Extended Selective Encryption of H.264/AVC (CABAC) and HEVC encoded video streams

Benoit Boyadjis, Cyril Bergeron, Beatrice Pesquet-Popescu, Frederic Dufaux

Extended Selective Encryption of H.264/AVC (CABAC) and HEVC encoded video streams

Benoît Boyadjis[(1), (2)], Cyril Bergeron[(1)], Béatrice Pesquet-Popescu[(2)], and Frédéric Dufaux[(2)]
[(1)] Thales Communications and Security, OPS/HTE/STR/MMP – Gennevilliers, France
[(2)]Institut Mines-Telecom; Telecom ParisTech; CNRS LTCI – 46 rue Barrault, Paris, France
[(1)]*first.last@thalesgroup.com* – [(2)]*first.last@enst.fr*

***Abstract*—This paper proposes an extended Selective Encryption (SE) method for both H.264/AVC (CABAC) and HEVC streams. Namely, this work addresses the main security issue that SE is facing : the content protection, related to the amount of information leakage through a protected video. Our contribution is the improvement of the visual distortion induced by SE approaches. If previous works on both H.264/AVC (CABAC) and HEVC limit encryption to bins treated by one specific mode of CABAC: its *by-pass* mode - which has the advantage of preserving the overall bitrate -, we propose here to also rely on the encryption of the more widely used mode of CABAC: its *regular* mode. This allows to encrypt a major codeword for video reconstruction : the prediction modes for Intra blocks/units. Disturbing their statistics may cause bitrate overhead, which is the trade-off for improving the content security level of the SE approach. A comprehensive study of this compromise between the improvement of the scrambling efficiency and the undesirable aftereffects is proposed in this paper, and a specific security analysis of the proposed CABAC regular mode encryption is conducted.**

***Index Terms*—HEVC, H.264/AVC, Selective Encryption, CABAC**

I. Introduction

H.264/AVC (Advanced Video Coding) [1] and its successor HEVC (High Efficiency Video Coding) [2] are two of the main video coding standards nowadays used for compressing video data, providing efficient tools adapted to multiple uses such as broadcasting, storage, video on demand, etc. Their high compression ratios are partly achieved thanks to the arithmetic coder CABAC (Context Adaptive Binary Arithmetic Coder) used by both standards (alongside with the Context Adaptive Variable Length Coder CAVLC in H.264/AVC). Equipped with a context modeler and a built-in table of probability models, it tries to exploit at best the syntax element statistics to improve the compression efficiency.

Video encryption techniques enable applications like Digital Rights Management (DRM) and video scrambling. When confidentiality is a major priority, encryption of the entire video stream is advised. However, in some application scenarios, Partial Encryption (PE) methods - which aim at scrambling a selected subset of the data to protect - have been proposed as a flexible alternative to standard encryption algorithms. Such approaches can save computational complexity or enable new system functionalities based on priors concerning the message to protect. In this work, we focus on PE methods that preserve the stream format, which we will refer to as Selective Encryption (SE) (Figure 1). Streams protected by SE are *decodable*, which ensures their correct processing by untrusted middlebox in the network (for watermarking, transcoding, etc.), but their intelligibility is restrained.

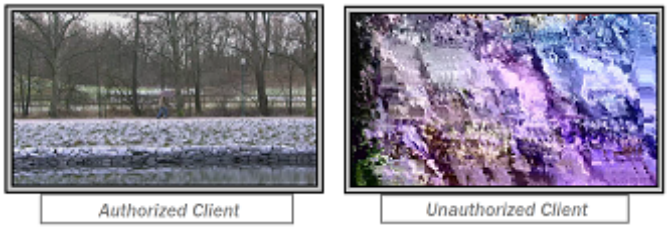


Fig. 1: Selectively Encrypted video streams can be decoded but are visually ciphered for unauthorized users.

SE approaches face two main challenges : first, the distortion induced on the video content heavily depends on the cipherable elements in the stream - constituting the Encryption Space (ES) -, so it is not possible to guarantee that all of the source information is concealed during encryption [3]. Second, the encryption security has to be validated and, in a broader perspective, has to be robust in all types of network / real-time / lossy communications [4].

In this paper, we address the specific issue of the content security and improve the scrambling effect of traditional SE approaches. Based on our previous work on a ciphering transcoder for H.264/AVC and HEVC streams [5], our main contribution implies an extension of the ES available with CABAC encoded streams. In practice, one novel syntax element, selected for its major importance in video reconstruction, is included into the ES : the prediction modes for Intra blocks (H.264/AVC) and Intra units (HEVC). Since their encoding relies on probability models, their encryption may disturb the symbol statistics and cause a bitrate overhead. However, we show that the approach significantly diminishes the information leakage through a protected frame. The present paper gathers a detailed description of the proposed Intra prediction mode encryption, and extensively discusses the miscellaneous consequences of the approach.

The rest of the document is organized as follows: Section 2 briefly presents H.264/AVC and HEVC video coding standards and discusses the use of SE on compressed video streams.

The research leading to these results has received funding from the European Union's Seventh Framework Program ([FP7/2007-2013]) under grant agreement n 288502 (ICT CONCERTO project).

Section 3 details the proposed encryption of Luma prediction modes for Intra blocks/units. In section 4, we propose an evaluation of the proposed scheme with regard to the state of the art approaches, and conduct a specific security analysis of the proposed encryption. Finally, conclusions are drawn in section 5.

II. Selective Encryption in Video Compression

A. Brief presentation of H.264/AVC and HEVC video coding standards

Compression standards H.264/AVC and HEVC implement an hybrid approach exploiting spatial and temporal redundancy contained in a video sequence.

The main characteristics of H.264/AVC standard [1] are the following: each frame is divided in 16x16 MacroBlocks (MB) encoded independently. The encoding protocol of each MB consists in successive steps: MB are first split in 4x4 and 8x8 blocks, then an Integer Transform (IT) followed by a quantization is applied on every block. Predictions between blocks for both Intra and inter frames are then computed, alongside with a motion estimation module, in order to exploit the spatial and temporal redundancy of the source video. Finally, two entropy coders are available for optimizing compression efficiency. H.264/AVC can either use a run length coder CAVLC (Context Adaptive Variable Length Coding) or an arithmetic coder CABAC (Context Adaptive Binary Arithmetic Coding). CAVLC, specified for all profiles in H.264/AVC, is a lower complexity but less efficient entropy coder than CABAC, which is only available in *main* and *high* profiles. This paper will not address the problem of SE of H.264/AVC/CAVLC streams [6][7][8], but will only focus on applying SE on streams encoded with CABAC. Figure 2 presents the overall architecture for encoding video frames with H.264/AVC.

HEVC standard [2] greatly improved the compression efficiency of its predecessor without modifying the whole encoding structure [9]. Nevertheless, several modifications in all the crucial stages were implemented: MBs are replaced with Coding Tree Units (containing Coding Units from 64x64 to 16x16) allowing flexible sub-partitioning using a quadtree structure. In addition, several improvements were made on the other steps of the encoding architecture; increase of the number of prediction modes, new transform sizes from 4x4 to 32x32, new modes available for parallel encoding (Tiles and Wavefront Processing), etc. The numerous features introduced by HEVC standard are explicitly highlighted in [10]. Despite a slightly higher computational cost, CABAC is the only proposed entropy coder. Although it has seen some modifications (essentially simplifications, as described in [11]), this final step of both standards is thus quite equivalent.

B. Main properties of SE

The strong relationship between data compression and encryption was pointed out by C. Shannon in [12] by demonstrating that removing source redundancies could strengthen encryption. Traditional approach for content access control consists of compressing data with a standard coder that

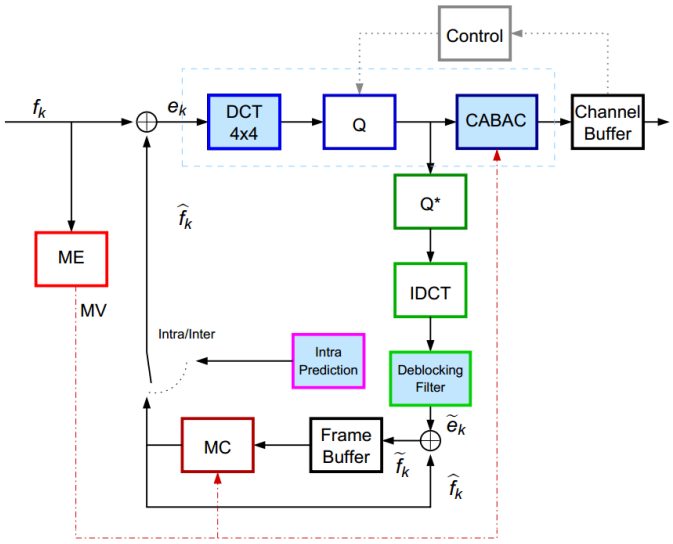


Fig. 2: H.264/AVC-CABAC encoding architecture.

removes (or tries to remove) all redundancies, and then encrypting the whole bitstream: all bits are assumed to be of equal importance or statistics. However, this assumption is hardly valid for H.264/AVC and HEVC video streams since both standards produce streams with a very specific syntax (start codes, headers, etc.). Moreover, in the traditional approach, standard cipher algorithms (e.g. the Advanced Encryption Standard (AES) [13]) bring an additional complexity requirement. If recent software and hardware optimizations of these methods allow the process of up to 150 MB/s [14], the encryption/decryption processes may become time and energy consuming for large-scale data: streams at high resolutions (4K, 8K) and high bitrates.

The most straightforward method to secure video content is thus to encrypt the whole bitstream using standard encryption algorithms. This method called Naive Encryption Algorithm (NEA), which treats the video bitstream as some random binary data, suffers from several drawbacks. First, the encryption/decryption processes are time and energy consuming for large scale data, limiting the suitability for real-time or mobile device supported transmissions. Second, NEA prevents untrusted middlebox in the network to perform post-processing operations on the bitsream such as transcoding or watermarking. Third, the NEA solution does not apply well to scalable video contents, preventing sub-stream extraction for network adaption and error resilience.

