SMPTE STANDARD

Open Binding of Distribution Channel IDs and Timestamps (OBID-TLC)

Table of Contents

Approved April 16, 2018

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 distribution channel identifiers (identifying entities involved in the distribution of the content) and timestamps (identifying the timeline of the content) to audiovisual content in such a way that it survives processing encountered on the way to the viewer, allowing the distribution channel and the content was distributed to be accurately identified, regardless of how it got to the viewer. This is collectively known as OBID-TLC (Open Binding of IDs – Timestamp and Labeling with Content distribution identifier)

1. Scope

This standard describes a method of binding Distribution IDs and Timestamps to media, utilizing audio watermarking, allowing this information to be detected both electronically and acoustically.

2. 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 labelled 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.

3. 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.

ISO 26324:2012, Information and documentation -- Digital object identifier system, International Organization for Standardization

4. Terms and Definitions

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

4.1 acoustic path

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

4.2 audio program

set of audio signals intended to be rendered simultaneously, such as a stereophonic audio program or 5.1 multichannel audio program

4.3 audio signal

single monophonic audio channel

4.4 audio watermark

data embedded in an audio stream such that it is minimally perceptible to a listener but detectable by a watermark decoder

4.5 decoder

device used to detect and extract an audio watermark embedded in a marked audio signal

4.6 Distribution Channel ID

unique identifier associated with an entity involved in the distribution of the content. This ID can overlay one or more advertising and/or programming elements.

4.7 Distribution Mark

a Distribution Channel ID and its associated Timestamp that are embedded together in the audio signal at a certain point of the distribution chain

4.8 distribution order

position of the embedder in the distribution chain

4.9 EIDR

Entertainment Industry Data Registry. Industry standard identifier for program content

4.10 embed

modify the audio signal by adding the audio watermark

4.11 embedder

device used to embed the audio watermark in the audio signal

4.12 marked audio

audio that has an audio watermark embedded in it

4.13 stuffing packet

sequence of 10 contiguous stuffing symbols that are used to fill time slots that do not contain a distribution mark

4.14 symbol

a phase sequence corresponding to an audio block of *T* audio samples

4.15 synchronization symbol

specific symbol that defines the start of a packet

4.16 time slot

Time interval that can contain a watermark packet or a stuffing packet

4.17 Timestamp

time instant at which the content was generated and broadcast

4.18 watermark cell

sequence of four contiguous packet time slots

4.19 watermark packet

sequence of 10 contiguous symbols that contains independently recoverable data.

4.20 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.

Note: The transformation is called "whitening" because it changes the input vector into a white noise vector.

5. System Overview (Informative)

Audio watermarking is a signal processing technology that enables the embedding of data into the audio signal itself.

This standard enables the binding of distribution related information (Distribution Channel ID and Timestamps) to the content. The distribution information binding occurs separately from the binding of OBID marks defined in SMPTE ST2112-10 standard and does not interfere with it.

Up to four independent Distribution Marks (each consisting of a Distribution Channel ID and its associated Timestamp) can be bound successively to the audio content by the various entities involved in its handling, prior to its reception by the viewer who is receiving the media. Refer to Figure 1 for guidance on where in the distribution chain these Distribution Marks (labelled “channel/timestamp” on the figure) can be bound to the media.

