detector block diagram

5.3.2 Audio Block Capture

The audio block capture shall split the audio input signal in a_j blocks of *T* samples that will be processed by the symbol detector. It shall receive a temporal realignment order from the watermark tracking process as described in clause 5.3.5 below

The audio block capture shall shift the next audio signal by this $realign$ value in order to capture audio samples that are block aligned with the detected watermark. If $realign$ value is zero, the audio block capture shall process the next *T* samples.

5.3.3 Audio Filtering

The audio filter shall process the audio signal a_j block in order to optimize decoding robustness.

Implementers shall use a whitening technique on the input audio signal to reduce the host signal interference.

$$a_w = WhiteningFilter(a)$$

The same whitening technique shall also be applied on reference symbols present in $symbolTable$[1].

Define $symbolTable_W$ as the whitened reference symbols table.

$$\forall k \ in \ [0, N_S - 1], symbolTable_W[k] = WhiteningFilter(symbolTable[k])$$

Note: A possible implementation for $WhiteningFilter$ is as follows:

> Transform the temporal signal a into frequency domain using discrete Fourier transformation, such that it carries the phase ϕ and magnitude $|A|$ of each Fourier coefficient.
>
> Normalize each Fourier coefficient magnitude to unity.
>
> The resulting set of coefficients A_w is then transformed back to temporal domain a_w using inverse discrete Fourier transformation.

5.3.4 Symbol Detection

Embedded symbol detection shall be computed by cross-correlation between the input audio signal and possible symbols defined in $symbolTable_W$. For each symbol $s_k = symbolTable_W[k]$, $k \in [0, N_S - 1]$, the following processes shall be performed:

- Cross-correlation with filtered input signal a_{W_j}
- Peak processing to compensate acoustic path propagation
- Symbol metrics calculation

Then the symbol decoder can determine the detected symbol and its confidence.

---

[1] See clause 5.1.1 for $symbolTable$ and N_S definition

5.3.4.1 Cross-Correlation

The cross-correlation between each whitened reference symbol s_k from $symbolTable_W$ and a_{W_j} the filtered audio block of T samples may be computed as follows:

$\forall m \ in \ [-T/2, T/2-1]$, $c_{s_k}(m) = \frac{1}{T} * \sum_{i=0}^{T-1} s_k(i)(a_{W_j}(Cyclic(i+m))$,

Where $Cyclic(i+m) = \ (i \ + \ m \ + T)\% \, T.$[2]

This cross-correlation may be performed via the use of an FFT to reduce the computational complexity.

5.3.4.2 Peak Processing

When processing an acoustically-captured audio signal, the impulse response of the acoustic path propagates into the correlation function of the watermark detector, resulting in decreased detection performance. To compensate for this loss in decoding performance, peak processing may be performed as following:

$c'_{s_k}(m) = AcousticProcessing(c_{s_k}(m))$

Note: A possible implementation for $AcousticProcessing$ is shown below:

Define $c_{s_{knorm}} = \frac{c_{s_k}}{\sigma}$ and $\sigma^2 = (\sum_m c_{s_k}(m)^2)/T$. Define $filter(b, a, x, zi)$ a rational transfer function by $H(z) = \frac{b}{1+az^{-1}}$ and zi as initial condition for the filter delay.

Compute[3] $y_{s_k} = \alpha * (filter\big(b, a, {c_{s_{knorm}}}^2, zi\big) - \beta)$. Define M the position of the absolute maximum in y_{s_k}.

$c'_{s_k}$is defined as $c_{s_k}(m) \ \forall m \ \in [0, T-1]/\{\ M\}$,

and $c'_{s_k}(\mathrm{M}) = y_{s_k}(M)$ if $|c_{s_k}(\mathrm{M})| < |y_{s_k}(M)|$, else $c'_{s_k}(\mathrm{M}) = \ c_{s_k}(M)$.

In cases of direct connection, when there is no acoustic path transmission to compensate, peak processing is not useful. In that case $c'_{s_k}(m)$ is defined as $c_{s_k}(m)$.

5.3.4.3 Symbol Metrics Calculation

The following parameters shall be calculated from the cross-correlation for each symbol $s_k = symbolTable_W[k]$:

- The correlation standard deviation, ${\sigma^2}_{s_k} = (\sum_m {c'}_{s_k}(m)^2)/T$
- The processed correlation maximum absolute value, $maxCorr_{s_k} = max_m|c'_{s_k}(m)|$
- The temporal position of the processed correlation maximum absolute value. It represents the time-shift between the signals, $offset_{s_k} = argmax_m|c'_{s_k}(m)|$, centered between $[-T/2, T/2-1]$.

