Encryption for High Efficiency Video Coding with Video Adaptation Capabilities

Glenn Van Wallendael, Andras Boho, Jan De Cock, Adrian Munteanu, and Rik Van de Walle

In: IEEE Transactions on Consumer Electronics, 59 (3), 634-642, Aug. 2013.

Encryption for High Efficiency Video Coding with Video Adaptation Capabilities

Glenn Van Wallendael, *Student Member*, IEEE, Andras Boho, Jan De Cock, *Member*, IEEE, Adrian Munteanu, *Member*, IEEE, Rik Van de Walle, *Member*, IEEE

Abstract — *Video encryption techniques enable applications like digital rights management and video scrambling. Applying encryption on the entire video stream can be computationally costly and prevents advanced video modifications by an untrusted middlebox in the network, like splicing, quality monitoring, watermarking, and transcoding. Therefore, encryption techniques are proposed which influence a small amount of the video stream while keeping the video compliant with its compression standard, High Efficiency Video Coding. Encryption while guaranteeing standard compliance can cause degraded compression efficiency, so depending on their bitrate impact, a selection of encrypted syntax elements should be made. Each element also impacts the quality for untrusted decoders differently, so this aspect should also be considered. In this paper, multiple techniques for partial video encryption are investigated, most of them having a low impact on rate-distortion performance and having a broad range in scrambling performance*[1].

Index Terms — High Efficiency Video Coding, encryption, transcoding, video scrambling

I. Introduction

Applications using video encryption can range from pay TV up to confidential military applications. When confidentiality is of the highest priority, encryption of the entire video stream is advised. In a scenario of television distribution, some disadvantages can be associated with full video encryption.

First, useful technical information, irrelevant for end users, becomes obscured. For example, a video stream contains information about the used compression standard, the used profile, and picture width and height. Leaving this information unencrypted can give the end user device an indication about the decodability of the video stream or the resolution of this premium service.

Second, encrypting the entire video stream removes its format compliance with the video standard. Because of encryption, the data gets randomized, resulting in unexpected behavior of the decoding equipment. With a format compliant encryption solution, there is certainty about the expected behavior of all intermediate and end user devices in the video chain. The downside of having to guarantee standard compliance is that by encrypting certain syntax elements, the statistics of the video stream change, resulting in a higher bitrate for the same video. Therefore, attention should be paid to which video stream elements get encrypted.

Third, with full stream encryption, adaptation devices in the network need to be trusted with the decryption key in order to make changes to the video stream. With only a small amount of encrypted elements in the video stream, adaptation devices are free to modify other elements in the video stream. In this way, without knowing the encryption key, modifications [1] like compressed domain watermarking [2] and transcoding [3] are possible.

Finally, when only encrypting certain syntax elements in the video stream, the scrambling intensity of the video stream can be configured. With full stream encryption, no visual information is revealed about the video stream. Sometimes it can be beneficial to offer low quality preview functionality in order to convince the viewer. Different syntax elements will impact the decoded quality of an untrusted device differently and therefore the elements to encrypt should be chosen depending on the quality offered to untrusted devices.

In this paper, encryption of different elements from the video stream is investigated. As a video compression algorithm to work with, the recently standardized High Efficiency Video Coding (HEVC) standard [4] is used. How such a video stream is structured and which syntax elements can be used for encryption is explained in Section II. How adaptation and encryption worked on older video compression standards like H.264/AVC and Scalable Video Coding (SVC) is described in Section III. Then, in Section IV, the proposed encryption algorithm working on HEVC will be explained. Finally, Section V and Section VI will show the broad range in scrambling performance and the conclusion, respectively.

[1] The research activities that have been described in this paper were funded by Ghent University, iMinds, Ph.D. and post-doctoral fellow grants of the Agency for Innovation by Science and Technology (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union. Furthermore, this work was carried out using the Stevin Supercomputer Infrastructure at Ghent University.

Glenn Van Wallendael is with the Department of Electronics and Information Systems – Multimedia Lab, Ghent University – iMinds, Ghent, Belgium (e-mail: glenn.vanwallendael@ugent.be)

Andras Boho is with the Department of Electrical Engineering, KU Leuven, Leuven, Belgium (e-mail: andras.boho@esat.kuleuven.be)

Jan De Cock is with the Department of Electronics and Information Systems – Multimedia Lab, Ghent University – iMinds, Ghent, Belgium (e-mail: jan.decock@ugent.be)

Adrian Munteanu is with the Department of Electronics and Informatics, Vrije Universiteit Brussel – iMinds, Brussels, Belgium (e-mail: acmuntea@etro.vub.ac.be).

Rik Van de Walle is with the Department of Electronics and Information Systems – Multimedia Lab, Ghent University – iMinds, Ghent, Belgium (e-mail: rik.vandewalle@ugent.be)

II. HIGH EFFICIENCY VIDEO CODING

In January 2013, a joint collaborative team between the Moving Picture Experts Group (MPEG) and the Video Coding Experts Group (VCEG) finished the standardization of HEVC. This standard is the successor of the dominantly present H.264/AVC [5] compression standard and it outperforms the former with 50% in bitrate reduction at similar subjective quality [6]. Although HEVC is still a hybrid block based video compression standard as H.264/AVC, there are some fundamental differences explaining this performance leap.

When looking at a video stream from a high level of abstraction, first, the video stream is divided in pictures. With inter predicted pictures, prediction from previously decoded pictures can happen, i.e. reference pictures. The set of reference pictures is signaled as a table in the short-term Reference Picture Set (RPS). During the actual inter prediction, as will be explained later, only an index in this table needs to be sent in the video stream in order to predict from a certain picture.