In contrast, SE considers the coding structure of the video bitstream and encrypts only a selected subset of the compressed stream. More flexible, SE solutions can provide a different range of functionalities. First, the encrypted stream remains compliant with the standard, and the compression ratio is not (or very little) affected. This ensures correct processing of the protected file by middlebox processes (transcoding, scalable sub-stream extraction, etc.) and makes the protection hardly detectable. Second, since the number of bits to encrypt is drastically reduced compared to NEA, SE introduces only minimum additional delay and complexity. Third, SE can benefit from the stream coding structure to propose specific

encryption architectures. For example, the scrambling effect induced by SE can be weighted as proposed in [15]. It allows video broadcasters to offer a low quality preview functionality that can be used to convince viewers to buy the full content. Another example is the adaption of SE to the protection of Regions of Interest (ROI). Namely, the work of M. Farajallah *et al.* [16] makes use of HEVC slicing of images in tiles to propose a robust ROI based encryption scheme.

Since SE cannot guarantee that all of the source information is hidden during encryption, it is not suitable for applications with strict security requirements. In such cases, SE and standard encryption may be coupled to cumulate the protection strengths. In contrast, when only full quality access is forbidden, SE can successfully be applied as a flexible and cost-effective content access control solution.

C. Overview of recent works on partial and selective of compressed video streams

In the literature, different domains have been considered as valid candidates for PE: pixels (where PE happens before compression), entropy coders or quantized transforms (where alternate compression schemes are exploited), and bitstreams (where PE is performed after - or during - compression). Characteristics of the PE methods directly depend on the domain where they are inserted.

In [17], Yeung *et al.* suggested a transform-based PE method, with the use of different unitary transforms than Discrete Cosine and Sine Transform (DCT/DST) specified in H.264/AVC standard, resulting in a lower coding efficiency. Their work was extended to 8x8 transform size for H.264/AVC high profile in [18]. In [19], Wu *et al.* propose an entropy coding-based PE scheme, with the use of Huffman tables as an alternative entropy coder. However, this technique also causes a bit-rate increase and is vulnerable to plain-text attacks [20]. Other techniques, like key-based interval splitting in [21] or randomized arithmetic coding [22], try to perform encryption during entropy coding. However, this first group of techniques has one main drawback: the resulting streams are not fully format-compliant, which drastically limits its interoperability.

The rest of this study will focus on bitstream-compliant applications of PE, referred to as SE in this paper.

A pixel-based SE scheme is proposed by Carillo *et al.* in [23]. A permutation of pixels that are in the Regions of Interest (ROI) is performed before compression (and conversely after decompression). Although the method ensures the format compliancy of the encrypted stream, some severe drawbacks can be exhibited: the compression efficiency drastically decreases as the ROI statistics are disrupted and it cannot ensure that compression artifacts do not affect the ROI.

Bitstream-based SE schemes may appear as the most efficient solution, as the one developed by Bergeron *et al.* in [6] for H.264/AVC CAVLC. This approach exploits an explicit parsing of the stream to identify and encrypt a set of cipherable syntax elements. The establishment of the ES and the monitoring of the modifications are the main issues of such methods [8][24].

Shahid *et al.* proposed to perform SE on CABAC encoded streams in [25]. The extended application of SE to CABAC encoded streams is a challenging work since inter-dependency between bits of the compressed streams is maximum. They proposed to encrypt *bin strings* instead of the bit-stream, and to select only *model-free bins* in order to preserve context models management. Their works led to a few others, like the proposal of Asghar *et al.* for the SE of scalable mode of H.264/SVC [26]. Some other extensions of the process rely on *tunable* SE schemes, which are either perceptual-based ([24], [27]) or region-based ([28], [29]).

This basis was exploited to perform bitstream-based SE on HEVC streams. Nevertheless, being a rather recent standard, few works can be referenced in this domain. Shahid *et al.* presented in [30] one of the first methods for SE of HEVC, inspired by their previous work on H.264/AVC CABAC encryption. Another method proposed in [31] suggests to pseudo-randomly flip a fixed percentage of *bins* - namely, coefficient signs - to provide a sufficient but minimal encryption. All of these methods rely on *model-free* encryption, which has the great advantage to preserve the overall stream bitrate.

In [15], Van Wallendael *et al.* details an adaptive SE approach and try to gather all the different cipherable syntax elements in HEVC streams. Their work distinguishes from previous methods as it also implies the explicit management of CABAC context modeler. Their encryption scheme authorizes bigger variations of cipherable codewords than the traditional bitwise modifications, which improves the induced distortion at the cost of a decreased compression efficiency. Their paper proposes one of the first analysis of this specific compromise between the scrambling performance and the bitrate increase. In the present paper, another approach based on an explicit monitoring of CABAC is proposed, which specifically focuses on Luma prediction modes for Intra blocks/units. Since they have a major role in the video reconstruction, encryption of this specific syntax element has been proposed for H.264/AVC-CAVLC in some papers [24][28][32]. In the case of CABAC encoded video streams, and more particularly within the recent HEVC standard, their encryption is not straightforward, and, to the best of our knowledge, has not been proposed yet.

III. Content security: CABAC regular mode selective encryption

A. Understanding the CABAC entropy coder

The generic design of CABAC is a combination of an adaptive binary arithmetic coding technique with a well-designed set of context models. Figure 3 shows the standard encoding process for a single syntax element with CABAC for both H.264/AVC and HEVC. We remind here that the major difference between the CABAC of both standards is the simplification of the context modeling stage: decrease of the number of probability models, increase of the number of *by-pass* treated bins [11]. The remainder of this section details the step-by-step encoding protocol.

- **Binarization**: First, a given non-binary valued symbol is mapped to a binary codeword called *bin string*. This unique representation obtained during the binarization process should be close to a minimum-redundancy code. In practice, several

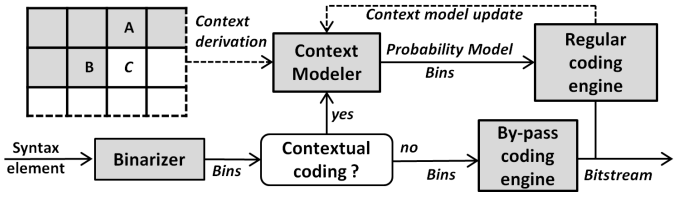


Fig. 3: The different steps of Context Adaptive Binary Arithmetic Coding (CABAC)

	H.264/AVC	HEVC	Has impact on :
MVD sign	Yes	Yes	P frames
MVD value	Yes*	Yes	P frames
MV ref idx	No	Yes	P frames
MV merge idx	-	Yes	P frames
MVP idx	No	Yes	P frames
CU/MB delta QP	Yes	Yes	I and P frames
Residual signs	Yes	Yes	I and P frames
Residual values	Yes	Yes	I and P frames
SAO filter related codewords	-	Yes	I and P frames

TABLE I: Cipherable *by-passed* syntax elements in both standards (*: under constraints)

basic code trees are used for a simple on-line computation [33]. If the input symbol is already in binary form, this first binarization module is by-passed.

Depending on the *a priori* estimations on the probability distribution of the symbol that is being encoded, two different modules can perform the arithmetic coding.

- **By-pass coding engine**: The *by-pass* mode is chosen for symbols whose distribution is assumed to be equiprobable. In this case, an adequate probability model does not exist, so *bin strings* are coded without any contextual information.

- **Regular coding engine**: The most interesting aspect of arithmetic coding is the possibility to interface efficiently a model representation and the subsequent arithmetic coding, which is the core of CABAC *regular mode*. The design of adequate statistical models that are to be kept up-to-date is thus of prime importance. In order to minimize the computational costs, CABAC protocol only exploits a limited number of context templates and limits its use to *bin strings* obtained after binarization. The probability model of each bin is determined by the type of syntax element, the bin position and from the context (e.g., spatial neighbors, position in CTU, ...). Once the probability model is selected, the arithmetic coding engine manages the *bin* encoding, then updates the model according to the encoded *bin* value.

B. CABAC selective encryption

The aim of SE techniques is to identify all (or a selected part) of the stream syntax elements that can be scrambled without disrupting a standard decoding process. However, due to the arithmetic encoding step, *bits* in CABAC compressed streams are very interdependent. The adequate spot for performing SE is thus after the binarization step [30], which allows to work on *bin strings* that have a meaning from a decoder point of view. This is the main difference between SE of CAVLC encoded streams, which can be performed on the fly, and SE of CABAC encoded streams, which requires a prior arithmetic decoding stage.

In the SE approach, a *bin string* is considered as cipherable if it satisfies the following constraints. First, the *bin* modification must not modify the overall decoding process. Namely, the interpretation of changes must not lead to any mismatch between encoding and decoding processes (initialization and termination protocols, number/order of calls for data, selection of the related probability models, etc.). Plus, encryption must preserve the symbol validity, which restricts modifications to a domain in which we ensure that a standard decoder will accept the modified symbol. Checking both constraints ensures the compatibility of the modified stream with the video standard.

Since *by-passed bins* do not exploit and update probability models during the arithmetic coding, their encryption is possible as long as the induced modifications do not disrupt a standard decoding process [30]. Some cipherable syntax elements are into this first category of *by-pass* encoded codewords: sign of Motion Vector Difference (MVD), Sample Adaptive Offset (SAO) related information in HEVC, etc. Table I summarizes all the identified cipherable codewords treated by the *by-pass* mode of CABAC, which will be considered in the rest of this paper as the **state-of-the-art ES** [15]. Note that a few more codewords - motion vector reference indexes (MV ref idx) and predictor indexes (MVP idx) - are available for encryption in HEVC standard, thanks to the simplification of the CABAC module. Two main categories can be distinguished: **motion encryption** (sign/value of MVD, MV ref idx, MVP idx) and **texture encryption** (sign/value of residual data, delta QP, SAO related codewords).