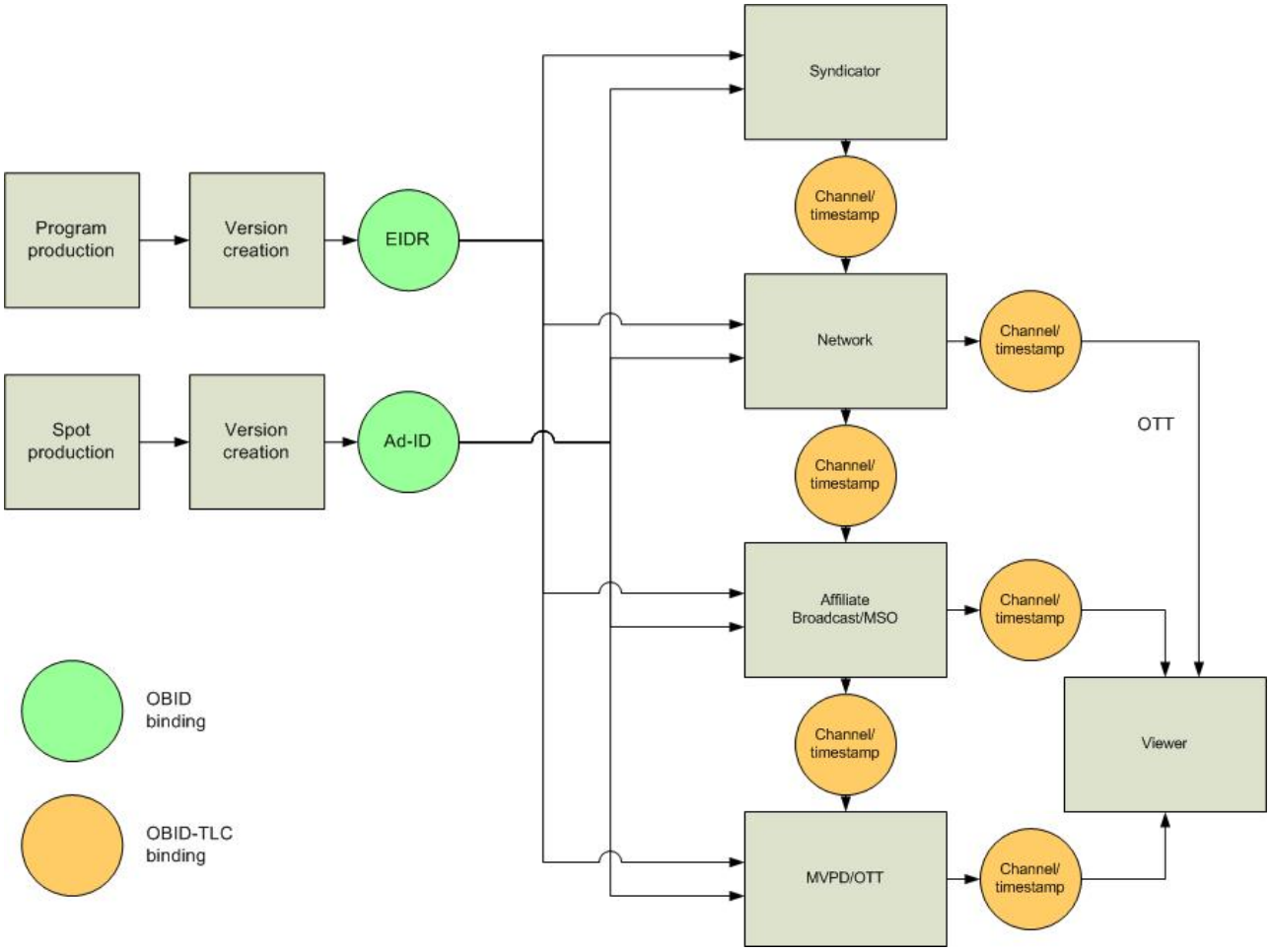

Figure 1: Binding of OBID and OBID-TLC marks in the media distribution chain

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

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

The audio watermarking system is also designed to support watermark detection and 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. This example is depicted in Figure 2.

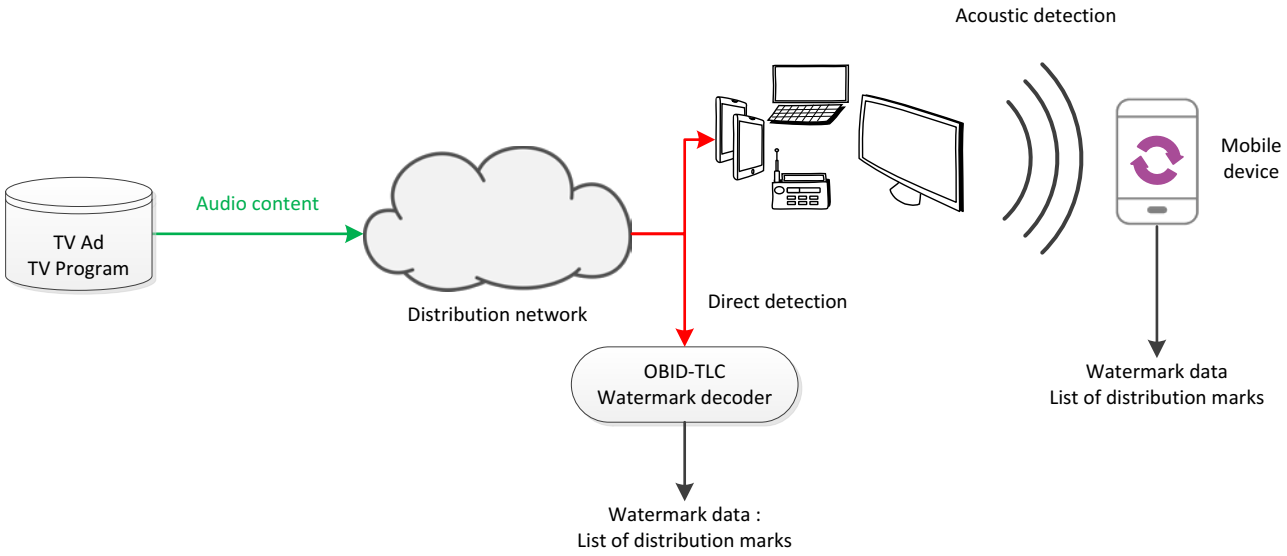

Figure 2: Audio watermark detection

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

6. Signal Processing Specifications

6.1 Physical Layer

The physical layer of the audio watermark is the audio signal itself. The audio signal shall be split into consecutive audio blocks, each of equal duration. This duration is referred to as *T*. Each audio block shall carry one single watermark symbol.

The duration *T* of the audio block shall be defined to be 32768 samples of a 48 Khz sampled audio signal.

The embedding of the watermark symbol shall be performed by applying phase modulation to 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	1968.75 Hz - 4078.125 Hz
Sample Frequency: *Fs*	48 kHz
Block duration: *T*	32768 samples @ 48kHz

Note 1: Several watermark technologies can coexist on the same frequency band.

Note 2: The watermark technology described in this standard uses spread-spectrum modulation – each symbol is spread over a wide range of frequencies (greater than 2000 Hz), and a long time period (T ~ 0.68 s) - and is therefore highly robust to interference to and from other watermark systems, as evidenced by comprehensive testing performed in the development of this standard. It furthermore uses phase modulation encoding, meaning that it does not modify or use amplitude information, and is therefore by design non-interfering with amplitude-modulation watermark systems.

6.1.1 TLC Symbol definition

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

$\boldsymbol{N_S}$ = 517 symbols are defined. These 517 symbols shall be divided into three groups:

1. 512 data symbols: each symbol shall correspond to 9 bits of watermarking information.

   They shall be mapped to symbol indexes ranging from 0 to 511 in the symbol definition table.

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

2. 4 synchronization dedicated symbols: these shall be used to define the start of watermarked packets and their time slot position in the watermark cell.

Each symbol shall correspond to one time slot position in the watermark cell as defined in the following table.

