|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  36th Meeting: Gothenburg, SE, 6–12 July 2019 | Document: JCTVC-AJ0024 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **NALU verification SEI message** | | |
| *Status:* | Input document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Pierre Andrivon  Edouard François | Email: | <pierre.andrivon@interdigital.com> <edouard.francois@interdigital.com> |
| *Source:* | InterDigital | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

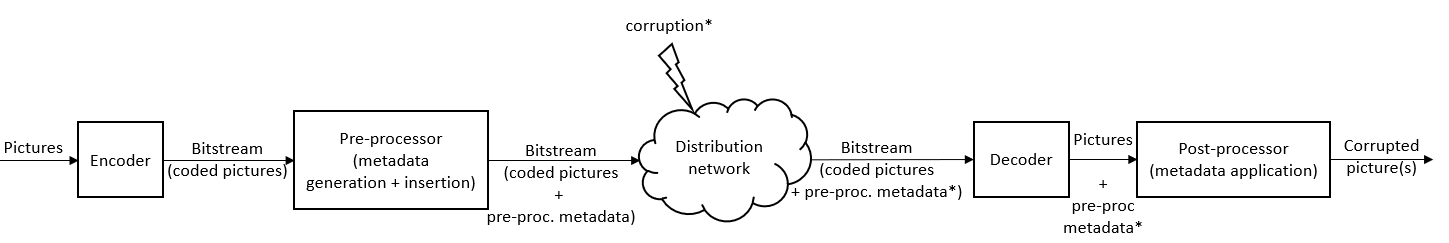
# Abstract

This contribution relates to error resilience and error concealment at the elementary stream level. It is proposed to carry in an SEI message a hash value characterizing (a group of) NALU so that corruption of the (group of) NALU may be detected prior to decoding of said (group of) NALU. It is asserted that this proposition allows verifying the integrity of a (group of) NALU or the entirety of the coded stream so that appropriate error-concealment or error resilience measures can be triggered when a corruption is detected.

# Introduction

Implementation of error-resilience strategies in video coding is either at higher level than elementary stream level (channel coding) or part of the coding tool design (source coding). It is generally observed that transmission errors are not pulled up to higher layers (for example to application layer). As an example, in conventional digital television rendering, it happens that macroblock/CU corruptions or corrupted rows of pictures are displayed on TV sets before the TV set indicates an issue related to the signal (for example signal is too low or too noisy to be decoded correctly).

It is difficult to determine whether a bitstream is corrupted at the elementary stream level without parsing and trying to decode the bitstream itself. A robust decoding implies a lot of low-level checks and countermeasures in the decoder.



**Figure 1: Example of bitstream corruption applied to metadata**

Besides, in case a video coded stream is accompanied with image enhancement parameters carried in an SEI message (e.g. Mastering Display Colour Volume, Colour Remapping Information, Knee Function information, Tone Mapping information SEI messages), it may be helpful to know whether the coded stream and/or the relevant processing metadata are corrupted or not. Indeed, if the decoded stream is corrupted, or carried processing metadata are corrupted, the output post-processed picture may have an undesired look.

Even in case processing metadata are corrupted but associated syntax elements remain in the definition range, the post-processor (applying metadata on the decoded picture) is unable to determine that the metadata values are not the intended ones and the decoded picture will be reconstructed with wrong metadata values begetting an unintended look to the picture which can dramatically hinder viewer's experience.



Figure 2: Left is picture intended look - right is picture whose metadata have been corrupted but still in definition range