The limited number of *by-passed* syntax elements raises the question of *sufficient deterioration* of the encrypted stream, particularly on I frames. Indeed, motion encryption has no impact on these key frames, limiting the visual deterioration to the scrambling of residual data (as SAO filter encryption has a very limited impact on the video reconstruction). To achieve superior levels of protection, authors of [15] performed non-bitwise encryption and introduced the concept of encryption strength. In the present paper, a complementary way of improving the encryption strength is proposed, which rely on the inclusion into the ES of a novel codeword encoded with CABAC *regular* mode.

From the arithmetic coding engine perspective, encryption of CABAC *regular* mode is similar to *bypass* mode encryption. However, particular attention has to be given to the context modeler: indeed, changing the value of a coded *bin* must be reflected in the subsequent probability model update. The main aftereffect of *regular* mode encryption is the perturbation of the syntax elements statistics, which decreases the CABAC compression efficiency. This is the main drawback of the proposed method, as *by-pass* encoded bins can be encrypted without modifying the source bitrate [25]. However, experimental results detailed in this paper show that the bitrate increase remains acceptable (empirically up to 0.5%). Some cipherable codewords may be identified within this available set of *regular* mode treated syntax elements. In order to significantly improve the impact of the ES on I frames, we chose to focus our study on the encryption of the Luma

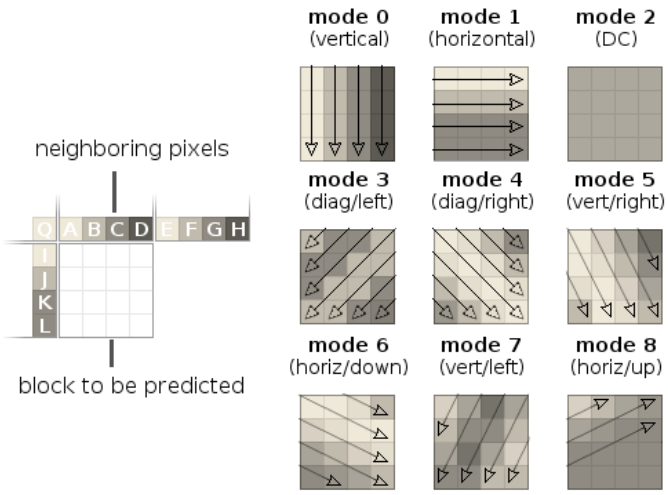


Fig. 4: The nine different prediction modes in H.264/AVC.

prediction modes for Intra blocks/units, because of their major role in the video reconstruction.

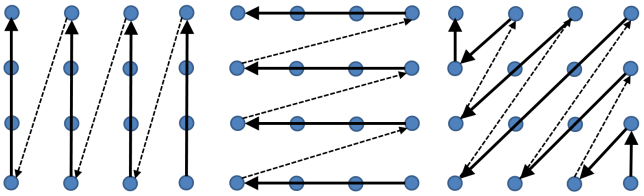
C. Encryption of Luma Prediction modes for Intra blocks in H.264/AVC-CABAC

H.264/AVC and HEVC both rely on Intra prediction for efficiently exploiting the spatial redundancy between neighboring blocks. Since both standard process images in raster scan order, previously decoded data is often available during the decoding of a block/unit. The Intra prediction is used to exploit this information, by interpolating top and left data alongside a given direction. In H.264/AVC nine modes are thus implemented (Figure 4) to efficiently propagate basic structural information between neighboring blocks.

In practice, for each current block during the decoding process, a most probable mode is derived from adjacent blocks as the minimum of their respective Intra prediction modes. The current block is either encoded with this mode, which is specified with a flag (*prev_Intra4x4_pred_mode_flag*), or encoded with one of the eight remaining modes. In this case, the mode is coded with a fixed length three *bits rem_Intra4x4_pred_mode* codeword. Both of these syntax elements use CABAC *regular mode* and probability models in order to improve the compression capabilities.

To modify the prediction modes is an efficient way to induce a strong visual scramble in protected streams. Thus, it has already been proposed for H.264/AVC-CAVLC in [32]. To ensure the validity of the protected stream, attention has to be paid to blocks laying on edges of image slices. Indeed, all modes are not permitted for these blocks, which prevents their encryption. In [32], prediction modes are thus modified for all other blocks with a combination of permutations and bitwise modifications of the original prediction modes.

In this paper, we propose to reuse this method to encrypt luma prediction modes for H.264/AVC-CABAC encoded video streams. Since their encoding rely on probability models, the relative codewords can be modified as long as model are updated accordingly. Doing so, we ensure that the resulting



(a) SCAN_VERT *5<pred mode<16*
(b) SCAN_HORIZ *21<pred mode<31*
(c) SCAN_DIAG *otherwise*

Fig. 5: The three different scanning modes in HEVC, depending on the prediction mode used by the current unit.

stream is format compliant and can be decoded by standard H.264/AVC-CABAC decoders. However, disturbing the symbol statistics may diminish the compression efficiency, which is the evident counterpart to the improvement of the scrambling performance of *by-pass only* scrambling schemes.

D. Encryption of Luma Prediction modes for Intra units in HEVC

Some modifications of the Intra prediction process have to be mentioned between H.264/AVC and HEVC. First, the number of prediction modes has increased (from 9 to 34). It ensures an increased precision in the direction interpolation and largely improves the predictive performance for big size units. Another important modification is the new dependency between a unit prediction mode and the scanning mode used for residual coding (Figure 5). This latter point imposes a very specific monitoring of the encryption with HEVC, which will be detailed in the next subsection. Finally, the encoding process has been improved compared to H.264/AVC.

Indeed, the use of a most probable mode has been extended to three default modes to improve the compression capabilities. In practice, their encoding still depends on a flag called *prev_Intra_Luma_pred_flag* (the syntax of this section is directly extracted from [10]).

Flag set to '1': encoding by MPM (Most Probable Mode). A *mpm_idx* is coded with two (or one) *by-passed* bin(s) (truncated unary code of length 2) specifying which of three pre-selected modes is chosen. Pre-selected modes are derived from the neighborhood (namely, from top and left unit modes) and stored in a table called *CandModList[]*.

Flag set to '0': the prediction mode is encoded separately. A *rem_Intra_Luma_pred_mode* codeword - fixed length 5 *by-passed* bins - is used to infer the prediction mode as follows (where *CandModList[]* is the same table as above, simply reordered in ascending order):

```
for (i = 0; i < 3; i ++)
    if (rem_Intra_Luma_pred_mode > CandModList [i])
        rem_Intra_Luma_pred_mode ++;
IntraPredMode = rem_Intra_Luma_pred_mode;
```

Consequently, one can note that in HEVC standard, there is a strong dependency between the current unit mode and its neighborhood, even in the case of a *prev_Intra_Luma_pred_flag* set to '0'.

We propose in this paper to encrypt all the prediction modes for Intra units, i.e. all *mpm_idx* and *rem_intra_Luma_pred_mode* present in the original stream.

Nb. coded Coeff.	SCAN_VERT	SCAN_HORIZ	SCAN_DIAG
1	(0,0)	(0,0)	(0,0)
2	(0,1)	(1,0)	(0,1)
3	(0,2)	(2,0)	(1,0)
4	(0,3)	(3,0)	(0,2)
5	(1,0)	(0,1)	(1,1)
..	..	..	..
16	(3,3)	(3,3)	(3,3)

TABLE II: Coordinates of the last coded coefficient in a 4x4 unit, depending on the number of coded coefficients and the scanning mode used. A similar look-up table is built for 8x8 units.

However, some precautions must be taken in order to ensure the format compliancy of the encrypted stream. The first constraint to be highlighted is the availability of the current unit to be encrypted. As for H.264/AVC, all prediction modes may not be allowed for units lying on the edges of the image slices. Therefore, these units are not encrypted for security.

This condition alone does not suffice to produce format compliant streams. Indeed, one novelty of HEVC standard is the use of three different scanning modes described in Figure 5 (SCAN_VERT, SCAN_HORIZ and SCAN_DIAG) for 4x4 and 8x8 units, particularly significant in our context. Indeed, the choice of a scanning mode modifies the way residual data is encoded. Therefore, if SE is applied to Intra prediction modes, a standard decoder will use potentially modified scanning modes to decode the residual data. Due to the very strict formatting of the encoded residual, such a modification disturbs the decoding process and prevents the normal stream decoding.

Consequently, we propose in the next subsection a low-cost method to re-encode the residual data according to the modified prediction mode obtained after encryption. One can note that the knowledge of the prediction mode *obtained after encryption* is not straightforward, due to the strict dependency between a unit mode and its neighborhood. In practice, our approach thus requires to keep track of the successive modifications in order to correctly anticipate the prediction modes which will be inferred by a standard decoder processing the protected stream.

E. HEVC: suitable encoding of Residual Data

Figure 5 details the different scanning modes used depending on the unit prediction mode (note that coding units strictly bigger than 8x8 always exploit the diagonal scan). In HEVC standard, the first operation made to decode the residual data is to recover the number of coded coefficients in the unit. This information is derived from the position of the last coded coefficient - namely, its coordinates inside the unit -, and from the scanning mode used. For example, if the coordinates of the last coded coefficient are (1,1), there are 5 remaining coded coefficients with both horizontal and vertical scans, whereas only 4 with the diagonal scan (Figure 5).

We now consider the specific case of an Intra unit whose scanning mode has been modified by SE of its prediction mode. One can note that its size is either 4x4 or 8x8, since all other unit sizes use the diagonal scan. If the residual data buffer has been rewritten without any precaution, there is a high risk of a modification of the number of encoded coefficients inside the unit, which prevents the stream normal decoding. To correctly encode the residual data according to the modified scanning mode, one solution would be to simply re-encode the whole unit. However, we propose an efficient and inexpensive alternative: to modify only the position of the last significant coefficient, ensuring that the *same* number of coefficients is processed in the original and the encrypted unit. This cost-effective solution allows to preserve the size (in *bins*) of the encoded buffer and induces an additional scramble in the unit due to the shift of the residual coefficients.

