IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 31, 2021, accepted February 3, 2022, date of publication February 7, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149599

Selective Encryption of the Versatile Video Coding Standard

MOUSA FARAJALLAH[1], GUILLAUME GAUTIER[2], WASSIM HAMIDOUCHE[2], OLIVIER DÉFORGES[2], AND SAFWAN EL ASSAD[3]

[1]College of Information Technology and Computer Engineering, Palestine Polytechnic University, Hebron 117310, Palestine
[2]Univ. Rennes, INSA Rennes, CNRS, IETR—UMR 6164, 5700 Rennes, France
[3]IETR, Polytech Nantes, 44300 Nantes, France

Corresponding author: Mousa Farajallah (mousa_math@ppu.edu)

ABSTRACT versatile video coding (VVC) is the next generation video coding standard developed by the joint video experts team (JVET) and released in July 2020. VVC introduces several new coding tools providing a significant coding gain over the high efficiency video coding (HEVC) standard. It is well known that increasing the coding efficiency adds more dependencies in the video bitstream making format-compliant encryption with the standard more challenging. In this paper we tackle the problem of selective encryption of the VVC standard in format-compliant and constant bitrate. These two constraints ensure that the encrypted bitstream can be decoded by any VVC decoder while the bitrate remains unchanged by the encryption. The selective encryption of all possible VVC syntax elements is investigated. A new algorithm is proposed to encrypt in format-compliant and constant bitrate the transform coefficients (TCs) together with other syntax elements at the level of the entropy encoder. The proposed solution was integrated and assessed under the VVC reference software model version 6.0. Experimental results showed that encryption drastically decreases the video quality while the encryption is robust against several types of attacks. The encryption space is estimated in the range of 15% to 26% of the bitstream size resulting in a lightweight encryption process. The web page of this work is publicly available at https://gugautie.github.io/sevvc/.

INDEX TERMS Versatile video coding, joint crypto-compression, selective encryption, video security.

I. INTRODUCTION

Security and confidentiality of multimedia contents are of prominent importance in many applications to ensure the safe storage and transmission of images and videos. The straightforward solution to perform secure transmission of a video is to encrypt the whole video bitstream with a secure encryption protocol such as advanced encryption standard (AES) [1]. However, this solution when applied to the video has several limitations related to their high computational complexity increasing both the energy footprint and end-to-end latency. This increase in complexity/latency is mainly caused by the processing complexity of the encryption algorithm used to cipher the whole video especially when the video is encoded at a high bitrate. Moreover, the deciphering and ciphering processes are required to perform post-processing operations such as transcoding for network adaptation. This may harm security since the secret key is shared with un untrusted middlebox in the network to perform splicing, quality monitoring, watermarking, and transcoding. The selective encryption solution has emerged as an effective alternative to perform secure and low complexity encryption of images and videos. The encryption process is performed in the compressed-domain where only a set of the most sensitive information is encrypted. This enables performing both format-compliant and constant bitrate encryption. The format-compliant property is very important enabling to decode the video bitstream without deciphering and thus all post-processing operations can be performed including packaging and transcoding without requiring access to the secret key used for encryption. Moreover, this property enables encrypting only some spatial regions in the image identified as region of interest (ROI) while keeping the rest of the image clear. The constant bitrate property preserves

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

the encoder coding efficiency. Selective encryption has been widely investigated for different still image and video coding standards including JPEG [2], JPEG-2000 [3], [4], advanced video coding (AVC) [5], scalable video coding (SVC) [6], and more recently HEVC [7]–[9] and its scalable extension SHVC [10]. Selective encryption of the HEVC standard has been widely investigated in the literature [11]–[14] enabling format-compliant, secure and low complexity encryption.

The ISO/moving picture experts group (MPEG) and ITU/video coding experts group (VCEG) developed the next generation video coding standard called VVC [15], [16]. This latter, released in July 2020, introduces new coding tools outperforming HEVC by up to 50% in terms of bitrate reduction for a similar visual quality [17], [18]. To the best of our knowledge, format-compliant and constant bitrate encryption of VVC has not yet been addressed. Moreover, it is well known from information theory [19] that enhancing the coding efficiency adds more dependencies in the bitstream making format-compliant and constant bitrate encryption more challenging.