Every picture is further divided in slices. These slices form independently decodable parts of the video stream. Each slice gets wrapped in a Network Abstraction Layer (NAL) unit containing a NAL header and the actual slice information. In the NAL header, high level information about the picture type can be found. This picture type can be used to distinguish the random access pictures from inter predicted pictures.

A slice consists of several Coding Tree Units (CTU). For HEVC, a rate-distortion-complexity trade off indicated a CTU size of 64x64 as a good compromise. Smaller sizes down to 16x16 can be chosen if preferred. As a comparison with H.264/AVC, the corresponding structure was called MacroBlocks (MB) and their size was fixed to 16x16 pixels.

The CTU is further divided in a quad tree of Coding Units (CU). CUs can be partitioned in sizes ranging from 64x64 down to 8x8. On the CU level, information about the block type is signaled (inter or intra). Additionally, for rate-control purposes, a change in quality and bitrate can be controlled up to this level, as will be explained later. The CU forms the root for further partitioning in Prediction Units (PU) and Transform Units (TU).

With an intra-coded CU, the CU can be further split in PU partitions of size 2Nx2N which represents no split and NxN which represents a split in four sub partitions. Partitioning inter-coded CUs can happen with more flexibility such that rectangular and asymmetrical splits can occur as well. For each of these inter-coded PU partitions, motion information is signaled in the video stream. The motion can be predicted from the motion in neighboring blocks. This merging of motion information starts with the creation of a list of merge candidates. From these candidates, one is selected and signaled in the video stream with a merge index (idx). When no merge occurs, all the motion information needs to be explicitly mentioned in the video stream. If the PU is unidirectional predicted, one Motion Vector (MV) and corresponding information is included. With bidirectional prediction, two sets of this information are signaled. First, because previously decoded pictures can be used to predict from, an index indicating the reference picture is signaled, i.e. reference picture index. Second, the MV is constructed by a MV Prediction (MVP) process. In this process, a candidate list of MVs is created and an index is included indicating the used MV predictor. This index will be further named MVP index. Now that a motion vector prediction is obtained, the difference with the motion vector, called the MV Difference (MVD), still needs to be signaled in the video stream. For this purpose, a MVD sign and MVD size are put in the video stream.

As with the PUs, again, a quad tree partitioning with the CU as a root can be performed obtaining TUs. The pixels corresponding to the TU size are transformed and quantized. The Quantization Parameter (QP) controlling the eventual quality and bitrate can be set and adjusted on different levels in the video stream, namely on picture level, on slice level, and on CU level. This is mainly used for matching the bitrate to a certain value in rate-control applications.

Within the TU, first the position of the last significant coefficient and the significance of the different coefficients is signaled. Then, the absolute level of the coefficients and their sign information are put in the video stream. So, with the positions of significant coefficients, their sign, and their size, a reconstruction of the residual information can be made. In HEVC, there is an additional tool giving small gains called Sign Data Hiding (SDH). With this tool enabled and under certain conditions, the sign of the first significant coefficient is inferred from the parity of the sum all the coefficients.

After reconstruction in an HEVC decoder, some final in loop filtering stages are processed, namely the deblocking filter and the Sample Adaptive Offset (SAO) filter. For these filters, also some signaling takes place in the video stream. More specifically, parameters controlling the strength of the deblocking filter and parameters to enable the SAO filter need to be communicated.

All the different elements like the partitioning structure, the motion information, and the residual information need to get in binary form when put in the video stream. For this purpose, HEVC uses a binarization stage followed by Context-Adaptive Binary Arithmetic Coding (CABAC). This CABAC engine learns about the statistics of a certain video element and throughout the slice spends less bits on frequently occurring values. With this CABAC entropy coder, some final redundancies are removed from the bit stream. From some elements, it was observed that they occur too random to spend CABAC processing on, so they get bypass encoded. This means that their binary form is put in the bit stream without further compression. In HEVC, bypass encoding is used for the MV sign, the residual sign, and SAO signaling.

III. Adaptation on Encrypted Video

Considering all different steps in the HEVC coding process, there is still a lot of resemblance with H.264/AVC. Currently, there is not a lot of research describing encryption and adaptation for HEVC. Therefore, in this section, an overview is given of frequently occurring adaptation operations and their requirements related to encryption.

As a first application, splicing can be considered. Splicing is the process in which a fragment from a different video stream is inserted in a video stream. Splicing is mainly used for advertisement insertion or other editing operations. For a splicing operation to work, there needs to be access to information related to random access in the video stream. This information can be found in the NAL header, so encryption of the NAL header should be avoided to enable this type of modification on the video stream.

Second, no-reference quality measurement or monitoring could be applied in the network. With quality monitoring, the influence of encoding parameters and packet loss on the visual quality of the video stream is measured. When this quality drops below a certain level, depending on the problem, more robust encoding schemes or more efficient compression could be applied in order to restore the quality again. Depending on the quality monitoring algorithm, different syntax elements are used. Some consider the boundary strengths of the deblocking filter and the QP [7] whereas others need motion information bit allocation information [8].

Third, compressed domain watermarking could be applied in the network for later identification of the source or the destination of the video stream. As for quality monitoring, also with watermarking, different algorithms exist. As an example, some algorithms insert the watermark in the absolute levels of the residual information at the TU level in the video stream [1]. Leaving the residual size unencrypted is important when this application is considered in the network.

Finally, there is the application of compressed domain transcoding [9] or more specifically, transrating. This form of transcoding slightly reduces the bit rate of the video stream without a full decoding and encoding loop, making it low complex and therefore suitable for in the network. With transrating, typically the residual information is requantized at a coarser QP while leaving the quadtree and prediction information unchanged [10]. Consequently, for this application to work, QP and residual information needs to remain unencrypted in the video stream.