Thus, we use two look-up tables (see Table II for the 4x4 look-up table) that summarize, depending on the scanning mode and the total number of encoded coefficients, the position of the unit last encoded coefficient. Then, for each unit whose scanning mode has been modified by SE, we use them to infer the equivalent coordinates (in terms of total number of coded coefficients) of the last coefficient in the encrypted unit.

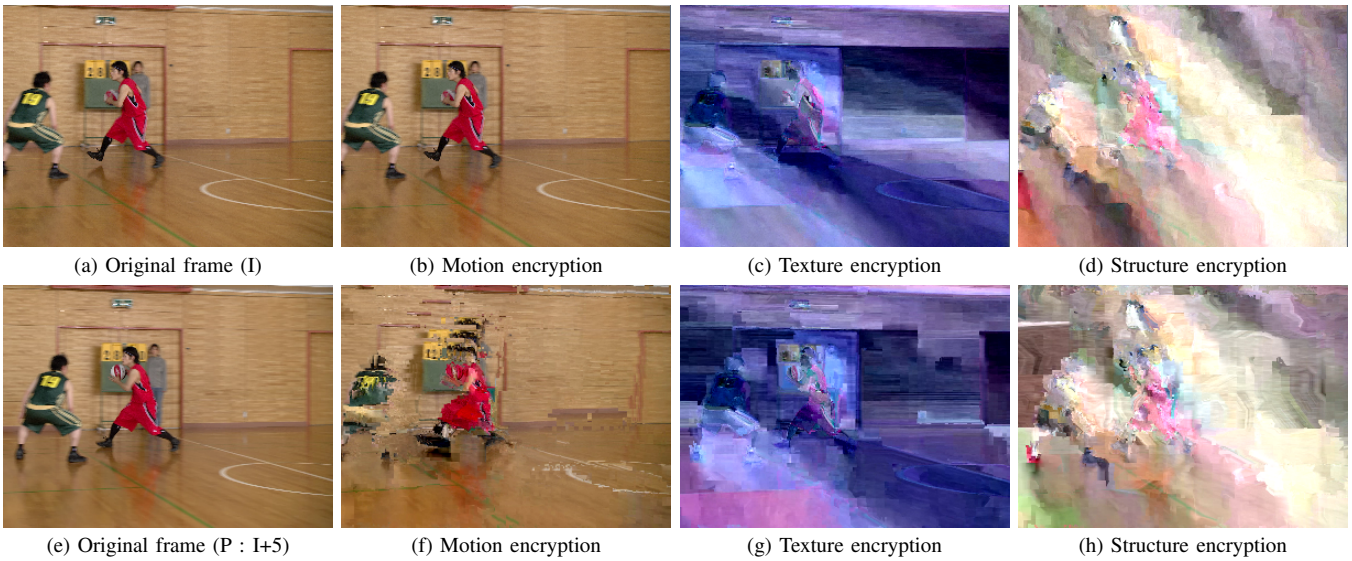
Note that these targeted coordinates exist and are unique: it is a bijective operation, which enables us to ensure the symmetry of the protocol. It only remains to binarize according to HEVC standard the values obtained [9]. These values then replace the original ones in the residual data buffer, while the remaining data is not modified. Doing so, we ensure that residuals fit with encrypted prediction modes, which guarantees the correct decoding of the stream by standard decoders.

Our approach can be summarized as follows. For each *cipherable* unit (not lying on an edge of a slice or a tile), the prediction mode is encrypted by bitwise modifications on the relative codewords. All changes are interpreted in order to infer the modified prediction modes, i.e. the ones that will be read by a standard decoder. Residual buffers are then considered only if their relative scanning mode has been changed during the encryption process. In this case, the position of the last coded coefficient is interpreted from the original stream. Two look-up tables (for 4x4 and 8x8 units) are used to derive the corresponding coefficient position according to the new scanning mode. These coordinates replace the original ones in the residual buffer, while the remaining data is not modified.

IV. Experimental results

A. Methodology

In a standard approach, SE is embedded inside the arithmetic coder of the video encoder/decoder. Such a deployment ensures that the complexity induced by SE is minimal. However, considering the use of off-the-shelf optimized encoders and decoders, the embedding of the specific SE module is not straightforward, which may limit the applicability of the method. To circumvent this difficulty, we choose to implement the SE framework inside a ciphering transcoder, as we already proposed in our previous work [5]. Although this choice is not optimal in terms of complexity, it is an autonomous tool that does not require changes of the pre-existing encoders /



(a) Original frame (I) (b) Motion encryption (c) Texture encryption (d) Structure encryption

(e) Original frame (P : I+5) (f) Motion encryption (g) Texture encryption (h) Structure encryption

Fig. 6: Visual impact of *motion*, *texture* and the proposed *structure* encryption on a zoomed part of H.264/AVC encoded BasketballDrive sequence

decoders and therefore retains the advantages of graphic accelerators or other hardware optimizations. Plus, it is developed as a symmetrical operation on the stream, which again limits the equipments needed to set up the device.

To perform the bitwise modifications on the *cipherable bins*, we use a standard encryption algorithm: the Advanced Encryption Standard (AES) [13]. More precisely, we implemented the CTR mode which uses a 128-bit Initialization Vector (IV) - often built from the concatenation of a fixed seed and a counter - to generate a pseudo-random sequence. When a *bin* is to be ciphered, the last bit of the sequence is extracted and XORed with the original *bin* to produce the protected *bin*. If the sequence is emptied, the counter part of the IV is incremented and a new pseudo-random sequence is generated. AES-CTR mode combines many advantages: its robustness is widely validated [34], it is a pre-calculable algorithm that allows to start decryption from any point in the stream, and its properties of symmetry and parallelism are particularly convenient for software/hardware optimizations [35].

The effect of transmission errors will not be extensively discussed in this paper. Yet, some solutions exist in the literature as the one we proposed in our recent work [4]. It implements a modified AES-CTR mode in order to allow a per-slice synchronization of the decryption, based on the use of identified *uncipherable* elements of the communication. This procedure prevents potential transmission errors from causing a decryption drift that propagates to the next slices. Thus, it provides robustness to SE communications over error-probe transmission channels. We invite the reader to refer to [4] for further details on the proposed AES-CTR specific set-up and an analysis of its robustness against transmission errors.

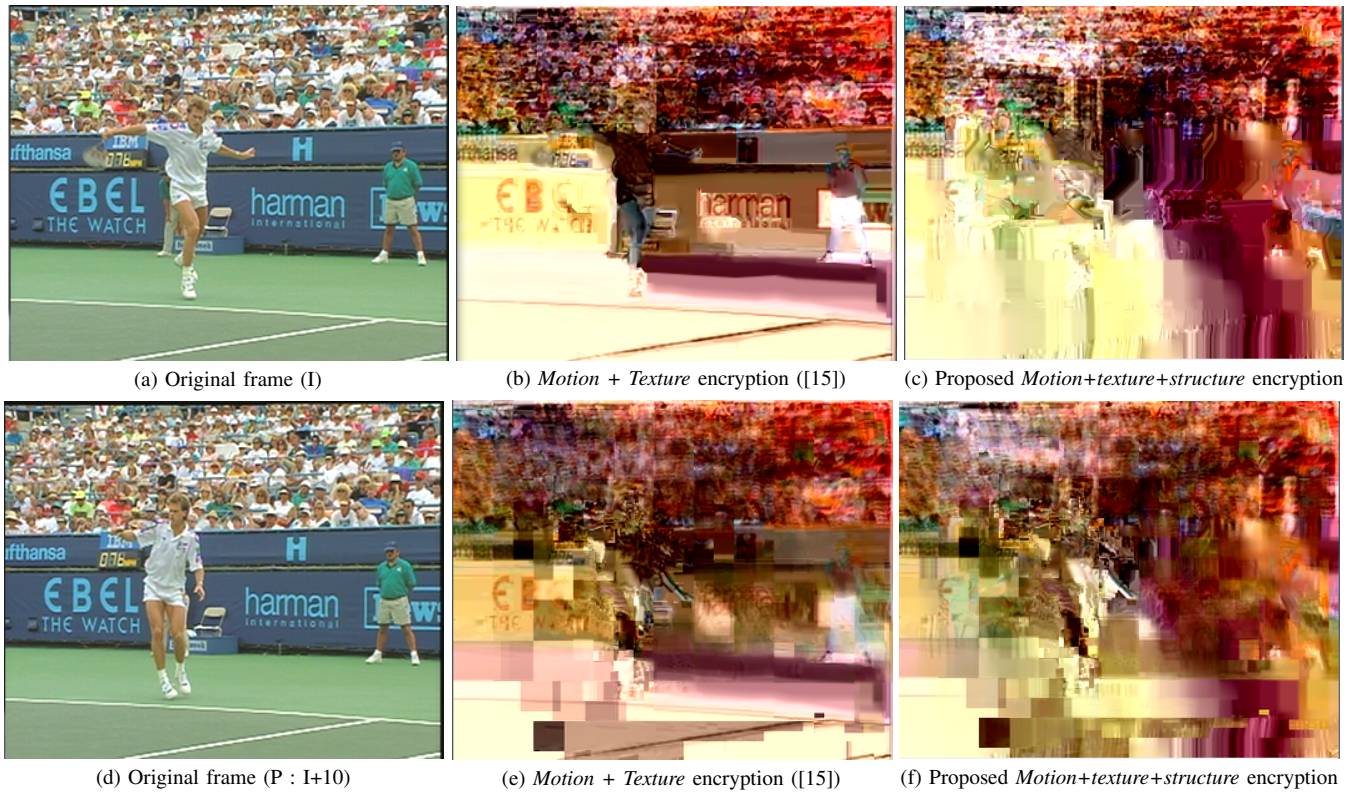
In this paper, security is measured as the amount of information leakage through a protected frame. The evaluation of the scrambling performance of the proposed encryption scheme will be based on a set of sequences, encoded with H.264/AVC and HEVC reference softwares (JM v18.4 and HM v16.3). In order to exhibit significant results, our set of test videos, extracted from HEVC standardization process [36], gathers resolutions ranging from 416x240 (BasketballPass) to 1080p (BasketballDrive). The encoding use a 16 frames Group Of Picture (GOP) with a hierarchical B structure for good compression performance, and a Quantization Parameter (QP) of 32. The set of clear sequences is transcoded and encrypted with the proposed extended ES to produce a set of protected sequences. A comparative evaluation is proposed with the method that produces the best scrambling performance in the literature [15]. Our implementation of the state of the art approach performs bitwise modifications only on each cipherable codeword, which allows to preserve the overall bitrate of the protected streams. Please note that the delta QP codeword is not available for encryption in our study since we do not perform bitrate constrained encodings. A second set of protected videos is thus generated with the aforementioned method.