Table 2 : Synchronization symbol indexes and their corresponding time slot positions in the watermark cell

Synchronization symbol	Synchronization symbol index $\boldsymbol{i_S}$	Time slot position in watermark cell
S_{sync1}	512	1
S_{sync2}	513	2
S_{sync3}	514	3
S_{sync4}	515	4

3. A stuffing symbol S_{st} at symbol index $\boldsymbol{i_S} = 516$ shall be used to fill empty time slots in the watermark cell.

A Symbol Table (ST 2112-20a) is provided as a normative non-prose element of this Standard.

This table contains 517 symbols, as specified earlier in this clause 6.1.1 Symbol Definition, with each of those having T float values (T=32768), as specified in clause 6.1.

This results in a total of 16,941,056 values in the table, each delimited by a carriage return.

The mapping between the Symbol Table file values and $symbolTable$ shall be defined by the following:

$\forall i_S \; in \; [0{,}516], \forall n \; in \; [0, T-1], symbolTable\,[\,i_S\,](n)$ shall correspond to value index [1] $i_S * T + n$ from the Symbol Table (ST2112-20a).

[1] index starts from 0

6.1.2 Definition of a TLC watermark packet

A watermark packet shall be defined as a sequence of 10 watermark symbols carrying one Distribution Mark (one Distribution Channel ID and its associated Timestamp). The 10 symbols corresponding to a watermark packet shall be embedded sequentially without intervals between symbols. The watermark packet shall have a duration of 327680 audio samples. The first symbol of a watermark packet shall be a synchronization symbol and the nine remaining symbols shall be data symbols.

6.1.3 Definition of a TLC stuffing packet

A stuffing packet shall be defined as a sequence of 10 stuffing symbols S_{st}. The 10 stuffing symbols shall be embedded sequentially without intervals between symbols. The stuffing packet shall have a duration of 327680 audio samples.

6.1.4 Definition of a TLC watermark cell

A watermark cell shall be defined as a sequence of four contiguous packet time slots. Each time slot shall have a duration of 327680 audio samples. The watermark cell shall have a duration of 1310720 audio samples (40*T samples). The watermark cell shall be repeated periodically with a period equal to its duration.

The time slots defined in the watermark cell shall be filled by either watermark packets or stuffing packets. The first time slot of a watermark cell shall always carry a watermark packet (no stuffing packet shall be allowed in the first time slot position)

Watermark packets present in the watermark cell shall start with the synchronization symbol corresponding to their time slot position in the cell:

- Watermark packet present at first time slot position shall start with synchronization symbol S_{sync1}
- Watermark packet present at second time slot position shall start with synchronization symbol S_{sync2}
- Watermark packet present at third time slot position shall start with synchronization symbol S_{sync3}
- Watermark packet present at fourth time slot position shall start with synchronization symbol S_{sync4}

Empty time slots shall be filled with stuffing packets.

Watermark packets and stuffing packets shall start and end exactly at their corresponding time slot position boundaries and shall not interfere with other time slot positions. As a consequence, symbols corresponding to watermark packets or stuffing packets in the cell shall be sample-wise aligned within a grid of T samples. See Figures 3 and 4.

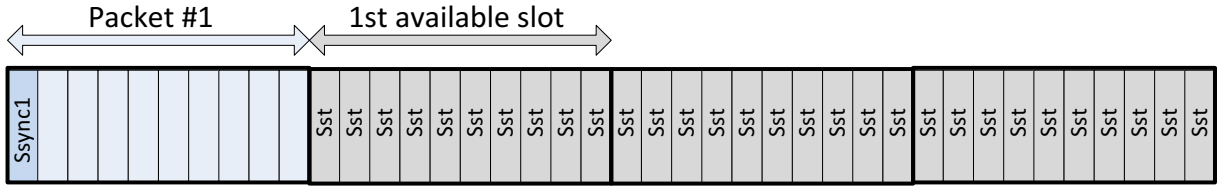

Figure 3 - Cell structure example – 1 slot used

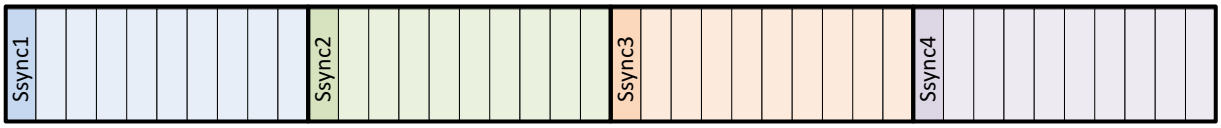

Figure 4 - Cell structure example – 4 slots used

The embedders present in the distribution chain shall each embed their respective Distribution Marks in the first available packet time slot of the cell. The first embedder in the distribution chain shall create the cell structure with its watermark packet embedded at the first time slot position, starting with the synchronization symbol S_{sync1} and shall fill the three other time slots with stuffing packets.

6.1.5 Definition of a TLC-compliant embedding system

The distribution order of an embedder $Dorder$ shall be defined as the embedder's position in the distribution chain.

Figure 5 gives an example of watermark embedders in the distribution chain and their respective distribution orders.

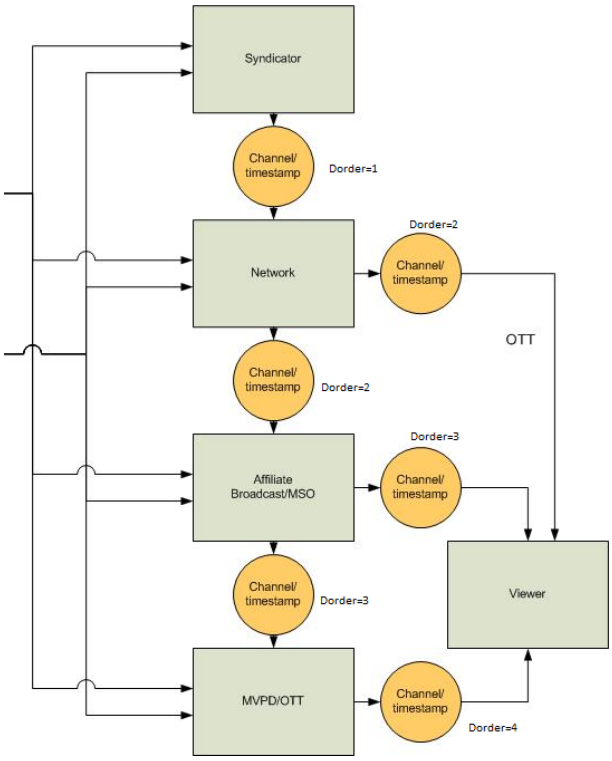

Figure 5: Embedder distribution order in an example distribution chain

The embedder shall implement the following rules:

- The embedder shall automatically detect its distribution order in the distribution chain.
  - If the input signal is not watermarked, it shall take $Dorder = 1$

- If the input signal is watermarked, it shall take $Dorder$ corresponding to the first available time slot. If no time slot is available (four distribution marks have already been watermarked), it shall take $Dorder = 5$.

- When $Dorder = 1$ (first in the distribution chain), the embedder shall embed its distribution mark at time slot 1 and fill the three other time slots with stuffing packets.

  Embedding of the first distribution mark shall be done using *normal_embedding* mode as specified in clause 6.2.2.1.

  Embedding of the three stuffing packets (also described as *stuffing_embedding* mode) shall be performed with reduced watermarking strength as specified in clause 6.2.2.2

- When $Dorder\ greater\ than\ 1$ and $Dorder\ less\ than\ or\ equal\ to\ 4$, the embedder shall replace the stuffing packet present at the time slot position corresponding to its distribution order by its Distribution Mark. This is done using the special *stuffing_replacement* embedding mode specified in clause 6.2.2.3. The embedder shall not modify the audio signal outside of the boundaries corresponding to its time slot. Replacement of stuffing symbols with symbols corresponding to the embedder Distribution Mark shall performed sample-wise aligned; that is: the first sample position of the new symbol is being embedded shall correspond to the first sample of the stuffing symbol being replaced.

- When $Dorder\ greater\ than\ 4$, there is no available time slot in which to insert the Distribution Mark. The signal shall not be modified. The embedder shall keep on scanning the input signal, and checking its distribution order. The embedder shall resume embedding according to the above rules when the detected distribution order becomes $less\ than\ or\ equal\ to\ 4$.

- There shall be no overmarking of a previously embedded mark.

The changes in the distribution chain shall be handled as following:

- Embedder startup in the distribution chain:
  - On startup, an embedder shall first check to determine whether there is a cell structure embedded in the incoming audio signal and shall start embedding in the first available time slot (i.e. the minimum time slot position available).

    The maximum latency to embed on a stream carrying no watermark (i.e. $Dorder\ equal\ to\ 1$) shall be $2 * T$ samples.

    If the stream already has a cell structure (i.e. $Dorder\ greater\ than\ 1$) and there is at least one time slot available, the maximum latency to embed shall be Tcell samples.

  - Downstream embedders shall detect the change in the distribution chain (input signal that was not watermarked that becomes watermarked, or time slot that was available for embedding that becomes filled with a distribution mark) and immediately stop embedding. Embedders shall identify the first available time slot position, update their $Dorder$ and embedding mode according to the new parameters and resume embedding with between [Tcell, 2*Tcell] samples latency.

- In the case of the stoppage or removal of an embedder from the distribution chain, whereupon a slot becomes available, the downstream embedders shall update their $Dorder$ and embedding mode and embed in the first available slot with the following average latency :
  - Between $[T, 2 * T]$ samples when the embedder stopped was the first in the distribution chain ($Dorder\ equal\ to\ 1$). As soon as the decision is taken, the downstream embedder shall update its $Dorder$ to 1 and start embedding a new cell structure starting with its distribution mark at time slot 1 and filling the other time slot with stuffing packet.

- Between [Tcell, 2*Tcell] samples when the distribution order of the stopped embedder was $Dorder\ greater\ than\ 1$. The downstream embedders shall decrement their $Dorder$ and embed their respective Distribution Marks in the positions corresponding to the updated $Dorder$.

In case of a signal with very low RMS audio level in the band of interest and an absence of watermark detection, the signal should be considered as not significant. In that case, the application of changes in the distribution chain rules shall be postponed for at least 10s (the embedder should maintain its current $Dorder$ and assume that the cell structure and timing remain the same and the time slot boundaries keep repeating with their usual period).

Once this safety period is over, the embedder can assume that there has been a change in the distribution chain and proceed to the changes in the distribution chain rules described above.

6.1.6 Watermarking of a Multichannel Audio Program

In a 5.1 channel audio, LFE channel may not be embedded. All other channels shall be embedded.

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. Watermark Embedding

6.1.7 Watermark Embedder Process

This clause defines how the watermark information shall be embedded in the original audio stream.

As specified in clause 6.1.5, the embedder shall first detect its distribution order in the distribution chain.

The watermark information to be embedded shall be converted to a symbol sequence following the specifications given in clause 6.4 and clause 6.5.

The symbol sequence shall then be embedded in the audio stream by the hereinafter defined phase modulation process, as shown in Figure 6.