For H.264/AVC, a lot of encryption strategies have been investigated. Every proposed algorithm uses its own selection of syntax elements mainly selected from the following list: transformation matrices [11], intra prediction modes, residual coefficients, MVD [12], and MVD signs [13]. Other algorithms see the benefit of encrypting bypass encoded elements [14] because changing the values of these elements has no impact on the bit rate.

For Scalable Video Coding (SVC) [15], the scalable extension of H.264/AVC, numerous encryption applications [16] are investigated as well. With SVC, a low quality base layer and high quality enhancement layer are packed together in a single video stream. Video scalability enables a low complex device to extract the low bit rate and low quality base layer from the video stream. By only encrypting this high quality enhancement layer, low quality preview functionality can be offered [17]. An overview of encryption techniques for H.264/AVC and SVC can be found in [18].

IV. Encryption for HEVC

To keep format compliance after encryption, it is important that encrypted syntax elements do not change the parsing behavior of the decoder. For example, when encryption changes the CU block type from intra to inter, motion information is expected in the video stream. This mismatch will then most likely introduce incompatibility. Therefore, HEVC syntax elements which can be changed without influencing the decoding process are identified (see Table I). From this set of independently parsable syntax elements, a selection is considered for encryption, as will be described later.

TABLE I
Independently Parsable Syntax Elements in HEVC

- Short-term reference picture set (RPS)
- QP information (initial QP, chroma delta QP, slice delta QP, CU delta QP)
- Inter information (reference picture indices, motion vector prediction indices, motion merge indices, motion vector differences)
- Residual information
- Deblocking filter parameters
- Sample adaptive offset parameters

Every selected syntax element is encrypted using the Advanced Encryption Standard (AES) [19]. This is a symmetric encryption algorithm, meaning that identical keys are used for both encryption and decryption. From a high abstraction level, AES can be viewed as a black box producing a sequence of pseudo random bits depending on the secret key. By transforming the input HEVC syntax element with these pseudo random bits, the encrypted syntax element is obtained. The applied transformation depends on the type of syntax element.

The first syntax element from Table I that is considered in this paper is the RPS. The list of reference pictures in the RPS is mixed using the bits produced by the AES algorithm. To transform this list, position swapping is applied. When a *one* bit is produced by the AES algorithm, the reference picture is swapped with the next reference picture in the list. This swapping operation is applied on every reference picture in the list. When applying this technique, it should be considered that the RPS is included in the slice header so there is only one RPS for every slice. Given that there is only a small amount of information encrypted, the low security aspect of only encrypting this element should be kept in mind.

Next, QP information is encrypted on the CU level. On this

level, a delta with the previously decoded QP is signaled. By pseudo randomly changing this delta QP over a wider range, the encryption strength can be controlled. Having larger variations implies a larger negative impact on compression efficiency, but a higher scrambling performance. Therefore, strengths varying from two up to eight are investigated. This strength indicates how much variation is put on delta QP. It specifies the range of values that the syntax element can be mapped to. For example, with a strength of four, a delta QP ranging from zero to three can be mapped on values in the same range depending on the pseudo random values generated by the AES algorithm.

On the PU level in the video stream, the motion information can be encrypted. When the PU is merged with a neighboring PU, there is a merge index indicating with which neighbor to merge. This merge index can be encrypted by changing it into another valid index. This is done by the modular addition of a random number, generated by the AES algorithm, to the merge index.

When MVs are signaled in the PU, first the reference picture index is signaled. Encrypting the reference picture index results in a motion compensation operation using a different reference picture. Similar to encryption of the merge index, a modular addition with the pseudo random number is made in order to encrypt the reference picture index. The idea of encrypting this syntax element is similar to encrypting the RPS. With both techniques the reference picture for motion compensation is changed. The difference is that this time encryption happens at a higher granularity. More data gets encrypted, so the security against brute force attacks is increased.

After choosing the reference picture, a motion vector predictor is signaled. Similar to both previous indices, also the MVP index gets encrypted by a modular addition with the pseudo random code.

The only remaining elements specifying the motion of a PU are the sign and the size of the motion vector difference. The MVD sign is represented by one bit, so AES will provide a pseudo random bit to perform an eXclusive OR (XOR) operation on. The advantage of encrypting the MVD sign is that it is bypass encoded in the CABAC engine. Consequently, encrypting this element will have no impact on the compression performance of the video stream [20]. For the size of the motion vector, there is again the tradeoff between scrambling performance and impact on compression efficiency. Therefore, as for delta QP, strengths varying between two and eight are investigated.

After the PU data, the video stream contains TU information. Within a TU, the residual sign and residual size can be identified as independent information. Similar to the MVD sign, the residual sign is represented as a bypass encoded bit as well. The 1-bit XOR operation is performed on this element in order to encrypt it. Again, the bypass property enables encryption of this element without impact on the compression efficiency [20]. When encrypting the sign information, special care must be taken when the sign hiding tool called SDH is enabled in the video stream. Changing the signs can change the parity resulting in a change of the residual size. For middlebox applications modifying the residual size it is advised to also avoid encryption of the residual sign when SDH is enabled. Encryption of the residual size can happen conforming to the encryption of delta QP and MVD size. Also for residual size the tradeoff caused by varying strengths between two and eight can be made.

As a final step in the decoding process, the in-loop filters are applied on the decoded picture. The downside of encrypting parameters of these filters is that they are only signaled once for every slice, similar to the RPS. Consequently, a low amount of information is encrypted, resulting in a security risk. As an additional risk, disabling these filters when they originally should be applied will have a low impact on the quality. To evaluate the impact of encrypting filter information, the SAO type is scrambled. This is a bypass coded syntax element, so there will be no impact on compression efficiency.