---

[2] % refers to the modulo operator.

[3] The following settings may be used: b= 0.000719557841560639, a = -0.999280700978101, zi = 1, α=16.7756616642476, β=1.

5.3.4.4 Symbol Selection

Define the average standard deviation $\sigma = sqrt((\sum_0^{N_S-1} \sigma^2{}_{s_k})\,/N_S)$.

The symbol index selected shall be the index of the symbol maximizing the correlation absolute value.

$$symbol\_index = argmax_k(maxCorr_{s_k})$$

Its confidence value shall be defined as the maximum absolute value of its correlation normalized by the average standard deviation $confidence = maxCorr_{s_{symbol\_index}}/\sigma$.

Its offset shall be defined as $offset = offset_{s_{symbol\_index}}$.

5.3.5 Watermarking Tracking

The watermarking tracking module shall determine whether the detected symbol is reliable or not. This may be implemented by comparing the detected symbol confidence with a detection threshold.

If a symbol is detected reliably, the temporal realignment order $realign$ sent to the audio capture module shall correspond to $offset$ which corresponds to the timeshift observed between the watermark and the signal.

Otherwise, the temporal realignment order $realign$ sent to the audio capture module shall be set to zero.

5.3.6 Symbol Sequence Decoder

The symbol sequence decoder shall perform the reliable detected symbols sequence aggregation, the error detection control and the watermarking information decoding (either Ad-ID or EIDR).

A suggested implementation uses a FIFO buffer to store the incoming reliable detected symbols with their attached detection time.

When one of the 16 synchronization symbols is detected, the packet structure and size shall be identified as specified in clause 5.4.1.

Once a full packet has been received, the parity symbol value shall be checked as specified in clause 5.4.2 to verify the packet validity.

The watermark Ad-ID or EIDR value shall then be extracted as described in clause 5.5.

5.4 Data Link Layer Architecture

5.4.1 Packet Structure

A packet shall be defined as a series of contiguous symbols embedded in the audio signal sequentially at regular time intervals *T* starting with a specific synchronization symbol.

The synchronization symbol index shall represent the packet's *MediaID_header* value (see clause 5.1.1 for more information).

Two types of packets shall be defined:

- Ad-ID packets that carry Ad-ID information. An Ad-ID packet shall start with a synchronization symbol *adid_sync*, shall be followed by a sequence of 4 data symbols *adid_payload*, and shall end with a parity symbol.
- EIDR packets that carry EIDR information. An EIDR packet shall start with a synchronization symbol *eidr_sync*, shall be followed by a sequence of 12 data symbols *eidr_payload*, and shall end with a *parity_symbol*.

Table 3: Syntax of Ad-ID packet structure

Syntax	No. of symbols
`adid_packet(){`	
`adid_sync`	1
`adid_payload`	4
`parity_symbol`	1
}	

Table 4: Syntax of EIDR packet structure

Syntax	No. of symbols
`eidr_packet(){`	
`eidr_sync`	1
`eidr_payload`	12
`parity_symbol`	1
}	

adid_sync symbol shall be the symbol corresponding to *MediaID_header* 0.

eidr_sync symbol shall be the symbol corresponding to *MediaID_header* 1.

adid_payload shall be a sequence of 4 data symbols, each one corresponding to 8 bits of Ad-ID payload data.

eidr_payload shall be a sequence of 12 data symbols, each one corresponding to 8 bits of EIDR payload data.

parity_symbol is a data symbol corresponding to 8 bits of payload data parity check. See clause 5.4.2 for further details.

5.4.2 Parity Symbol

All packets shall be terminated by a *parity_symbol*.

Each bit of the *parity_symbol* shall be the inverse of the exclusive-or sum of the corresponding payload bits (i.e. having the same position in the binary representation).

The error detection control shall be implemented as follows:

During watermark decoding, the *parity_symbol* shall be used to detect watermark decoding errors. The packet shall be considered invalid if there is at least one bit of the *parity_symbol* which is not the inverse of the XOR sum of the corresponding decoded payload bits.

5.5 Payload Structure

The structure of the payload depends on packet type (whether it is an Ad-ID or EIDR packet). The syntax of the Ad-ID payload shall be as shown in Table 5. The syntax of the EIDR payload shall be as shown in Table 6.