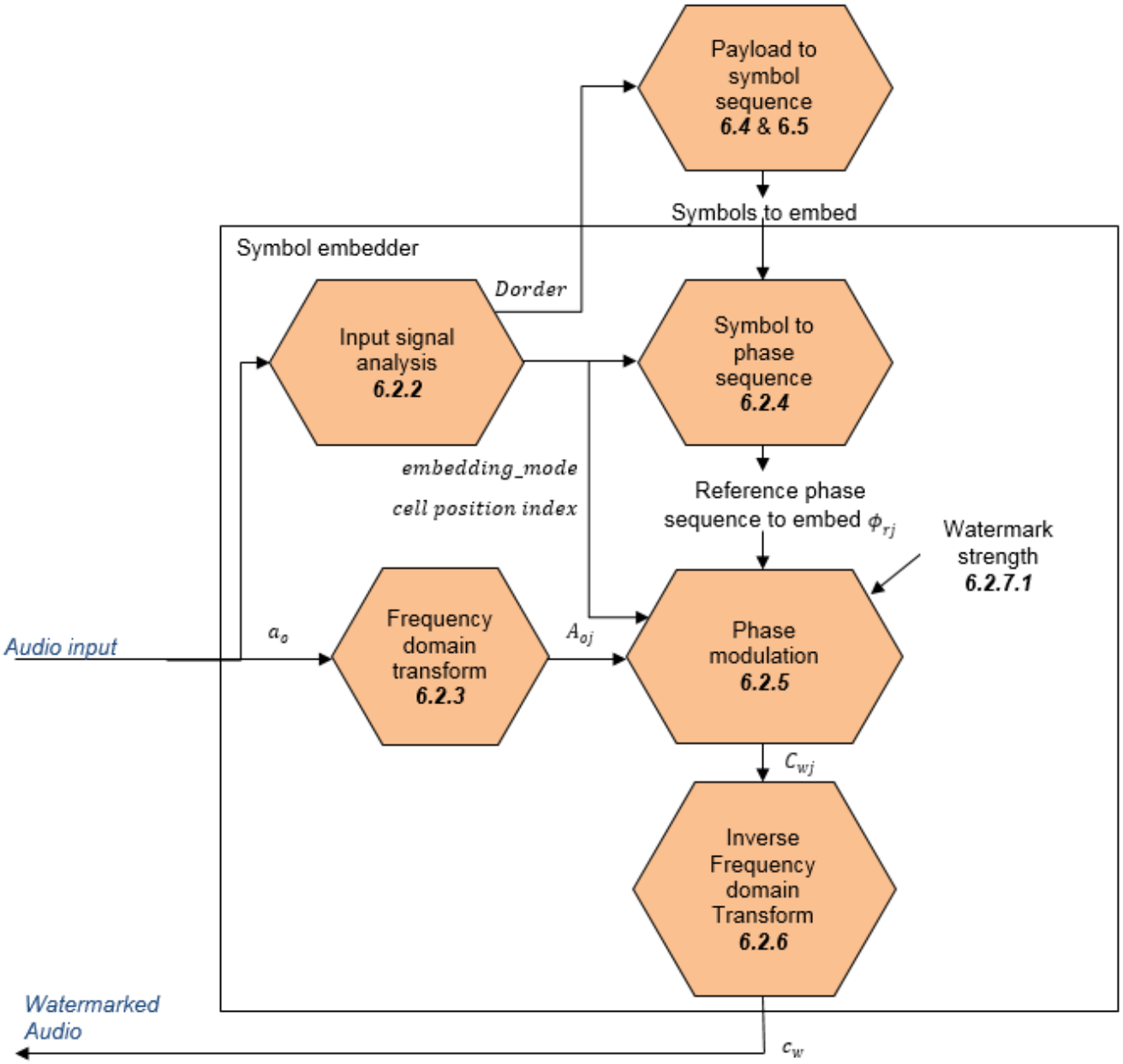

Figure 6: Watermark embedder block diagram

6.1.8 Input Signal Analysis

This module shall have the responsibility of monitoring the audio input signal and determining the key parameters that are required by the embedding process:

- It shall detect the embedder $Dorder$
- If the input signal already has a watermark cell structure embedded ($Dorder\ equal\ to\ 2, 3\ or\ 4$), it shall detect the time slot boundaries and align the audio block to embed with the incoming symbols
- It shall detect changes in the distribution chain as specified in clause 5.1.5.

- It shall determine the embedding mode depending on time slot position and its $Dorder$. The phase modulation will be applied with a given watermarking strength, modulated by a watermarking attenuation factor A defined in accordance with the embedding mode.

6.1.8.1 *normal_embedding* mode

normal_embedding mode shall be applied by the embedder with $Dorder\ equal\ to\ 1$ to watermark its Distribution Mark during 1st time slot embedding.

In *normal_embedding* mode symbols are retrieved from $symbolTable$ and watermarking shall be applied with full watermarking strength. The watermarking attenuation factor is therefore $A_{norm}\ equal\ to\ 1$.

6.1.8.2 *stuffing_embedding* mode

stuffing_embedding mode shall be applied by the embedder with $Dorder\ equal\ to\ 1$ to watermark the stuffing packets in the second, third and fourth time slots.

In *stuffing_embedding* mode, stuffing symbols are retrieved from $symbolTable$ and watermarking shall be applied with a reduced strength. The watermarking attenuation factor shall be $A_{Stuff}\ equal\ to\ 0.7071068$.

6.1.8.3 *stuffing_replacement* mode

stuffing_replacement mode shall be applied by embedders with $Dorder\ equal\ to\ 2, 3\ or\ 4$ to watermark their respective Distribution Marks during their respective time slots to replace the previously-embedded stuffing symbols.

In *stuffing_replacement* mode, the stuffing symbol already embedded in a previous embedding shall be replaced by the symbol to be embedded.

In *stuffing_replacement* mode, symbols to embed shall be retrieved from $symbolTable$ and watermarking shall be applied with the watermarking attenuation factor $A_{replace}$.

The value of $A_{replace}$ is implementation dependent. It shall be calculated such that symbols resulting from embedding in *normal_embedding* mode and in *stuffing_replacement* mode have similar imperceptibility and watermark confidence. See clause 6.2.7.2 for a method that may be used to address the replacement of a stuffing watermark.