V. RESULTS

A. Methodology

To measure the scrambling performance and the bit rate impact of encrypting the different syntax elements of HEVC, first a set of sequences is encoded without encryption, called the original video streams. Under the same conditions, this set is compressed with encryption of a certain element enabled. These video streams are the encrypted video streams. Then, the encrypted streams are also decoded by an untrusted decoder which does not have the decryption key, generating the untrusted video streams. Consequently, the bit rate impact of enabling encryption can be measured between the original and the encrypted video streams. The scrambling performance is measured between the untrusted and the encrypted video streams.

All the video streams are generated with a modified version of the HEVC reference Model (HM) v10.0 [21]. Modifications are applied to enable AES encryption on all the listed syntax elements. HM v10.0 is the first reference software version conforming to the finalized standard in January 2013. Although the encryption process is included in the HEVC encoder, it is written independently from the encoding process. Therefore, results obtained from these measurements are also applicable on encryption algorithms performing the encryption process after encoding, or even in the network. To measure the impact of the proposed encryption methods, all of these methods are tested on a set of sequences as listed in Table II. This set contains 25 test sequences (8-bit) ranging in resolution from 416x240 up to 2560x1600. The content varies between synthetically generated content (ChinaSpeed) and difficult to compress natural content (PeopleOnStreet).

As encoding parameters, a Group Of Picture (GOP) size of eight is chosen. This implies that seven bidirectionally

TABLE II
EVALUATED TEST SEQUENCES

Sequence	Frame count	Frame rate	Resolution
Traffic	300	30fps	2560x1600
PeopleOnStreet	150	30fps	2560x1600
Kimono	240	24fps	1920x1080
ParkScene	240	24fps	1920x1080
Cactus	500	50fps	1920x1080
BasketballDrive	500	50fps	1920x1080
BQTerrace	600	60fps	1920x1080
FourPeople	600	60fps	1280x720
Johnny	600	60fps	1280x720
KristenAndSara	600	60fps	1280x720
Vidyo1	600	60fps	1280x720
Vidyo3	600	60fps	1280x720
Vidyo4	600	60fps	1280x720
ChinaSpeed	500	30fps	1280x720
SlideEditing	300	30fps	1280x720
SlideShow	500	20fps	1280x720
BasketballDrill	500	50fps	832x480
BQMall	600	60fps	832x480
PartyScene	500	50fps	832x480
RaceHorses	300	30fps	832x480
BasketDrllTxt	500	50fps	832x480
BasketballPass	500	50fps	416x240
BQSquare	600	60fps	416x240
BlowingBubbles	500	50fps	416x240
RaceHorses	300	30fps	416x240

TABLE III
RESOLUTION DEPENDENT RATE POINTS FOR CONSTANT BITRATE ENCODING

Sequence	Rate 1 [kbps]	Rate 2 [kbps]	Rate 3 [kbps]	Rate 4 [kbps]
2560x1600	30000	20000	10000	5000
1920x1080	20000	10000	5000	2500
1280x720	10000	5000	2500	1000
832x480	10000	5000	2500	1000
416x240	5000	2500	1000	500

predicted pictures are inserted between every unidirectionally predicted picture. A hierarchical-B structure is chosen for its good compression performance. To create a realistic broadcasting scenario, a random access period of one second is configured. These random access pictures are encoded as open-GOP intra pictures. To evaluate the encryption performance over a realistic quality range, QP values of 22, 27, 32, and 37 are defined. For the scenario where delta QP syntax elements are encrypted, constant bitrate compression is applied. With constant bitrate compression, a rate control algorithm changes the delta QP values for every CU to obtain a nearly constant bit distribution throughout the video stream. The resolution dependent rate points at which the video streams are encoded can be found in Table III.

To compare the bitrate impact between the original and the encrypted video streams, the Bjøntegaard delta (BD) bitrate [22] is used. For this metric, first, rate-distortion curves are plotted on a graph. Rate is expressed in bits per second and distortion is expressed with the Peak Signal-to-Noise Ratio (PSNR) between both decoded video streams. By taking the average bitrate difference at the same quality, a general number indicating the bitrate increase can be calculated. The BD-rate increase therefore expresses the average bitrate increase over the evaluated quality range.

To measure the scrambling performance, the PSNR and Structural SIMilarity (SSIM) measures are used. More specifically, the delta PSNR and delta SSIM between the encrypted and the untrusted video stream is calculated. This delta indicates how much quality is lost for an untrusted decoder compared to the quality of a trusted decoder. First, in subsection B, the quality impact after encryption of each of the described syntax elements will be discussed followed by the measured impact on compression efficiency (see subsection C).

The impact of enabling encryption and decryption in the encoder and decoder is negligible compared to encoder and decoder time. Therefore, no results are given on time measurements of enabling encryption.

B. Scrambling Performance

An overall comparison of the scrambling performance of all the described syntax elements can be found in Fig. 1 and Fig. 2. In the delta PSNR curves of Fig. 1, there seems to be a lower impact on higher QP values. It should be noted that higher QP values have lower PSNRs to start with. When expressed as a percentage, the delta PSNR would be more constant over the QP range, comparable to the SSIM measurements (see Fig. 2).

The syntax elements with the least impact (-0.30 dB) on the decoded quality of the untrusted decoder are the SAO parameters. The exact average values can be found in overview Table IV. Considering the ease of getting around this encryption strategy by disabling loop filtering, the minimal amount of encrypted coefficients, and the low impact on the untrusted quality, only encrypting SAO parameters or loop filter coefficients in general is not a good strategy.