Table 5: Syntax of Ad-ID payload structure

Syntax	No. of bits	Format

`adid_payload(){`
`adid_value`
}
	32	Unsigned integer LSB first

The Ad-ID payload specified in Table 5 shall be the 32-bit hashed-value representation of the 64 bit compact Ad-ID identifier, specified in SMPTE RP2092-1:2015

See the following example, illustrating the equivalence of a full Ad-ID, its compact Ad-ID identifier, and the 32-bit version of that identifier:

```
<adid code="A2ID0517000" cid="10010198">
  <prefix pid="100001">A2ID</prefix>
  <code>0517000</code>
  <adid_fullcode>A2ID0517000</adid_fullcode>
  <guid>c855babb</guid>
  <slate>
```

Table 6: Syntax of EIDR payload structure

Syntax	No. of bits	Format

`eidr_payload(){`
`eidr_value`
}
	96	Unsigned integer LSB first

See SMPTE RP2079:2013 for syntax of the compact binary representation of the Canonical EIDR Identifier.

As specified above, the payload symbols shall be watermarked starting from the one containing LSB byte data to the one containing MSB byte data. Each payload byte shall also be encoded LSB first so the corresponding symbol index in the symbol table is the inverse bit order value of the transmitted byte value.

5.6 Ad-ID and EIDR Watermarking Examples

5.6.1 Ad-ID Watermarking

Ad-ID watermarking with *MediaID_header* = 0 and Ad-ID value = 0x0A0B0123 parameters shall be achieved by embedding the following symbol sequence:

Ad-ID Sequence	Data value (or header value for sync symbols)	Symbol index to embed
Synchronization symbol	*MediaID_header* = 0	256
Payload symbols	0x23	196
	0x01	128
	0x0B	208
	0x0A	80
Parity symbol	0xDC	59

5.6.2 EIDR Watermarking

EIDR watermarking with *MediaID_header* = 1 and EIDR value = 0x0A0B0C0D0E0F112233445566 parameters shall be achieved by embedding the following symbol sequence:

EIDR Sequence	Data value (or header value for sync symbols)	Symbol index to embed
Synchronization symbol	*MediaID_header* = 1	257
Payload symbols	0x66	102
	0x55	170
	0x44	34
	0x33	204
	0x22	68
	0x11	136
	0x0F	240
	0x0E	112
	0x0D	176
	0x0C	48
	0x0B	208
	0x0A	80
Parity symbol	0x89	145

Annex A: Replacing Watermarks (Overmarking) (Informative)

OBID embedding offers the option to replace a mark with another watermark while preserving the subjective quality of the content, or it can be configured exclusively to insert a mark.

This annex gives a suggestion to address the replacement of a watermark present in an audio stream by another watermark with respect to the subjective quality of the content.

This subjective-quality-friendly replacement option requires the implementation of a detector within the embedder to identify the symbol embedded previously prior to the embedding of the mark. It consequently adds a minimum latency of 1024 samples to the embedder latency.

A detector can be implemented within the embedder, in order to identify on the current audio block the symbol S_{prev} embedded previously and its sub-block temporal index k. Let r_prev_k be its k sub-block temporal reference signal.

Once identified, it can be replaced: following clauses 5.2.2 and 5.2.3,

> S is the symbol to embed on the current audio block. Let r_j be its j sub-block temporal reference signal.
>
> Define the replacement symbol $S_{repl} = S - \alpha * S_{prev}$. [4]
>
> The j sub-block temporal signal corresponding to S_{repl} is r_repl_j defined by:
>
> $$\forall n \; in \; [0, B-1], \; r\_repl_j\,(n) = r_j(n) - \alpha * r\_prev_k\,(n)$$
>
> Each sub-block $repl_j\,(n)$ is transformed into the frequency domain using Fourier transformation, such that it carries the phase ϕ_{replj} of each Fourier coefficient.

The replacement is the application of the phase modulation embedding defined in clause 5.2.4, utilizing the angle to watermark ϕ_{replj}, and maximum deviation constraint $F * A_{replace}$.

The embedder inverse frequency domain transform isthen applied as specified in clause 5.2.5.

---

[4] $0 \le \alpha \le 1$, α and $A_{replace}$ should be adjusted according to implementation such that symbols resulting from a single embedding and replacement have similar imperceptibility and watermark confidence.

Bibliography

ATSC A/334:2016 – Audio Watermark Emission

ATSC A/336:2017 - Content Recovery in Redistribution Scenarios