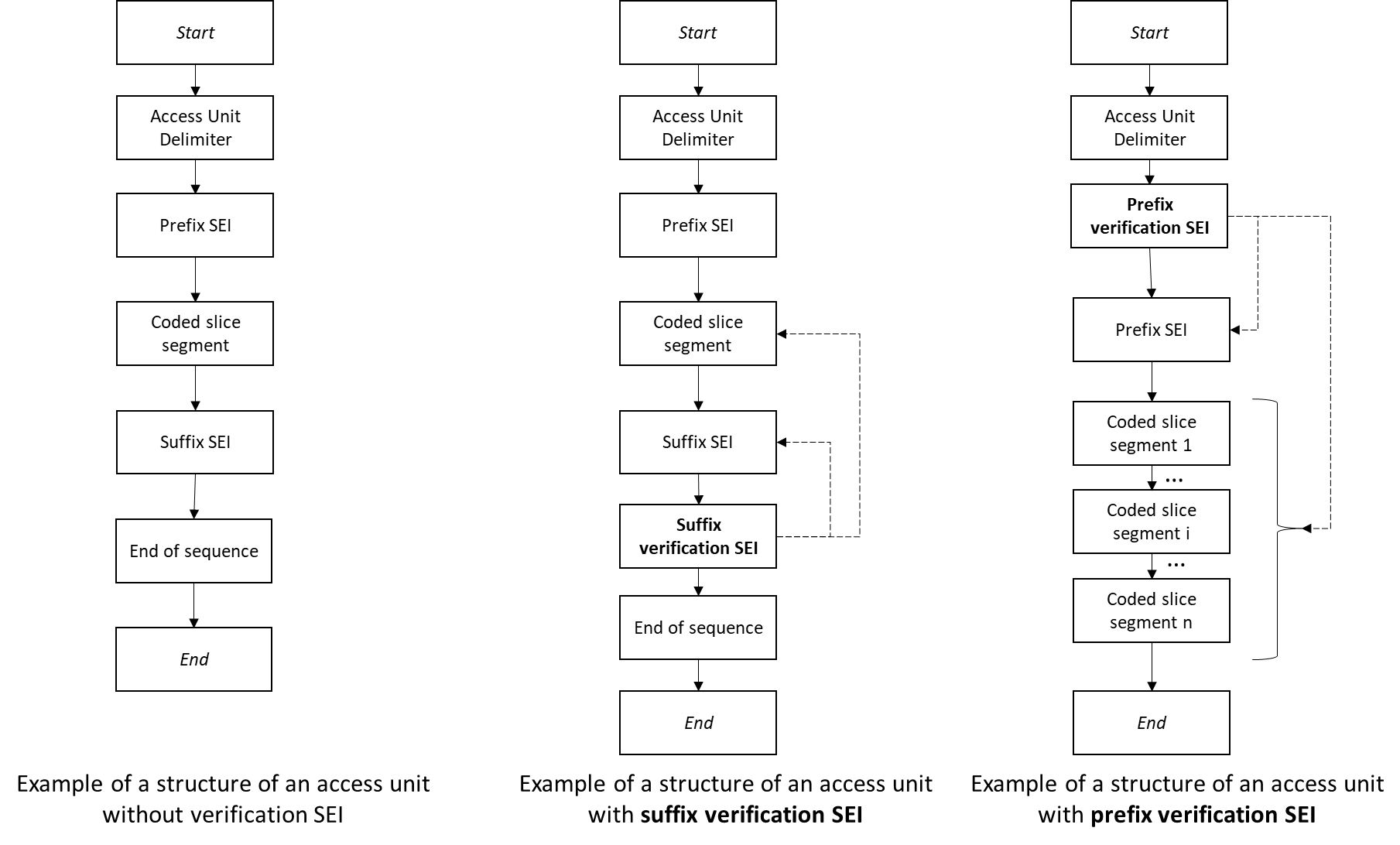
One may think of using the decoded picture hash SEI message to detect whether the reconstructed picture is corrupted or not. However this requires a full decoding of the picture and according to the corruption severity the decoding may crash the decoder, or it may impact visually several pictures (e.g. all along the GOP of the incriminated Access Unit) and thus are not acceptable.

# Proposition

It is proposed to signal the hash of coded video bitstream highest-level structures (e.g. NALU or AU) in the elementary stream. Typically, a hash characterizing a (set of) NALU and/or an AU is carried aside in an SEI message (a so-called verification SEI message). Thus, the integrity of a portion or the entirety of the coded stream can be verified and may trigger appropriate error-concealment or error resilience measures when a corruption is detected before decoding stage.

It is proposed to specify the verification SEI message as a prefix SEI message. This has the advantage to provide a hash value prior to receiving the NALU which correspond to the data characterized by the hash value.