Next on the quality impact scale, all the motion related information can be found. When encrypting motion information, random access pictures remain undistorted. In our measurements, this results in one undistorted picture every second. A second observation about motion encryption is that texture information remains recognizable for the viewer. Objects or people can still be identified, but their appearance is largely deformed with blocky structures. The impact of encrypting motion related information ranges from -3.3 dB PSNR for RPS encryption up to -15.5 dB for merge index or MV sign encryption. So, first in this list is the encryption of RPS information. Together with the low impact on quality by an untrusted decoder, it was also mentioned that there is only a small amount of data encrypted, resulting in a vulnerability risk.

A syntax element having a low impact without being a risk for brute force attacks is the MVD size encrypted with a strength of two. Depending on the strength, the delta PSNR impact of MVD size encryption varies between -5.2 dB at strength two down to -10.6 dB at strength eight. A detailed illustration of how the scrambling performance can be gradually increased with increased strength can be found in

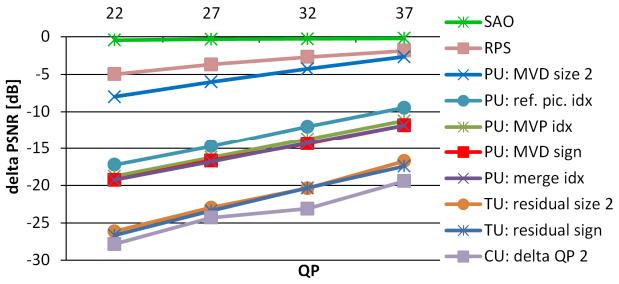


Fig. 1. Scrambling performance, expressed in delta PSNR, of encrypting different syntax elements. Larger absolute deltas indicate higher scrambling impact on video streams decoded by an untrusted decoder.

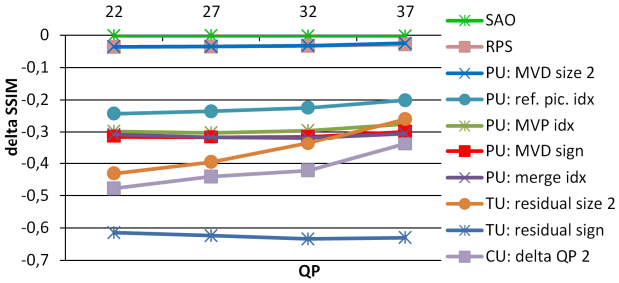


Fig. 2. Scrambling performance, expressed in delta SSIM, of encrypting different syntax elements. Larger absolute deltas indicate higher scrambling impact on video streams decoded by an untrusted decoder.

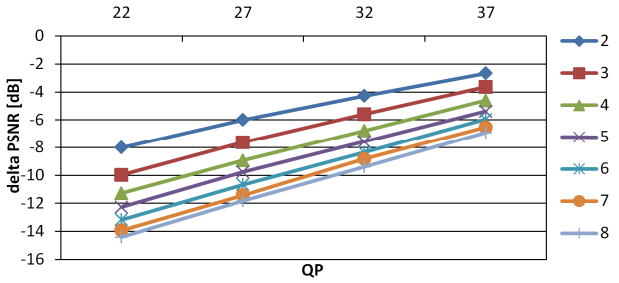


Fig. 3. Scrambling performance, expressed in delta PSNR, of encrypting the MV size at different strengths. Larger absolute deltas indicate higher scrambling impact on video streams decoded by an untrusted decoder.

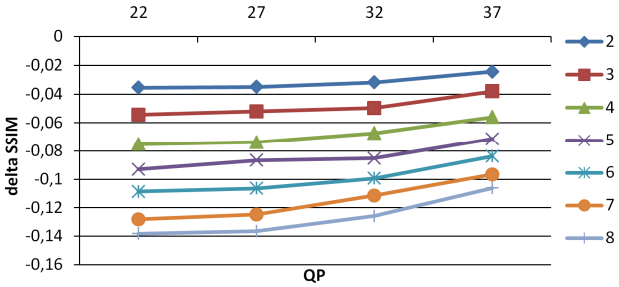


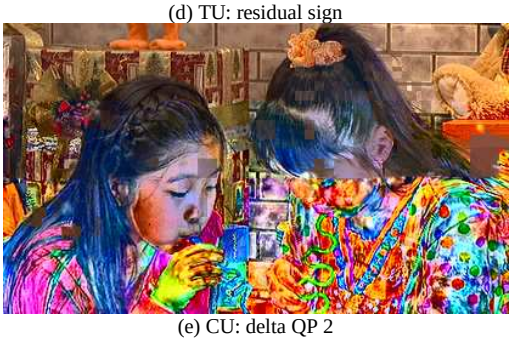
Fig. 4. Scrambling performance, expressed in delta SSIM, of encrypting the MV size at different strengths. Larger absolute deltas indicate higher scrambling impact on video streams decoded by an untrusted decoder.

(a) original

(b) PU: MVD size 2

(c) PU: ref. pic. idx

(d) TU: residual sign



(e) CU: delta QP 2

Fig. 5. Example decoded picture from the BlowingBubbles sequence (picture 142). (a) Decoded by trusted decoder having the decryption key. (b-e) Encrypted with a subset of the investigated syntax elements and decoded by an untrusted decoder.

Fig. 3. Scrambling performance can be further improved by encrypting the reference picture index. With an average delta PSNR of -13.3 dB a slightly more deformed video stream can be obtained. The next syntax elements on the scrambling performance scale related to motion information are the MVP index, the merge index, and the MVD sign. These syntax elements deform the pictures for an untrusted decoder by -15.0 dB, -15.5 dB, and -15.5 dB respectively.