Both of the protected sets are then decoded by an untrusted device (which does not have the encryption key), and the results are compared to the original streams.

B. Visual analysis

Since SE methods aim at protecting a video stream by deteriorating its comprehensibility, the available content security level is directly related to the amount of information leakage in the protected frames. Note that the visual analysis performed in this section is complemented with additional examples (images and videos) on our academic website [37].

As we already pointed out in Table I, the different cipherable syntax elements have different impacts on the video rendering. To correctly describe this property, we propose a semantical decomposition of the ES as follows:



(a) Original frame (I) (b) *Motion + Texture* encryption ([15]) (c) Proposed *Motion+texture+structure* encryption

(d) Original frame (P : I+10) (e) *Motion + Texture* encryption ([15]) (f) Proposed *Motion+texture+structure* encryption

Fig. 7: Overall visual results of the SE approaches on HEVC encoded Stefan sequence (CIF, QP32).

Type of encryption	*Related cipherable elements*
Motion encryption	MVD sign/value, MV ref idx*, MV merge idx*, MVP idx*
Texture encryption	Residual sign/value, delta QP, SAO filter related codewords*
Structure encryption	Luma prediction modes (Intra)

**: HEVC only*

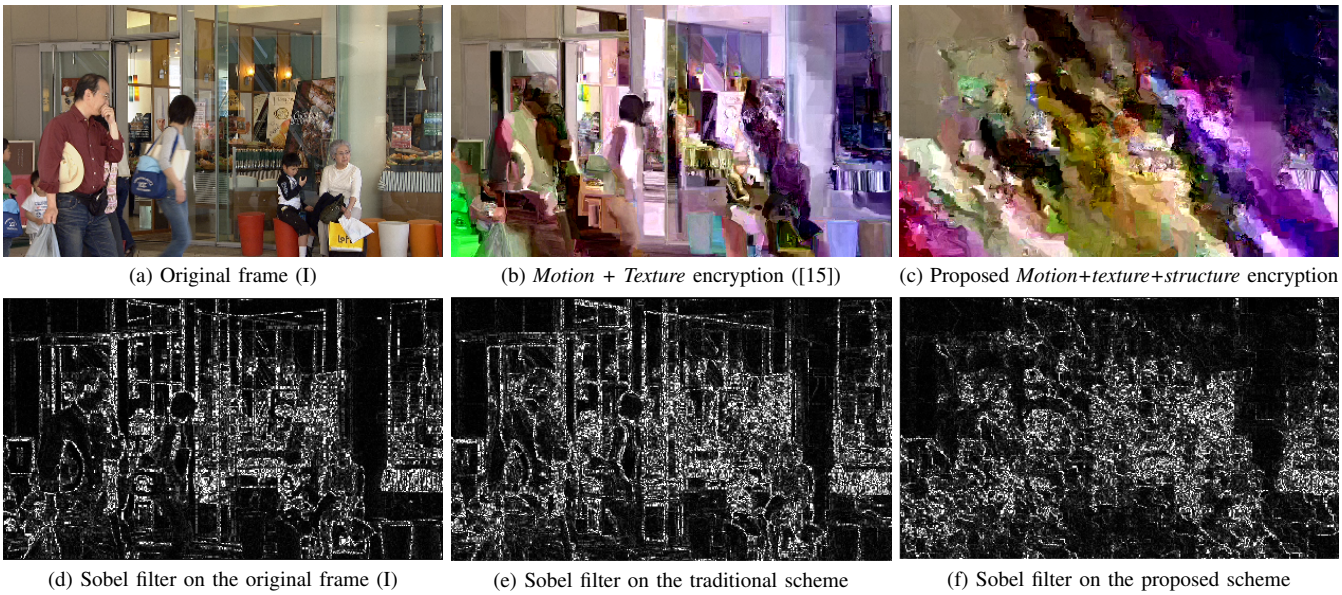
Two quick remarks can be made on this proposed decomposition. First, the first two categories form the state-of-the-art ES as defined in [15] whereas the latter corresponds to our present contribution. Second, as illustrated in Table I, one can note that *motion encryption* has no impact on Intra-coded frames.

In Figure 6, we gathered the visual results of performing SE with each separate encryption domain. It confirms the idea that each encryption domain has a specific effect. Particularly, Figures 6c and 6g justify the choice of denoting as *texture encryption* the modification of the residual data. Indeed, the induced scrambling impacts colors and flat regions whereas it hardly dissimulate the structural information (background, white lines on the floor). In contrast, strong structural deterioration is observed with *motion encryption*, but on moving areas of P/B frames only.

In Figure 7, a comparative evaluation of the proposed SE scheme is performed with regard to the standard approach [15]. In Figure 7e, it is noticeable that *motion encryption* has contributed to the degradation of some structural contents (body of the player, background advertisements) compared to the Intra frame (fig. 7b). The main drawback of the standard ES is highlighted here: the protection of Intra frames, relying on *texture encryption* only, cannot efficiently conceal structural information. In comparison, our approach (figs. 7c and 7f) offers a far more robust dissimulation of the content, thanks to the efficient structural scrambling, even on the Intra coded frame. This improvement is obtained thanks to the proposed *structure encryption* isolated in Figures 6d and 6h, which efficiently diminishes the structural coherence of protected frames. In order to emphasize this observation, the application of a Sobel filter [38] on both protected streams is performed in Figure 8. The Sobel filter is based on an isotropic 3x3 gradient which emphasizes edges and transitions. Each pixel in the filtered frames in Figures 8d, 8e and 8f thus corresponds to the magnitude of the local gradient on that pixel. Even if it produces relatively crude results - particularly for regions with high frequency variations -, the Sobel filter allows an inexpensive and rather efficient analysis of the structural information within an image. It is clearly noticeable that our method conceals more efficiently edges and shapes (figs. 8e and 8f), which confirms that the proposed ES extension diminishes the information leakage through the protected frames.

C. Numerical analysis

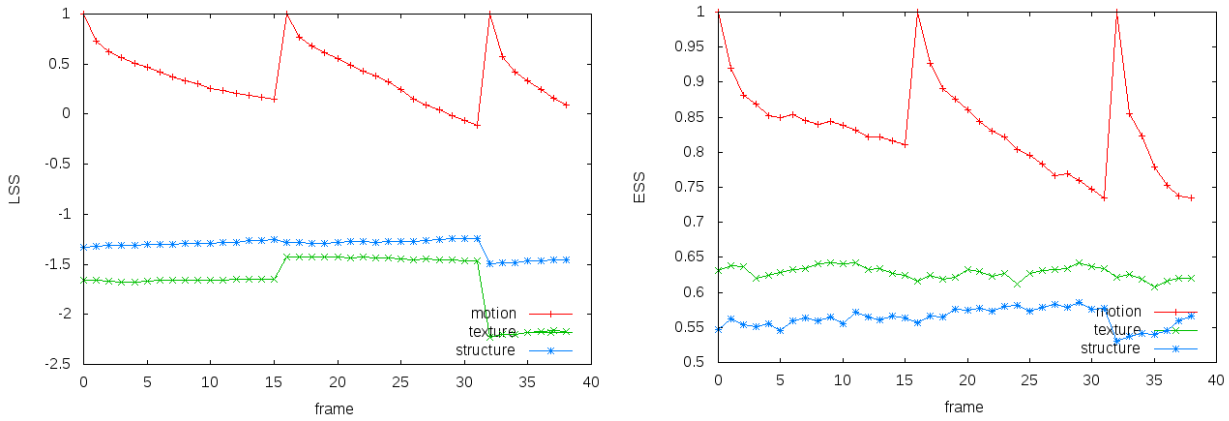
1) Quantifying the increased visual distortion: To support the visual results illustrated previously, we provide in this section a numerical analysis of the SE efficiency. The metrics used in the literature to describe the scrambling effect on video contents are usually the Peak Signal to Noise Ratio (PSNR), the Structural SIMilarity (SSIM) and their Bjontegaard formulation [39]. Their computation allows to efficiently describe the



(a) Original frame (I) (b) *Motion* + *Texture* encryption ([15]) (c) Proposed *Motion+texture+structure* encryption

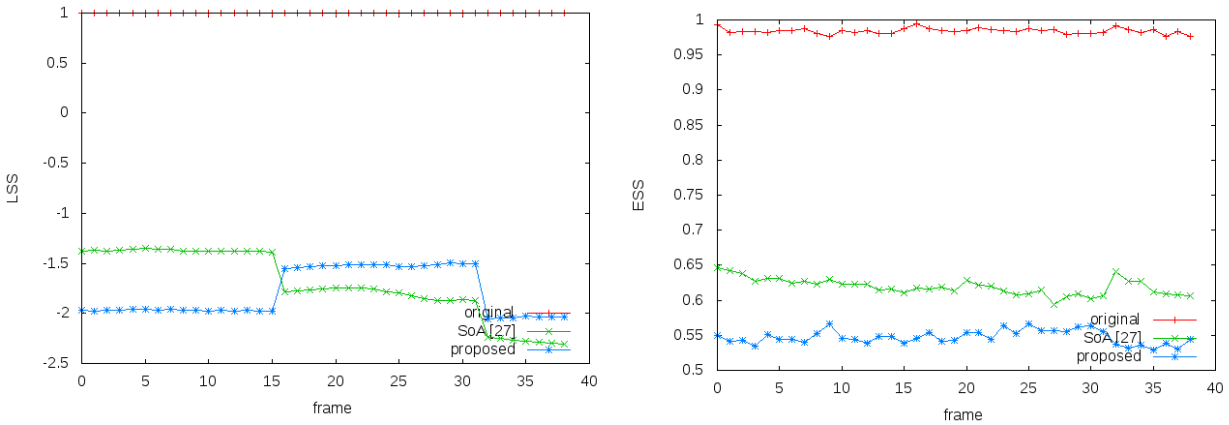
(d) Sobel filter on the original frame (I) (e) Sobel filter on the traditional scheme (f) Sobel filter on the proposed scheme

Fig. 8: Application of a Sobel filter on the first (I) frame of H.264/AVC encoded BQMall sequence encrypted with both SE methods. One can note the high preservation of the structure in the center image.