6.1.9 Embedder frequency domain transform

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

The audio block a_o shall be subdivided into N_B sub-blocks of *B* samples. Define a_{oj} as a resulting sub-block.

Each sub-block a_{oj} of the original audio signal is 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 to or greater than 1024 to ensure sufficient frequency resolution.

6.1.10 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 in ST 2112-20a.

S is either a data symbol ($i_S < 512$), a synchronization symbol ($i_S \geq 512\ and\ i_S < 516$) or a stuffing symbol ($i_S\ equal\ to\ 516$)

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

For $n\ in\ [0, T-1]$,

$r(n) = symbolTable[i_S](n)$

To generate the phase sequence of this symbol, this temporal signal $r(n)$ shall be subdivided into 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 r_j resulting from 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.

6.1.11 Phase Modulation

Let A_{oj} be the original signal and $\phi_{oj}$the original signal angle.

$$A_{oj} = |A_{oj}| \times e^{i\,\phi_{oj}}$$

Let C_{wj} be the watermarked signal, ϕ_{rj} the angle to watermark, and *F* the watermark 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 $Fdyn$.

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

$Fdyn$ is the current watermark strength. $Fdyn$ shall be defined according to *F* and $embedding\_mode$.

$$if\ normal\_embedding \Rightarrow Fdyn = F * A_{norm}$$

$$if\ stuffing\_embedding \Rightarrow Fdyn = F * A_{stuff}$$

$$if\ stuffing\_replacement \Rightarrow Fdyn = F * A_{replace}$$

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

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

6.1.12 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.

6.1.13 Symbol Embedding Recommendations (Informative)

6.1.13.1 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 hearing, for every frequency in the watermarking frequency band.

Note: Acceptable results have been demonstrated by an implementation that employed on average a maximum phase deviation for symbols $F = 0.28 * \pi$.

6.1.13.2 Symbol Replacement

This clause gives an optional method to address the replacement of the stuffing symbols present in an audio stream by a watermark with respect to the subjective quality of the content.

A detector can be implemented within the embedder, in order to identify in the current audio block the sub-block index j of the stuffing symbol embedded previously. Let $r\_stuffing_j$ be its j sub-block temporal reference signal.

Once identified, the stuffing symbol can be replaced using the technique defined in clauses 6.2.3 and 6.2.4,

S is the symbol to embed on the current audio block. Let r_j be its j sub-block temporal reference signal.

Let the replacement symbol $S_{repl} = S - \alpha * S_{st}$. [2]

The j sub-block temporal signal corresponding to S_{repl} is r_repl_j defined by:

$$\forall n \; in \; [0, B-1], \; r\_repl_j\,(n) = r_j(n) - \alpha * r\_stuffing_j\,(n)$$

Each sub-block $r\_repl_j\,(n)$ 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 6.2.5, considering as the angle to watermark ϕ_{replj} , and maximum deviation constraint $Fdyn = F * A_{replace}$.

The embedder inverse frequency domain transform is then applied as specified in clause 6.2.6.

[2] $0 \leq \alpha \leq 1$, α and $A_{replace}$ are recommended to be adjusted according to implementation such that symbols resulting from embedding in normal_embedding mode and in stuffing_replacement mode have similar imperceptibility and watermark confidence.

6.2 Watermark Decoding

6.2.1 Watermark decoder process

This clause defines how the watermarked information embedded in an audio stream shall be recovered and decoded. The process is illustrated in Figure 7.