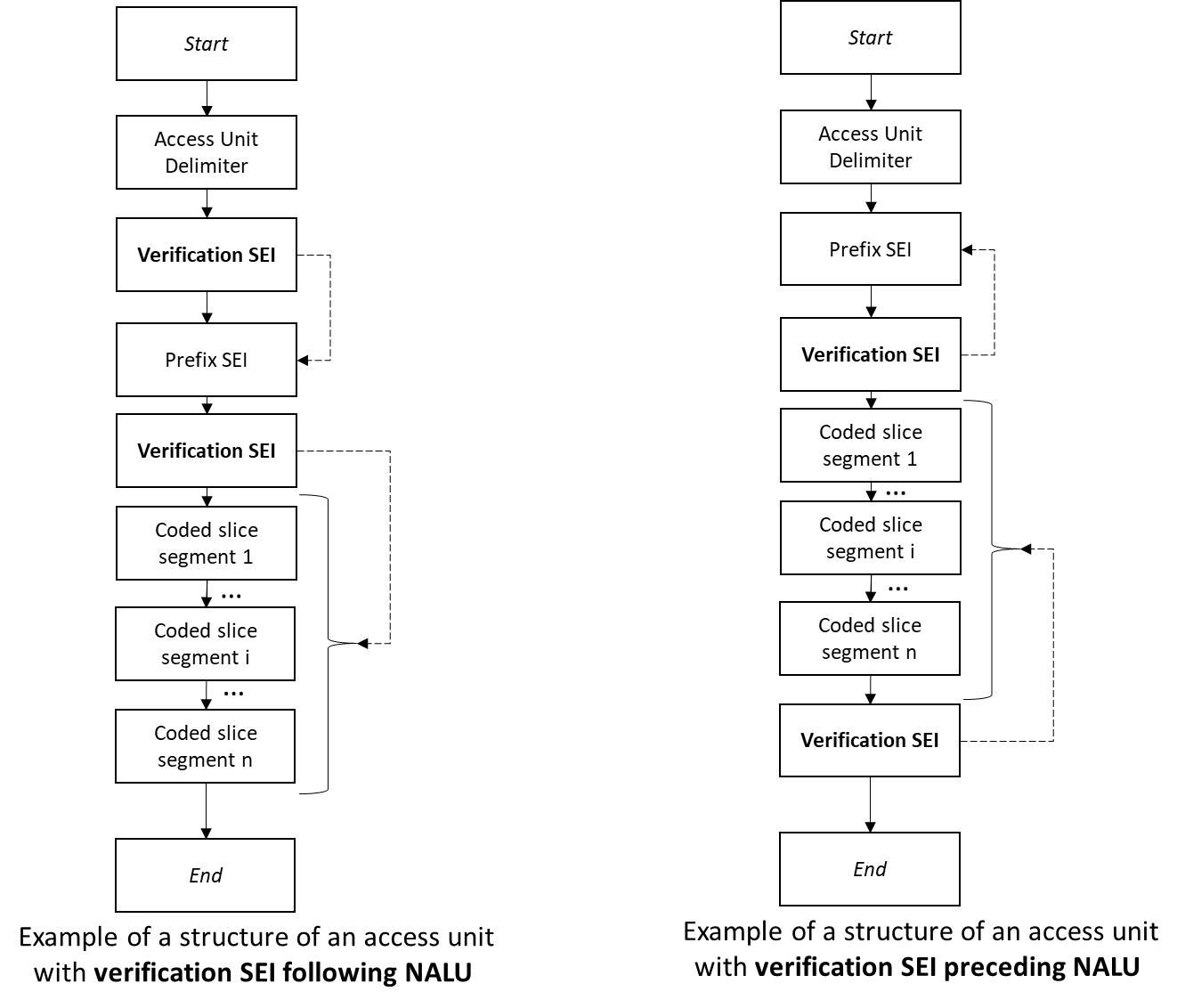
In the figure below, the figure at the left shows a conventional access unit without any verification SEI message, the figure in the middle shows a suffix version of the verification SEI message which encompasses a hash of either a suffix SEI message, or a coded slice segment NALU (or a whole Access Unit). Right figure shows a verification SEI message as the first prefix SEI message and may characterize another prefix SEI message, a group of coded slice segment NALU (or a whole access unit).



**Figure 3: Example of structures of Access Unit without (left), with suffix (middle) or with prefix (right) verification SEI**

The verification SEI message can include a signalling of the type of NALU it relates to.

Alternatively, it may be implicitly determined according to the position of the verification SEI message in the Access Unit as an example, the verification SEI message directly precedes the NALU that it characterizes as shown in the following figure.



**Figure 4: Example of structures of Access Unit with verification SEI characterizing following (left) or preceding (right) NALU.**

An exemplary syntax of the proposed verification SEI message is provided below:

| **Syntax** | **Descriptor** |
| --- | --- |
| coded\_data\_hash( payloadSize ) { |  |
| **cdh\_type** | u(8) |
| **cdh\_nalu\_type** | u(8) |
| if( cdh\_type == 0 ) |  |
| for(i = 0 ; i < 16 ; i++ ) |  |
| **cdh\_md5**[ i ] | b(8) |
| if( cdh\_type == 1 ) |  |
| **cdh\_crc** | u(16) |
| } |  |

Figure 5 Example of syntax for the verification SEI message

where *cdh\_type* indicates the method used to calculate the verification SEI value, i.e. the checksum, according to the following table:

| **cdh\_type** | **Method** |
| --- | --- |
| 0 | MD5 (IETF TFC 1321) |
| 1 | CRC |
| 2 .. 255 | *Reserved values* |

and *cdh\_nalu\_type* indicates type of NALU characterized by the checksum according to the following table:

| **cdh\_nalu\_type** | **NALU type** |
| --- | --- |
| 0 | All NALU in the AU (precluding verification SEI message) |
| 1 | All VCL NALU of the current AU |
| 2 | All non-VCL NALU of the current AU |
| 3 | Following SEI message |

*cdh\_nalu\_type* may be removed in case the verification SEI message implicitly characterizes for instance the NALU it precedes or follows.

*cdh\_md5* and *cdh\_crc* have respectively the same semantics as *picture\_md5* and *picture\_crc* syntax elements of the decoded picture hash SEI message except that it applies to a NALU (or set of NALU) such as implicitly determined or explicitly determined in *cdh\_nalu\_type*.

# Patent rights declaration(s)

**InterDigital may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**