For the encryption of SAO and motion related parameters, the same trend in scrambling performance can be observed in the delta SSIM graphs as given in Fig. 2 and Fig. 4. On an SSIM scale, the impact of SAO encryption gets rounded to 0.00 indicating the small influence of its encryption. For the motion related elements, low impacts can again be observed for RPS and MVD size elements. For encryption of RPS and MVD size with a strength of two, an equal delta SSIM of -0.03 can be noticed in Fig. 2 and Table IV. Similar as for PSNR, depending on the strength of the MVD size encryption, the decrease in SSIM can be observed in Fig. 4. Again, next in line, reference picture index, MVP index, merge index, and MVD sign can be found. Illustrations giving an idea about the impact of MVD size encryption at strength two and reference picture index encryption can be found in Fig. 5 (b) and (c) respectively.

Next, syntax elements deforming the texture information can be found in the delta PSNR comparison in Fig. 1. Based on this delta PSNR, encrypting residual size at strength two and residual sign should have the same PSNR impact on the encrypted video stream. On the contrary, when looking at the delta SSIM graphs in Fig. 2 encryption of the residual sign has a far larger impact than encrypting the residual size at strength two. Visually, it is observed that SSIM more closely corresponds to reality. The result of decoding the BlowingBubbles sequence with the residual sign encrypted, can be found in Fig. 5 (d). In our tested scenario, most residual information can be found in the random access pictures of the video stream. This is caused by the fixed QP setting combined with the hierarchical-B coding structure. Consequently, the main impact of residual encryption could be observed in the random access pictures every second. In these pictures, no prior decoded pictures are used, so totally different colors get introduced by the pseudo random encryption information in these random access pictures. Inter predicted pictures contain less residual data, so the impact on the initially wrong textured information is low. What can be observed are strangely textured objects that move realistically. Printed in this paper and relying on the delta PSNR or delta SSIM graph, encrypting these residual elements seem to deform the video stream a lot, but the natural motion of these strangely colored shapes reveals their identity.

Finally, encryption of the delta QP can be seen as a technique impacting the PSNR the most. Based on the SSIM measurements, there is again a PSNR-SSIM mismatch when encrypting the residual sign, but the exact numbers are irrelevant compared to the real deformations each technique makes. In Fig. 5 (e), an example is given of deformations on the BlowingBubbles sequence. In the left top corner of the picture, it starts with low mismatches between the original and the deformed picture. Going more to the right bottom corner, the QP more and more mismatches the original QP resulting in more deformation of the texture. In the right bottom corner, the mismatch even becomes so high that saturation of the pixel information results in saturated colors. Also for this texture based encryption scheme, the motion information remains unchanged. So, the observation of strangely colored objects moving with natural motion can be made for this technique.

TABLE IV
OVERVIEW OF BD-RATE INCREASE, DELTA PSNR, AND DELTA SSIM OF EACH INVESTIGATED HEVC SYNTAX ELEMENT (IN BOLD: TECHNIQUES WITH THEORETICAL 0.00% BD-RATE INCREASE. 0.00% GAIN AFTER ROUNDING NOT IN BOLD.)

Encrypted element	Strength	BD-rate	Avg. delta PSNR	Avg. delta SSIM
-RPS	-	0.62	-3.3	-0.03
-CU: delta QP	2	0.09	-23.6	-0.47
-CU: delta QP	3	0.13	-23.7	-0.48
-CU: delta QP	4	0.17	-24.6	-0.49
-CU: delta QP	5	0.23	-24.9	-0.51
-CU: delta QP	6	0.26	-24.9	-0.52
-CU: delta QP	7	0.28	-25.0	-0.53
-CU: delta QP	8	0.31	-25.2	-0.53
-PU: ref. pic. idx	-	0.52	-13.3	-0.23
-PU: merge idx	-	4.56	-15.5	-0.31
-PU: MVP idx	-	0.05	-15.0	-0.29
-PU: MVD sign	**-**	**0.00**	**-15.5**	**-0.31**
-PU: MVD size	2	0.00	-5.2	-0.03
-PU: MVD size	3	0.18	-6.7	-0.05
-PU: MVD size	4	0.36	-7.9	-0.07
-PU: MVD size	5	0.39	-8.7	-0.08
-PU: MVD size	6	0.37	-9.5	-0.10
-PU: MVD size	7	0.47	-10.2	-0.12
-PU: MVD size	8	0.73	-10.6	-0.13
-TU: residual sign	**-**	**0.00**	**-21.9**	**-0.63**
-TU: residual size	2	0.33	-21.5	-0.36
-TU: residual size	3	0.97	-21.2	-0.36
-TU: residual size	4	1.49	-21.5	-0.38
-TU: residual size	5	2.48	-21.5	-0.38
-TU: residual size	6	3.10	-21.7	-0.40
-TU: residual size	7	3.99	-21.5	-0.41
-TU: residual size	8	4.64	-21.6	-0.42
-SAO	**-**	**0.00**	**-0.30**	**-0.00**
-MVD sign+residual sign	**-**	**0.00**	**-22.1**	**-0.68**
-MVD sign+residual sign+delta QP	2	0.09	-25.5	-0.77
-MVD sign+residual sign+ delta OP+residual size	2	0.42	-26.1	-0.81