(a) Luminance similarity scores (b) Edge similarity scores

Fig. 9: Frame per frame numerical analysis of the effect of *motion*, *texture*, and the proposed *structure* encryption on the first 40 frames of BQMall sequence (HEVC, 832x480, QP22, GOP16).



(a) Luminance similarity scores (b) Edge similarity scores

Fig. 10: Frame per frame numerical analysis of the proposed encryption (*motion* + *texture* + *structure*) compared to the state of the art approach [15] (*motion* + *texture*) on the first 40 frames of BQMall sequence (HEVC, 832x480, QP22, GOP16).

H.264/AVC												
Sequence (Class)	Original				State of the art [15]				Proposed Extended Encryption Space			
	PSNR	SSIM	LSS	ESS	PSNR	SSIM	LSS	ESS	PSNR	SSIM	LSS	ESS
BasketballDrive (B1)	44.2	0.943	1.0	0.907	12.3	0.32	**-2.4**	0.37	**10.2**	**0.27**	-1.45	**0.33**
Kimono (B2)	43.1	0.945	1.0	0.947	9.17	0.33	-2.5	0.45	**6.2**	**0.29**	**-3.78**	**0.42**
BQMall (C)	40.0	0.934	1.0	0.924	10.9	0.25	-1.6	0.60	**10.1**	**0.24**	**-1.9**	**0.49**
BasketballPass (D)	41.9	0.932	1.0	0.922	**13.8**	**0.24**	**-1.2**	0.43	13.9	**0.24**	-1.1	**0.37**
Vidyo1 (E)	43.6	0.979	1.0	0.960	10.9	**0.46**	-1.91	0.46	**10.3**	**0.46**	**-1.93**	**0.40**
HEVC												
Sequence (Class)	Original				State of the art [15]				Proposed Extended Encryption Space			
	PSNR	SSIM	LSS	ESS	PSNR	SSIM	LSS	ESS	PSNR	SSIM	LSS	ESS
BasketballDrive (B1)	41.75	0.952	1.0	0.911	11.4	**0.40**	-1.83	0.38	**10.4**	0.43	**-2.10**	**0.36**
Kimono (B2)	43.1	0.958	1.0	0.952	10.1	0.32	**-2.11**	0.48	**6.6**	**0.27**	-2.05	**0.42**
BQMall (C)	41.7	0.943	1.0	0.962	**10.1**	0.28	-1.70	0.62	10.2	**0.24**	**-1.79**	**0.51**
BasketballPass (D)	44.6	0.936	1.0	0.910	**12.5**	0.35	-1.22	0.44	12.6	**0.34**	**-1.23**	**0.37**
Vidyo1 (E)	44.9	0.992	1.0	0.982	11.6	**0.45**	-1.67	0.49	**11.0**	**0.45**	**-1.77**	**0.38**

TABLE III: Overall results of the proposed scheme for both H.264/AVC and HEVC compressed streams (200 frames).

	I	P/B	Sequence	Min./Max.
Bitrate increase	+0.30%	+0.03%	+0.08%	+0.0% / **+1.4%**
ES extension	+36%	+4.2%	+12%	+0.0% / **+123%**
$\Delta PSNR$ (dB)	-25.0	-24.32	-24.35	**-30.1** / -14.1
$\Delta SSIM$	-0.32	-0.298	-0.299	**-0.42** / -0.25

TABLE IV: Other numerical consequences of the proposed Intra prediction mode encryption on H.264/AVC CABAC mix_sequence stream (QP30, CIF, 2700 frames).

	I	P/B	Sequence	Min. / Max.
Bitrate increase	+0.89%	+0.21%	+0.40%	+0.0% / **+2.0%**
ES extension	+57.0%	+11.4%	+24.1%	+0.0% / **+149%**
$\Delta PSNR$ (dB)	-27.9	-26.13	-26.25	**-31.4** / -13.3
$\Delta SSIM$	-0.42	-0.38	-0.39	**-0.49** / -0.33

TABLE V: Other numerical consequences of the proposed Intra prediction mode encryption on HEVC mix_sequence stream (QP28, CIF, 2700 frames).

losses in quality (PSNR) and in structural coherence (SSIM) inside the frames. Nevertheless, PSNR/SSIM correlation to the human perceived intelligibility is limited [40] and they do not perform well in comparing two heavily deteriorated versions of the same source video. Indeed, such an analysis is somehow far from the traditional field of use of these metrics [41], and to compare PSNR scores below 15dB can hardly suffice for a robust and reliable evaluation of our results.

Therefore, our evaluation will also rely on two other metrics proposed in [42], the Luminance Similarity Score (LSS) and the Edge Similarity Score (ESS), which are particularly adapted to partially encrypted content. LSS is a block-based luminance similarity measure based on the comparison of the average luminance values of 16x16 (or 8x8) non-overlapping blocks. The resulting score can be negative, and caps to 1 (in the case of two identical or *almost* identical images). On the other hand, ESS aims at measuring the degree of resemblance of edge and contour information. Its computation exploits the same block-partition as LSS, then implements a Sobel operator to extract a *dominant edge direction* per block. These edge directions are then compared to produce the ESS score that ranges from 0 to 1, where 1 indicates the highest level of similarity. Alongside with a precise description of these measures, [42] also provides thresholds LSS_{th} and ESS_{th} corresponding to heavily distorted contents are set to 0 and 0.5 respectively.

In Figure 9, we use these latter metrics to study the scrambling effect of the three different ES over the first 40 frames of BQMall sequence. As we already pointed out, *motion encryption* has no impact on I frames. Its impact on P frames increases due to the hierarchical configuration of the encoding, but remains quite limited (above the thresholds of 0 and 0.5 set in [42] to LSS and ESS respectively). In comparison, *texture* and *structure* encryption provide better results from both LSS and ESS metrics. If *texture encryption* slightly outperforms the proposed *structure encryption* considering the luminance based metric, the opposite effect is observed with the structural metric ESS. Plus, it is to highlight in this Figure that the impact of the proposed Intra prediction mode encryption is not limited to Intra frames. Indeed, the effect propagates naturally throughout the entire GOP due to the predictive coding strategy used in video encoders. In Figure 10, the frame per frame analysis is complemented by a comparison between the state of the art approach [15] and the extended scheme proposed in this paper. If both methods efficiently conceal the luminance information (results far under the 0 threshold in Figure 10a), our approach improves the structural deterioration measured by the ESS (fig. 10b).

We gather in Table III a comparative evaluation of the SE scrambling efficiency based on sequences used during HEVC standardization [36]. Numerical results tend to confirm that, on substantially all tested videos, the proposed encryption scheme offers a higher deterioration than the traditional bypass only encryption scheme. The increased structural impact is confirmed by both SSIM and ESS measures. More generally, since the four metrics - and particularly the LSS and the ESS - intend to measure the amount of information leakage through an attacked image, the proposed numerical analysis efficiently

supports the visual analysis performed in the previous section. Our method, by associating the three different domains of encryption, brings a notable improvement to the state-of-the-art approach relying on *motion* and *texture encryption* only.

2) Other consequences of encrypting Intra prediction modes: Apart from the visual impact on the video content, our approach exhibits some other noticeable characteristics.

First, since the encryption of CABAC regular-mode is performed in addition to the traditional by-pass only encryption, the ES is widened and the amount of *bins* encrypted per frame is increased. Second, the more regular-mode treated syntax elements are encrypted, the more their relative statistics are modified. As a direct result, there is a high risk that the probability model used during the table-based binary arithmetic coding does not describe efficiently the disturbed statistics. In practice, this side effect reduces the CABAC compression efficiency, causing the encrypted bitstream to be slightly bigger in size than its original version.

To measure both of these effects, a 2700 frames video stream is encoded from a concatenation of 9 CIF reference sequences (akiyo, mobile, bus, container, foreman, etc.). Doing so, we ensure that a wide variety of scene types are gathered (textures, object motion, camera motion, etc.). The results are presented in Table IV and Table V. A significant difference is observed between I and P/B frames. Indeed, the proposed Intra prediction encryption affects mostly I frames, from both the ES extension (+36% and +54% *bins* attacked) and the bitrate increase perspectives. In the case of P/B frames, *motion encryption* is included into the ES and the presence of Intra blocks diminishes, which explains the reduction of the relative impact of *structure encryption*. Average bitrate increase (+0.08% and +0.4%) shows that the induced overhead remains negligible, and perfectly viable in most use cases (e.g. a transport protocol should not need to repacketize stream chunks due to excessive sizes). More specifically, the maximum observed is +2% which ensures that no drastic bitrate modification is caused by the proposed extended scheme.