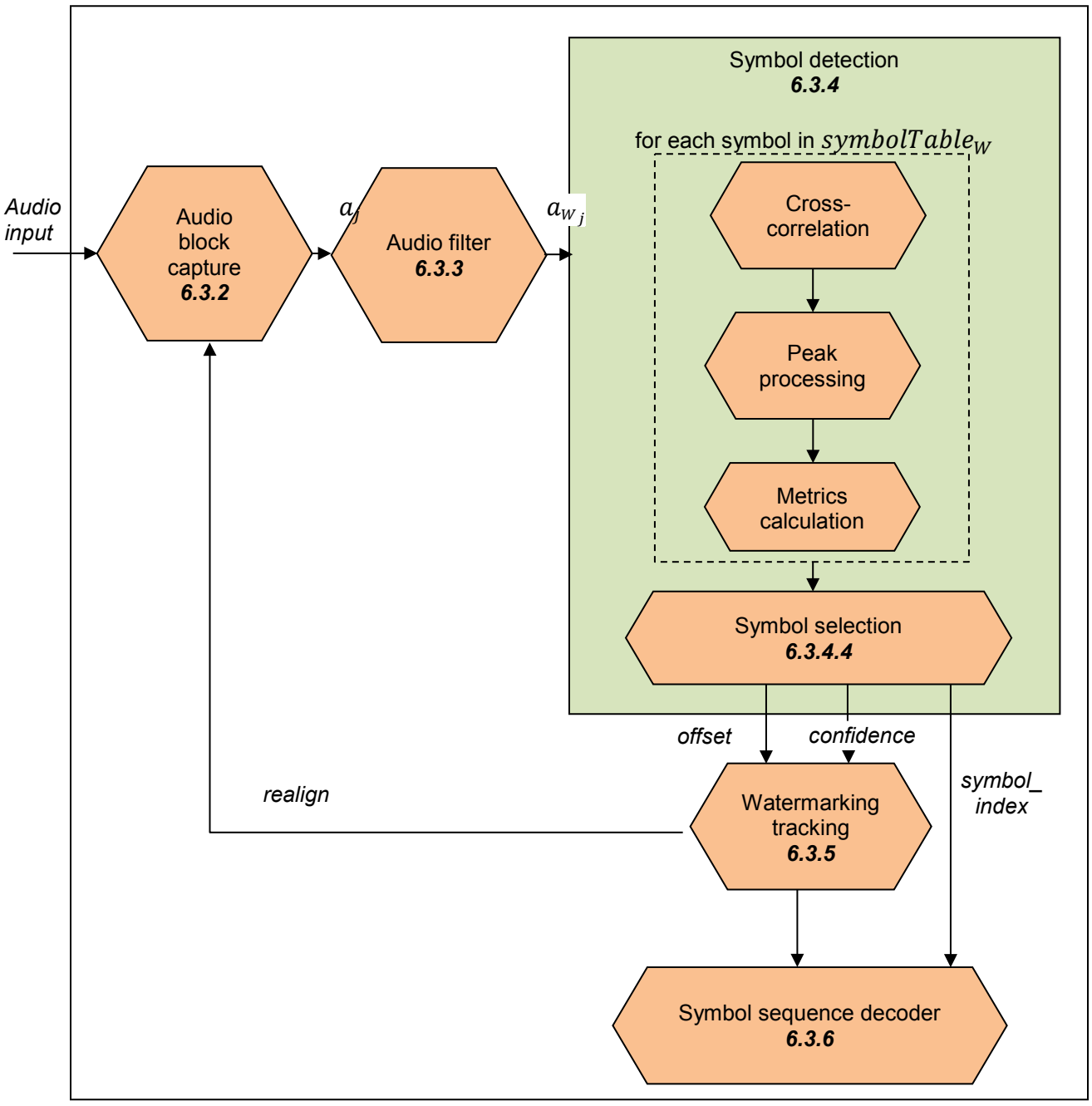

Figure 7: Watermark detector block diagram

6.2.2 Audio Block Capture

The audio block capture shall split the audio input signal into a_j blocks of *T* samples that will be processed by the symbol detector.

If a non-zero temporal realignment order value $realign$ is received from the watermark tracking, 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.

6.2.3 Audio Filtering

The audio filter processes 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 interference from the host signal.

$$a_w = WhiteningFilter(a)$$

The same whitening technique shall also be applied to the reference symbols present in $symbolTable$[3].

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 given below:

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 the temporal domain a_w using inverse discrete Fourier transformation.

6.2.4 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 should be performed:

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

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

6.2.4.1 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 $Cyclic(i + m) = (i + m + T)\%\, T$.[4]

---

[3] See clause 6.1.1 for $symbolTable$ and N_S definition

[4] % refers to the modulo operator.

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

6.2.4.2 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 the following:

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

Compute[5] $y_{s_k} = \alpha * (filter(b, a, {c_{s_k norm}}^2, zi) - \beta)$. Define M as 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 the case of direct connection there is no acoustic transmission path to compensate for, so peak processing is not useful. In such case ${c'}_{s_k}(m)$ is defined as $c_{s_k}(m)$.

6.2.4.3 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 $[-T/2, T/2-1]$.

6.2.4.4 Symbol Selection

The average standard deviation shall be defined as: $\sigma = sqrt((\sum_0^{Ns-1} {\sigma^2}_{s_k})/N_S)$.

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 $offset = offset_{s_{symbol\_index}}$.

6.2.5 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.

---

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

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 time shift observed between the watermark and the signal.

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

6.2.6 Symbol Sequence Decoder

The symbol sequence decoder shall perform the reliable detected symbols sequence aggregation, the error detection control and the watermarking information decoding.

A FIFO buffer may be used to store the incoming reliable detected symbols with their attached detection time.

When one of the four synchronization symbols is detected, the packet structure and size shall be identified as specified in clause 6.4.1.

Once a full packet has been received, the parity symbol values shall be checked as specified in clause 6.4.2 to verify the packet validity.

The watermark DCID or DCTL values can then be extracted as described in clause 6.5.

6.3 Data Link Layer Architecture

6.3.1 Packet Structure

A packet shall be defined as a series of 10 contiguous symbols sequentially embedded in the audio signal starting with a specific synchronization symbol. See Table 3.

The synchronization symbol index *sync* shall represent the watermark packet position in the watermark cell (see clauses 6.1.1 and 6.1.4 for more information).

Table 3: Syntax of a watermark packet structure

Syntax	No. of symbols
`watermark_packet() {`	
`sync`	1
`DCIDpayload`	4
`DCIDparity`	1
`DCTLpayload`	3
`DCTLparity`	1
}	

sync symbol shall be one of the four specific synchronization symbols defined by this standard. Synchronization symbol indexes and their corresponding time slot position are listed in Table 2.

DCIDpayload shall be a sequence of four data symbols, each one corresponding to nine bits of data.

DCTLpayload shall be a sequence of three data symbols, each one corresponding to nine bits of Timestamp data.

DCIDparity shall be a data symbol corresponding to nine bits of *DCIDpayload* parity check. See clause 6.4.2 for further details.

DCTLparity shall be a data symbol corresponding to nine bits of *DCTLpayload* parity check. See clause 6.4.2 for further details.

6.3.2 Parity Symbol

Both payload elements of the watermark packet (*DCIDpayload* and *DCTLpayload*) shall each be followed by a parity symbol.

6.3.2.1 *DCIDparity*

Each bit of the *DCIDparity* 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 *DCIDparity* symbol shall be used to detect *DCIDpayload* decoding errors: the decoded information shall be handled as invalid if there is at least one bit of the parity symbol that is not the inverse of the XOR sum of the corresponding decoded *DCIDpayload* bits.

6.3.2.2 *DCTLparity*

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

The error detection control is implemented as following:

During watermark decoding, the *DCTLparity* symbol shall be used to detect *DCTLpayload* decoding errors. The decoded information shall be handled as invalid if there is at least one bit of the parity symbol which is not the XOR sum of the corresponding decoded *DCTLpayload* bits.

6.4 Payload Structure

Each watermark packet shall carry a Distribution Channel ID (DCID) and its associated Timestamp (DCTL).