C. Impact on Compression Efficiency

With the BD-rate metric, the impact on compression efficiency between the original and the encrypted video stream is calculated. In Table IV, an overview is given of the average BD-rate increase of encrypting the different syntax elements. In bold, the elements encoded with bypass compression are marked. Because of the bypass compression, encrypting these values gives a theoretical 0.00% BD-rate increase. Measurements confirmed that the change of these values has no impact on the compression of that element or any other syntax element related to it. As interesting bypass coded

elements MVD sign and residual sign should be mentioned. These two elements can already provide motion scrambling and texture deformation at no compression efficiency cost. When preview functionality needs to pass through more visual information, SAO, RPS, and MVD size can be considered. Certainly the MVD size encryption at strength two can be interesting as well because of its negligible impact on compression efficiency (0.00%). Regarding compression efficiency loss, encrypting the MVD size up to a strength of eight only increases the bitrate with 0.73%. Depending on the application and the restrictions of middleboxes in the network, this can still be a valuable option. From the motion related syntax elements, only caution needs to be applied if encryption of the merge index should be considered. Encrypting this element increases the bitrate on average with 4.56%, which is certainly significant for encryption purposes. For deformation of the texture, encrypting the residual size can be an alternative for the residual sign, but the strength should not be increased too much. At strengths two and three, a reasonable BD-rate penalty of respectively 0.33% and 0.97% can be observed. Higher strengths increase this BD-rate up to 4.64%. Improving the strength for this element is also less relevant because the quality is not deformed a lot with increasing strength. This was also observed during informal visual tests. Lower quality impacts can be found when encrypting delta QP information. With a strength of two, only 0.09% BD-rate increase is measured. With encryption of delta QP, the impact of higher strengths does not impact the BD-rate significantly. Delta PSNR, delta SSIM, and observations did not indicate more explicit deformations as well. Therefore, choosing higher strengths than two seems to be irrelevant.

As already mentioned in [20], combining encryption of different syntax elements results in adding together BD-rate increases. This is obvious because encryption happens independently from the encoding process. PSNR and SSIM measurements are not additive because deformations of the same kind can influence each other. Encryption of motion related elements will influence each other's quality results.

Depending on the application and therefore on the constraints of the adaptation devices or other middleboxes in the network, a selection of different syntax elements to encrypt can be chosen. When preview functionality is of importance, motion information based techniques provide a low impact on the quality of untrusted decoders. Except from the merge index, all other motion related elements provide equal kind of deformations at low compression efficiency cost. Then it is only a matter of tuning the amount of scrambling to the preview requirements. When deformation for untrusted decoders needs to be higher, texture deforming syntax elements like residual sign, residual size, or delta QP can be chosen. For residual size encryption, the encryption strength should not be increased because it would only impact compression efficiency without increased scrambling performance.

D. Combinations of different elements

To increase scrambling performance, combinations of different encryption techniques can be applied. Three example combinations are given at the bottom of Table IV. By combining two syntax elements having a theoretical impact of 0.00%, again no impact on compression efficiency can be observed. When the techniques deform different aspects of the video, then visually, the distortions add up as well, but this is not reflected in delta PSNR and SSIM results. The third example adds delta QP encryption and residual size encryption to the combination resulting in an addition of the BD-rate increases. Visually, the impact of residual size will be a small addition compared to the impact of residual sign.

E. Comparison with H.264/AVC

In H.264/AVC, two types of entropy coding can be chosen, namely Context-Adaptive Variable-Length Coding (CAVLC) and the more efficient CABAC. With CAVLC, everything except the residual information has fixed codewords, so encryption can easily be applied without loss in compression efficiency. When CABAC is used in H.264/AVC, the suffix part of the MVD, residual sign, and suffix from residual data are bypass encoded. Encrypting these elements will therefore also result in 0.00% BD-rate increase, at a similar scrambling performance as in this paper with HEVC. So, looking at history, the shift from CAVLC to CABAC in H.264/AVC made it more difficult to perform encryption without impacting compression efficiency. With the shift from H.264/AVC to HEVC, one would expect the same phenomenon, because improved compression efficiency comes at the cost of increased dependency between syntax elements. This paper shows that with HEVC it is still possible to encrypt certain bypass encoded elements at no cost. In addition to the elements that were used in H.264/AVC, results from previously uninvestigated elements are added (e.g RPS, CU: delta QP, PU: ref. pic. Idx, PU: merge idx, PU: MVP idx, and SAO). With these measurements, showing such large variety in encryption strengths, preview functionality can easily be provided at a desired quality. In contrast, with H.264/AVC, it was more challenging [17] to enable such preview functionality.

VI. CONCLUSION

When advanced video applications, located in the network, are included in the video distribution chain, it is beneficial to encrypt only syntax elements from the video stream which are not used by these applications. In this way, the decryption key must not be entrusted to these middlebox devices. Additionally, providing standard compliant encryption guarantees proper operation of devices handling the encrypted video stream. With the provided results on compression efficiency and scrambling performance, decisions can be made about which elements should be encrypted at what compression efficiency loss.

REFERENCES

[1] A. Boho et al., "End-to-end security for video distribution: the combination of encryption, watermarking, and video adaptation," *IEEE Signal Process. Mag*., vol 30, no.2, pp. 97-107, Mar. 2013.

[2] R. Iqbal, S. Shirmohammadi, A. El Saddik, and J. Zhao, "Compressed-domain video processing for adaptation, encryption, and authentication," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 38–50, Apr. 2008.

[3] N. Thomas, D. Redmill, and D. Bull, "Secure transcoders for single layer video data," *Signal Process.- Image Comm.*, vol. 25, no. 3, pp. 196–207, Mar. 2010.

[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, and T.Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649 –1668, Dec. 2012.

[5] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[6] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669 –1684, Dec. 2012.