Tables IV and V also gather the delta PSNR and SSIM results obtained using the proposed Intra prediction encryption alone. Since we suggest in this paper to include this specific syntax element into the traditional ES, we have to ensure that their encryption is worth the subsequent bitrate increase. In [15], the impact of each cipherable HEVC codeword is analyzed independently, which allows to conduct a fair comparison. The major effect of Intra prediction encryption on the video content rendering is illustrated by delta PSNR of -24.35 and -26.25dB and delta SSIM of -0.299dB to -0.39dB. These results have to be put into perspective with the analysis conducted in [15] : the other impacting codewords like residual signs or delta QP cap respectively to -22dB and -25dB of delta PSNR. Moreover, the bitrate increase caused by their method of non-bitwise modifications (allowing larger variations of the cipherable codewords) is more important that the one our approach exhibits. From our study and the previous work [15], it thus appears that Luma prediction mode for Intra blocks/units is the most impacting codeword in the extended ES, and causes only very little bitrate increase.

We now discuss the effect of QP variations on the scrambling performance of the SE approach, illustrated by Table VI. When the QP increases, the compressed stream bitrate diminishes in parallel. Consequently, the ES shrinks and one could have concerns that there is not enough cipherable codewords for robust SE. We measured this effect on two very different sequences : Vidyo1 has only very limited movement and relies heavily on Intra prediction, whereas Kimono scene exhibits more motion information (camera tracking, moving textures). As expected for both streams, the ES diminishes when QP increases, causing the SE scrambling effect to globally weaken. More importantly, the relative impact of each encrypted *bin* increases when the ES is really small, which augments the variability of SE visual impact. Extending the ES as proposed in this paper allows to get better scrambling results over the whole tested QP range and, as a additional

QP	Vidyo1	Original	SoA [15]	Proposed
8	PSNR (dB)	44.3	10.2	**9.82**
	SSIM	0.98	0.52	**0.49**
	Encryption Space	-	47500	62500 (+30%)
16	PSNR (dB)	43.7	10.6	**9.90**
	SSIM	0.97	0.57	**0.52**
	Encryption Space	-	13950	20700 (+48%)
24	PSNR (dB)	42.5	11.1	**11.0**
	SSIM	0.96	0.57	**0.53**
	Encryption Space	-	3875	6250 (+61%)
32	PSNR (dB)	39.8	12.9	**11.2**
	SSIM	0.95	0.61	**0.55**
	Encryption Space	-	1125	2435 (+116%)
40	PSNR (dB)	36.0	14.2	**11.9**
	SSIM	0.92	0.70	**0.62**
	Encryption Space	-	357	1057 (+196%)
QP	**Kimono**	Original	SoA [15]	Proposed
8	PSNR (dB)	51.45	8.7	**8.28**
	SSIM	0.99	0.50	**0.47**
	Encryption Space	-	1060000	1181000 (+11%)
16	PSNR (dB)	44.4	11.0	**10.8**
	SSIM	0.96	0.50	**0.49**
	Encryption Space	-	171800	198240 (+15%)
24	PSNR (dB)	41.76	9.62	**9.23**
	SSIM	0.93	0.49	**0.48**
	Encryption Space	-	42690	44700 (+4%)
32	PSNR (dB)	37.7	14.0	**11.9**
	SSIM	0.90	0.56	**0.53**
	Encryption Space	-	10550	11400 (+8%)
40	PSNR (dB)	33.47	10.8	**9.82**
	SSIM	0.86	0.57	**0.53**
	Encryption Space	-	2280	2800 (+22%)

TABLE VI: Effect of QP variation on the SE efficiency on two different sequences. *Top:* Vidyo1 (HEVC, 1280x720, 200 frames); *Bot:* Kimono (HEVC, 1920x1080, 200 frames). The state of the art approach of [15] is denoted as SoA. The encryption space is measured as the average number of encrypted *bins* per frame.

advantage, reduces the results uncertainty for the highest QPs.

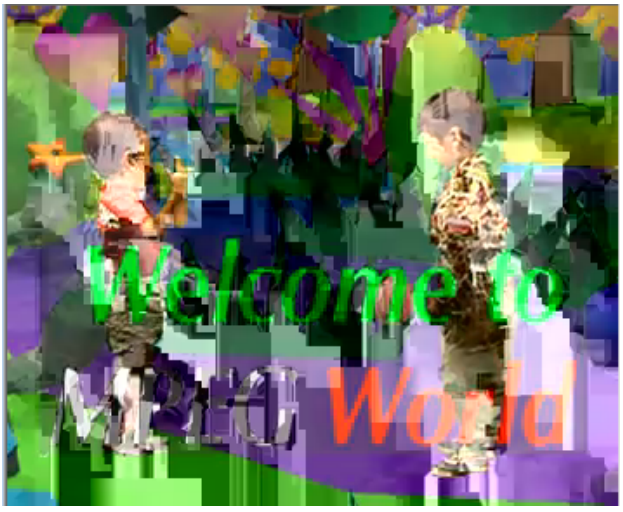
Finally, the impact of SE is evaluated in terms of complexity. Since the encryption process is embedded inside the encoder/decoder, the processing time related to the computation of AES CTR block cipher is added to the standard video processing. Considering our implementation of SE as a symmetric transcoder, we cannot provide explicit results on the relative complexity of SE compared to the encoding step. However, we can refer to [25] and [30] where required processing powers are evaluated for H.264/AVC and HEVC sequences respectively. The orders of magnitude are consistent between both standard: the encoding step requires less than 0.5% additional processing time, whereas SE impacts the decoding with an additional 2% to 3% time. This distinction arises from the major complexity difference between encoding and decoding processes in H.264/AVC and HEVC standards. To correlate these results with our symmetric transcoder implementation, we re-use the 2700 frame CIF sequence presented previously and perform its transcoding with HEVC. When the standard ES is used (i.e. Intra prediction modes are untouched), the transcoding time increases by 1.94% (from 15.21 to 15.50 seconds). This is coherent with results of [25] and [30] since the complexity of the transcoder is slightly higher than the one of a decoder, due to the necessary CABAC re-encoding. When Intra prediction modes are included into the ES, an additional 1.28% of processing time is needed (from 15.50 to 15.71 seconds). Such a result remains a negligible increase in processing power, and can be managed well even by hand-held devices.

D. Cabac regular mode statistical attack

From a security perspective, SE exhibits very different characteristics than standard encryption algorithms. Indeed, a stream that has been fully or partly (e.g.,with headers preserved) protected by raw encryption is not format compliant anymore. Since the structure of compressed streams is highly specific, attacks can try to exploit format priors (presence of start codes, symbols validity, etc.) to help the brute force search. In contrast, SE is more a scrambling technique based on valid stream modifications, which prevents the attacker from having any prior information on the protected stream. The only possible angle of attack is a computer vision assisted unscrambling approach which aim at recovering maximum intelligibility. Moreover, the lack of prior information prevents any attack from converging surely to the original stream. In such a context, our extended approach that increases the scrambling performance - hardening computer vision recovery - and the number of protected *bins* - widening the search domain - brings an improvement to the overall robustness of the SE protection.

Unscrambling methods can be based on known-plaintext - or replacement - attacks as presented in [30]. It consists in setting to a predefined value all cipherable *bins* in order to limit the scrambling effect of SE. Such an approach could possibly be efficient to attack the proposed Luma Intra prediction mode encryption. Indeed, codewords encoded with CABAC regular mode rely on probability models, providing statistical

(a) Intra prediction encryption visual result (PSNR encrypted: 15.43dB).



(b) MPS-based recovery (PSNR attacked: 15.28dB).

Fig. 11: Effect of *structure* encryption (scrambling of Intra prediction modes) on one frame of Mix_sequence (2700 frames, CIF, PSNR original: 41.3dB) and visual result of the MPS-based replacement attack.

information that can be exploited by an attacker to recover the modified bins. Namely, a Most Probable Symbol (MPS) which depends on the current context is available for each bin encoded in regular mode. In this section, we thus evaluate the performance of an unscrambling replacement approach relying on the MPS to recover the encrypted Luma prediction modes.

In H.264/AVC, a 3-bins codeword - designated as *rem_intra_pred_mode* - allows to infer which prediction mode amongst the nine available is used by the current block. This codeword is fully encoded with CABAC regular mode, which allows to envision a statistical attack. In HEVC, the encoding of Intra prediction modes has been modified and the arithmetic coding now relies on CABAC by-pass mode to encode the corresponding bins. Their encryption do impact - through the residual encoding - the CABAC regular-mode, but this cannot be exploited to conduct an unscrambling MPS-based approach. The attack performed in this section is thus only applicable

	Protected stream
Nb. Intra blocks	690 192
Nb. separately coded *	324 581 (47%)
Nb. cipherable modes	291 065 (42%)
Nb. cipherable *bins* (3/mode)	873 195
	Attacked stream
Nb. protected *bins*	873 195
Nb. *bins* correctly derived from MPS	418 959 (47.98%)
Nb. protected modes	291 065
Nb modes recovered	29 980 (10.3%)

TABLE VII: Intra prediction mode statistics in the protected Mix_sequence stream, and results of the MPS-based replacement attack. **: An Intra block is separately coded if it uses the 3-bin rem_intra_pred_mode to infer the prediction mode.*

to H.264/AVC encoded streams. Note that we suppose that the attacker exactly knows which blocks are considered as cipherable and which bins are encrypted.

The MPS-based attack on Intra prediction modes is conducted on the 2700 frames Mix_sequence presented in the previous section, protected using the proposed *structure encryption*. Thus, to show the robustness of our method, we propose in this paragraph to only encrypt one codeword: the *rem_intra_pred_mode* specifying the intra mode when it is not derived from the neighborhood. The visual result of encrypting only this codeword is depicted in Figure 11 and the corresponding statistics are presented in the first part of Table VII.