- The Distribution Channel ID shall be encoded using one **`DCID_value`** field of 32 bits.
    - The Distribution Channel ID value used shall be the suffix portion of the EIDR Video Service ID. Video Service IDs are a particular class of Digital Object Identifier – DOI, as specified in ISO 26324, and administered by EIDR).
    - An example of an EIDR Video Service ID could look like this:
        - 10.5239/B25B-5C67
            - The 10.5239/ prefix identifies this within the DOI space as an EIDR Video Service ID
            - The B25B-5C67 suffix identifies the particular Video Service as a particular Network.
    - As the only type of DCID used in the context of this document is the EIDR Video Service ID, the prefix is not needed.
    - The hyphen between the fourth and fifth hex digits is discarded, yielding an 8-digit hexadecimal number, example 0xB25B5C67.
    - Converting the hex number to binary yields a 32-bit unsigned integer, MSB first, example 10110010010110110101110001100111.
    - This binary number is embedded in reverse order (least significant bit first). See Table 4.

- The Timestamp T_C to be embedded is the time relative to yyyy-01-01T00:00:00.00 (UTC), where yyyy is the current year, corresponding to the last audio sample of the watermark packet to embed. Due to differences between UTC and the local broadcast day, the relative time may not be updated immediately following the last instant of the old year nnnn-12-31T23:59:59.99. Instead, the time may continue to be expressed relative to the old year until the end of the broadcast day, at which point the time shall be

expressed relative to the new year. Receiving devices shall be designed to correctly interpret either value.

- The year shall not be updated before yyyy-01-01T00:00:00.00, because the timestamp cannot represent negative numbers. Consequently, distribution channels that originate in regions East of the Prime Meridian and West of the International Date Line might not be able to reference the local current year until the January 2 broadcast day.

Note: The 32-bit timestamp has a range of well over 600 days, so it is possible to represent timestamp values for January 1 and January 2 relative to either the current year or the previous year. A broadcast day can be greater or less than 24 hours long and it does not necessarily begin or end at midnight local time.

The Timestamp is encoded using two fields:

- one **`DCTL_timecode`** field of 21 bits which represents the media timeline quantized in a number of watermark cells.
- one **`DCTL_accuracy`** field of 10 bits that represents the offset to add to get precise Timestamp value.
  This offset is given in $1/2^{10}$ of a watermark cell duration: 1310720 / 1024 = 1280 samples.

The Timestamp T_C to be embedded shall be encoded as:

$T_{encod} = \text{DCTL\_timecode} * 40 * \text{T} + \text{DCTL\_accuracy} * 1280$, with $\text{DCTL\_timecode} \geq 0$ and $\text{DCTL\_accuracy} \geq 0$. [6]

DCTL_accuracy shall furthermore be chosen such that it minimizes the absolute value distance between T_C and T_{encod}. As a consequence, the difference between the timestamp to embed and the encoded timestamp value will always be lower than or equal to 640 samples.

Note: The difference between any timestamp to encode and its encoded timestamp value will consequently be in average 320 samples (i.e. 6.7 milliseconds).

The **`DCID_value`** field is transmitted in the *DCIDpayload* symbol sequence

The **`DCTL_timecode`** field is transmitted in the *DCTLpayload* symbol sequence

The **`DCTL_accuracy`** field is split in two parts: the four MSB are transmitted in the *DCIDpayload* symbol sequence while the six LSB are transmitted in the *DCTLpayload* symbol sequence.

Define the following notations:

$\text{DCTL\_accuracy} = \sum_{k=0}^{9} \text{DCTL\_accuracy}_k * 2^k$,

$\text{DCID\_value} = \sum_{k=0}^{31} \text{DCID\_value}_k * 2^k$,

$\text{DCTL\_timecode} = \sum_{k=0}^{20} \text{DCTL\_timecode}_k * 2^k$,

$\text{DCID\_payload} = \sum_{k=0}^{35} \text{DCID\_payload}_k * 2^k$,

$\text{DCTL\_payload} = \sum_{k=0}^{26} \text{DCTL\_payload}_k * 2^k$.

---

[6] Example: The last audio sample of the watermark packet to embed corresponds to the time 15th January 15h 20 minutes 6 seconds 857 milliseconds, i.e. the time elapsed since the beginning of the year is 1264806.857 seconds.

The encoded timestamp shall be DCTL_timecode * 40 * T + DCTL_accuracy * 1280, with DCTL_timecode = 46318 and DCTL_accuracy = 625. The difference between the encoded timestamp and the timestamp to embed is 3.7 milliseconds.

The syntax of the DCID payload shall be as shown in Table 4.

Table 4: Syntax of DCID payload structure

Syntax	No. of bits	Format	

DCID_payload() {
DCTL_accuracy_msb
DCID_value
}
	
	4
	32
		
		Unsigned integer MSB first
		Unsigned integer LSB first
			$\forall$ i in $[0,3]$, $\text{DCID\_payload}_i = \text{DCTL\_accuracy}_{9-i}$
			$\forall$ i in $[4,35]$, $\text{DCID\_payload}_i = \text{DCID\_value}_{i-4}$

The syntax of the DCTL payload shall be as shown in Table 5.

Table 5: Syntax of DCTL payload structure

Syntax	No. of bits	Format	

DCTL_payload() {
DCTL_accuracy_lsb
DCTL_timecode
}
	
	6
	21
		
		Unsigned integer LSB first
		Unsigned integer LSB first
			$\forall$ i in $[0,5]$, $\text{DCTL\_payload}_i = \text{DCTL\_accuracy}_i$
			$\forall$ i in $[6,26]$, $\text{DCTL\_payload}_i = \text{DCTL\_timecode}_{i-6}$

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 symbol shall also be encoded LSB first.