[7] S.-O. Lee, K.-S. Jung, D.-G. Sim, "Real-time objective quality assessment based on coding parameters extracted from H.264/AVC bitstream," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 1071–1078, May 2010.

[8] X. Lin, H. Ma, L. Luo, and Y. Chen, "No-reference video quality assessment in the compressed domain," *IEEE Trans. Consum. Electron*., vol. 58, no. 2, pp. 505– 512, May 2012.

[9] M. Kim, H. Lee, and S. Sull, "Efficient transform domain transcoding: intra frame of H.264/AVC to JPEG," *IEEE Trans. Consum. Electron*., vol. 57, no. 3, pp. 1362– 1369, Aug. 2011.

[10] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Requantization transcoding for H.264/AVC video coding," *Signal Process.:Image Commun.*, vol. 25, no. 4, pp. 235-254, Apr. 2010.

[11] S.-K. Au Yeung, S. Zhu, and B. Zeng, "Partial video encryption based on alternating transforms," *IEEE Signal Process. Lett.*, vol. 16, no. 10, pp. 893–896, Oct. 2009.

[12] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Secure advanced video coding based on selective encryption algorithms," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 621– 629, May 2006.

[13] Y. Wang, M. O'Neill, and F. Kurugollu, "The improved sign bit encryption of motion vectors for H.264/AVC," 2012, pp. 1752–1756, *20th European Signal Process. Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012.

[14] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," *IEEE Trans. Circuits Syst. Video Technol*., vol. 21, no. 5, pp. 565–576, May 2011.

[15] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[16] Z. Shahid, M. Chaumont, and W. Puech, "Selective and scalable encryption of enhancement layers for dyadic scalable H.264/AVC by scrambling of scan patterns," *IEEE Int. Conf. on Image Process. (ICIP 2009)*, Cairo, Egypt, pp. 1265–1268, Nov. 2009.

[17] E. Magli, M. Grangetto, and G. Olmo, "Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video," *Signal Process.*, vol. 91, no. 5, pp. 1103–1114, May 2011.

[18] T. Stuetz and A. Uhl, "A survey of H.264 AVC/SVC encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 325–339, Mar. 2012.

[19] NIST, "Advanced encryption standard (AES)," FIPS Publication 197, Nov. 2001.

[20] G. Van Wallendael, A. Boho, J. De Cock, A. Munteanu, and R. Van de Walle, "Encryption for high efficiency video coding with video adaptation capabilities," *IEEE Int. Conf. on Consum. Electron. (ICCE 2013)*, Las Vegas, USA, Jan. 2013.

[21] I.-K. Kim, K. McCann, B. Bross, S.-i. Sekiguchi, and W.-J. Han, "HM8: high efficiency video coding (HEVC) test model 8 encoder description," JCTVC-J1002, Stockholm, Sweden, Jul. 2012.

[22] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," document VCEG-M33 of ITU-T Video Coding Experts Group (VCEG), Apr. 2001.

BIOGRAPHIES

Glenn Van Wallendael (S'12) obtained the M.Sc. degree in Applied Engineering from the University College of Antwerp, Antwerp, Belgium, in 2006 and the M.Sc. degree in Engineering from Ghent University, Ghent, Belgium in 2008. Since then, he is working towards a Ph.D. at Multimedia Lab, Ghent University, with the financial support of the Agency for Innovation by Science and Technology (IWT). Main topics of interest are video compression including scalable video compression and transcoding.

Andras Boho received his M.Sc. in Electronic and Computer Engineering at the Pázmány Péter Catholic University, Hungary, in 2009. Since 2010, he has been working towards a Ph.D. at the COSIC research group of the Katholieke Universiteit Leuven.

Jan De Cock (M'06) obtained the M.S. and Ph.D. degrees in Engineering from Ghent University, Belgium, in 2004 and 2009, respectively. Since 2004 he has been working at Multimedia Lab, Ghent University, and iMinds. His research interests include video compression and transcoding, scalable video coding, and multimedia applications.

Adrian Munteanu (M'07) is professor at the Vrije Universiteit Brussel, Belgium, and research leader of the 4Media group at the Institute of Broadband Technology - IBBT, Belgium. Adrian Munteanu received the MSc degree in Electronics and Telecommunications from Politehnica University of Bucharest, Romania, in 1994, the MSc degree in Biomedical Engineering from University of Patras, Greece, in 1996, and the Doctorate degree in Applied Sciences (awarded with highest honors) from Vrije Universiteit Brussel (VUB), Belgium, in 2003. In the period 2004-2010 he was post-doctoral fellow with the Fund for Scientific Research - Flanders (FWO), Belgium, and since 2007, he is professor at VUB.

Adrian Munteanu is the author of more than 200 journal and conference publications, book chapters, and contributions to standards and holds 7 patents in image and video coding. Adrian Munteanu is the recipient of the 2004 BARCO-FWO prize for his PhD work, the co-recipient of the Most Cited Paper Award from Elsevier for 2007 and of the Best Paper Award at the 2011 ACM/IEEE International Conference on Distributed Smart Cameras. Adrian Munteanu currently serves as Associate Editor for IEEE Transactions on Multimedia. His research interests include image, video and 3D graphics coding, distributed source coding, error-resilient coding, multimedia transmission over networks, and statistical modeling.

Rik Van de Walle (M'99) received his M.Sc. and PhD degrees in Engineering from Ghent University, Belgium in 1994 and 1998 respectively. After a visiting scholarship at the University of Arizona (Tucson, USA), he returned to Ghent University, where he became professor of multimedia systems and applications, and head of the Multimedia Lab. His current research interests include multimedia content delivery, presentation and archiving, coding and description of multimedia data, content adaptation, and interactive (mobile) multimedia applications.