The attack simply consists in replacing every *bin* by its corresponding MPS, which is accessible in the CABAC engine during decoding. The visual result of this replacement attack is illustrated in Figure 11b and the corresponding statistics are summarized in Table VII. The MPS-based recovery correctly deciphers 47.98% of the encrypted bins, and solely 10.3% of the protected prediction modes are recovered. This does not allow to efficiently unscramble the video content (fig. 11b), and the attacked stream even has a lowered PSNR (-0.15dB). Thus, using the MPS as a prior for replacement attacks does not seem reliable enough, and cannot be easily exploited by attackers to efficiently unscramble the protected contents.

More generally, such an analysis questions the interest of using a context modeling stage in the encoding of the Intra prediction modes. Indeed, in terms of compression efficiency, CABAC regular mode outperforms the by-pass mode only if a majority of the encoded bins are equal to their corresponding MPS. Encoding a codeword which is badly predicted by a context model results in a decreased compression efficiency and an increased computational cost. We evaluated the pertinence of the MPS representation of the *rem_intra_pred_mode* codeword on almost 1 million *bins* of the original Mix_sequence stream: only 52.4% are equal to their respective MPS, which tend to confirm that the model representation is hardly efficient for encoding the Intra prediction modes.

We showed in this security analysis that H.264/AVC MPS-based replacement attacks do not allow to efficiently unscramble protected streams because of the unpredictable probability distribution of the *rem_intra_pred_mode* codeword, which is further randomized by the encryption process. Eventually, such a result can be put in perspective with two other claims: first, the simplification of this step in HEVC (related codewords now encoded in by-pass mode) is justified by the inability to effectively predict the corresponding bins with a context model. Second, the fact that Intra prediction encryption only causes very little bitrate increase is confirmed: since the probability model used cannot really exploit the symbol statistics to improve the compression efficiency, the modification of these statistics do not drastically disrupt the compression capabilities.

V. Conclusion and future work

This paper proposes a novel method for SE of H.264/AVC (CABAC) and HEVC compressed streams. Our approach tackles the main security challenge of SE: the limitation of the information leakage through protected video streams.

The encryption of Luma prediction modes for Intra blocks/units is proposed in addition to the traditional ES in order to significantly improve the *structural* deterioration of video contents. Especially on I frames, it brings a solid complement to previous SE schemes relying on residual encryption (mainly affecting the *texture* of objects) and motion vector encryption (disrupting the *movement* of objects). The proposed scheme shows both numerical and visual improvements of the scrambling performance regarding state-of-the-art SE schemes. Based on the encryption of *regular*-mode treated syntax element, the approach exhibits a loss in compression efficiency that has been estimated empirically up to 2% on I frames, which remains, in a lot of application cases, a totally tolerable variation.

In this paper, we complemented the state of the art analysis proposed in [15] of the impact of each cipherable syntax element in HEVC standard with the proposed Intra prediction encryption. The study confirms the premise that targeting this specific element is particularly efficient from the SE perspective: the PSNR and SSIM variations we obtain tend to demonstrate that it is the most impacting codeword in the proposed ES. Considering their role in the video reconstruction, such a result is not really surprising: their encryption heavily impacts Intra frames but also Intra blocks/units in Inter frames, and the generated error naturally drifts both spatially and temporally within the video stream.

We did not provide in this paper an exhaustive analysis of CABAC regular mode cipherable syntax elements. Some of them are easily identifiable (for example, the process proposed in this paper could almost identically apply to the Intra prediction modes for Chroma components), but a large majority of these cipherable elements would require a very specific monitoring which we did not address in our study. An analysis of the *maximum* ES for both H.264/AVC and HEVC, based on the proposed *regular* mode encryption, is thus one major track for future research. Nevertheless, such a study would have to pay a particular attention to a critical trade-off we highlighted in our study; the underlying correlation between the improvement of the scrambling performance and its consequences on the compression efficiency.

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, July 2003.

[2] J. Ohm and G. Sullivan, "High efficiency video coding: the next frontier in video compression [standards in a nutshell]," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 152–158, Jan 2013.

[3] A. Said, "Measuring the strength of partial encryption schemes," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, Sept 2005, pp. II–1126–9.

[4] B. Boyadjis, C. Bergeron, and S. Lecomte, "Auto-synchronized selective encryption of video contents for an improved transmission robustness over error-prone channels," in *IEEE ICIP Proceedings (accepted)*, September 2015.

[5] B. Boyadjis, M.-E. Perrin, C. Bergeron, and S. Lecomte, "A real-time ciphering transcoder for H.264 and HEVC streams," in *Image Processing (ICIP), 2014 IEEE International Conference on*, Oct 2014, pp. 3432–3434.

[6] C. Bergeron and C. Lamy-Bergot, "Complaint selective encryption for H.264/AVC video streams," in *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, Oct 2005, pp. 1–4.

[7] L. Dubois, W. Puech, and J. Blanc-Talon, "Smart selective encryption of CAVLC for H.264/AVC video," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*, Nov 2011, pp. 1–6.

[8] N. Khlif, T. Damak, F. Kammoun, and N. Masmoudi, "Selective encryption of CAVLC for H.264/AVC," in *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2013 14th International Conference on*, Dec 2013, pp. 314–317.

[9] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[10] B. Bross, W.-J. Han, J.-R. Ohm, G. Sullivan, and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 8," in *JCTVC-J1003*, Stockholm, Sweden, July 2012.

[11] V. Sze and M. Budagavi, "A comparison of CABAC throughput for HEVC/H.265 VS. AVC/H.264," in *Signal Processing Systems (SiPS), 2013 IEEE Workshop on*, Oct 2013, pp. 165–170.

[12] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, no. 28, pp. 656–715, 1949.

[13] *Advanced Encryption standard (AES)*, FIPS Publication 197 Std., Nov. 2001.

[14] Crypto++, "Speed benchmarks for some of the most commonly used cryptographic algorithms," http://www.cryptopp.com/benchmarks.html, Jun. 20015.

[15] 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," *Consumer Electronics, IEEE Transactions on*, vol. 59, no. 3, pp. 634–642, August 2013.

[16] M. Farajallah, w. hamidouche, S. E. A. Deforges, and O., "ROI selective video encryption in the HEVC video standard," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[17] S.-K. A. Yeung, S. Zhu, and B. Zeng, "Design of new unitary transforms for perceptual video encryption," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 9, pp. 1341–1345, Sept 2011.

[18] ——, "Perceptual video encryption using multiple 8x8 transforms in H.264 and MPEG-4," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 2436–2439.

[19] C.-P. Wu and C.-C. Kuo, "Design of integrated multimedia compression and encryption systems," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 828–839, Oct 2005.

[20] G. Jakimoski and K. Subbalakshmi, "Cryptanalysis of some multimedia encryption schemes," *Multimedia, IEEE Transactions on*, vol. 10, no. 3, pp. 330–338, April 2008.

[21] J. Wen, H. Kim, and J. Villasenor, "Binary arithmetic coding with key-based interval splitting," *Signal Processing Letters, IEEE*, vol. 13, no. 2, pp. 69–72, Feb 2006.

[22] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *Multimedia, IEEE Transactions on*, vol. 8, no. 5, pp. 905–917, Oct 2006.

[23] P. Carillo, H. Kalva, and S. Magliveras, "Compression independant reversible encryption for privacy in video surveillance," *EURASIP J. Inf. Security*, vol. 2009, no. 5, Jan. 2009.

[24] Y. Wang, M. O'Neill, and F. Kurugollu, "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 9, pp. 1476–1490, Sept 2013.

[25] 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," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 565–576, May 2011.

[26] M. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of H.264/SVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 3, pp. 425–437, March 2013.

[27] F. Liang, X. Peng, and J. Xu, "Scene-aware perceptual video coding," in *Visual Communications and Image Processing (VCIP), 2013*, Nov 2013, pp. 1–6.

[28] L. Tong, F. Dai, Y. Zhang, and J. Li, "Restricted H.264/AVC video coding for privacy region scrambling," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 2089–2092.

[29] F. Dufaux and T. Ebrahimi, "H.264/AVC video scrambling for privacy protection," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 1688–1691.

[30] Z. Shahid and W. Puech, "Visual protection of HEVC video by selective encryption of CABAC binstrings," *Multimedia, IEEE Transactions on*, vol. 16, no. 1, pp. 24–36, Jan 2014.

[31] H. Hofbauer, A. Uhl, and A. Unterweger, "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 1986–1990.

[32] J. Jiang, S. Xing, and M. Qi, "An intra prediction mode-based video encryption algorithm in H.264," in *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on*, vol. 1, Nov 2009, pp. 478–482.

[33] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 620–636, July 2003.

[34] H. Isa, I. Bahari, H. Sufian, and M.-R. Zaba, "AES: current security and efficiency analysis of its alternatives," in *IEEE international conf. on IAS*, Melacca, Malaysia, Dec. 2011.

[35] S. Gueron and V. Krasnov, "Speeding up counter mode in software and hardware," in *11th International Conference on ITNG*, Las Vegas, NV, United States, April 2014.

[36] V. Baroncini, J.-R. Ohm, and G. Sullivan, "Report of subjective test results of responses to the joint call for proposal on video coding technology for High Efficiency Video Coding," in *Tech. Rep. JCT-VC*, Geneva, Switzerland, 2010.

[37] B. Boyadjis, "Image and video processing," https://webperso.telecom-paristech.fr/boyadjis.html.

[38] I. Pitas and A. Venetsanopoulos, "Edge detectors based on nonlinear filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 4, pp. 538–550, July 1986.

[39] P. Hanhart and T. Ebrahimi, "Calculation of average coding efficiency based on subjective quality scores," *Journal of Visual communication and image representation*, vol. 25, no. 3, pp. 555–564, 2014.

[40] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryption," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 3, pp. 325–339, March 2012.

[41] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, June 2008.

[42] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2061–2075, July 2006.