This paper investigates a format-compliant and constant bitrate encryption of a video bitstream encoded with the VVC standard. To meet these two constraints, the encryption is performed at the level of the context-adaptive binary arithmetic coding (CABAC) engine. We first investigate all possible syntax elements that can be encrypted in both format-compliant and constant bitrate. A set of VVC syntax elements including TC values and signs, chroma prediction candidate, motion vector (MV) differences and signs are encrypted. We propose a new algorithm that determines the encryptable bins within the TCs. The proposed selective encryption solution has been extensively assessed under the VVC common test conditions (CTCs) using three image and video quality assessment metrics including peak signal to noise ratio (PSNR), structural similarity (SSIM), and video multimethod assessment fusion (VMAF), and security metrics such as encryption quality (EQ) [20], histogram analysis, edge detection and edge differential ratio (EDR) [21]. The proposed solution has also been tested against brute force attack, number of pixels change rate (NPCR) and unified average changing intensity (UACI) [22]. The encryption space giving the percentage of encrypted bits in the bitstream varies in the range of 15% to 26% for different targeted bitrates. This low encryption space results in a very low deciphering complexity which remains lower than 6% of the decoding time.

The rest of this paper is organized as follows. Section II gives a brief review on selective encryption solutions proposed for HEVC and then Section III describes the entropy coding of syntax elements in VVC. Section IV presents the proposed solution to encrypt VVC syntax elements in format-compliant and constant bitrate. The performance of the selective encryption solution is assessed in Section V in terms of video quality degradation, resilience to different attacks and complexity overhead. Finally, Section VI concludes this paper.

II. RELATED WORKS

In this section, we review the existing solutions for HEVC standard encryption. The first format-compliant encryption solution of HEVC was proposed by Shahid *et al.* [7]. In this solution, AES was used in cipher feedback (CFB) mode to perform selective encryption of the selected syntax elements at the CABAC stage. This work considered an earlier version of the HEVC standard and some encryptable syntax elements of HEVC were not identified in this solution. Farajallah *et al.* [8] proposed a selective encryption solution to cipher the ROI in HEVC standard. This solution relies on the tile concept introduced in HEVC enabling a frame partitioning into independent rectangular regions. The encryption process encrypts only tiles within the ROI and keeps the background clear. The tiles within the ROI are encrypted in format-compliant and constant bitrate by ciphering only a set of sensitive syntax elements. Moreover, to prevent the encryption propagation outside the ROI, MVs of the background tiles are constrained to only refer to background area (no ROI) in the reference frames. This solution has been implemented in [23] under the kvazaar HEVC software encoder and the OpenHEVC [24] decoder for a real time end-to-end ROI encryption system relying on the HEVC standard and format-compliant selective encryption. Boyadjis *et al.* [9] presented a selective encryption algorithm in order to increase the visual distortion. The presented research moves selective encryption from bypass mode to regular mode, which negatively affects the bitrate. Luma intra prediction modes are selected to be encrypted in addition to the residuals. The presented solution enables more scrambling performance while the compression efficiency has changed leading a slight bitrate increase. Hamidouche *et al.* [10] investigated a selective encryption of the final version of HEVC. The authors have proposed a real time selective encryption solution for the scalable extension of HEVC named scalable HEVC (SHVC). The presented solution has analyzed all SHVC syntax elements in order to perform format-compliant, constant bitrate and low latency encryption while preserving all SHVC features. The presented results showed the high security level of the selective encryption solution with a low complexity overhead below 6% of the decoder complexity. Van Wallendael *et al.* [11] presented a format-compliant selective encryption solution for HEVC. They selected a set of syntax elements from HEVC that preserve the format compliance. Several techniques to selectively encrypt the video are investigated. The obtained result showed that most of the selected syntax elements have a low effect on the rate-distortion performance while having a broad range in scrambling performance. Memos *et al.* [12] presented an algorithm that encrypts only Intra (I) frames of the HEVC bitstream based on the idea that predicted (P) and bidirectional predicted (B) frames are useless without I frame. Moreover, encrypting only I frames will decrease the encryption time by 50% and propagates the encryption to other frames. The presented algorithm merged two algorithms proposed in [25], [26], while introducing some modifications to the selection

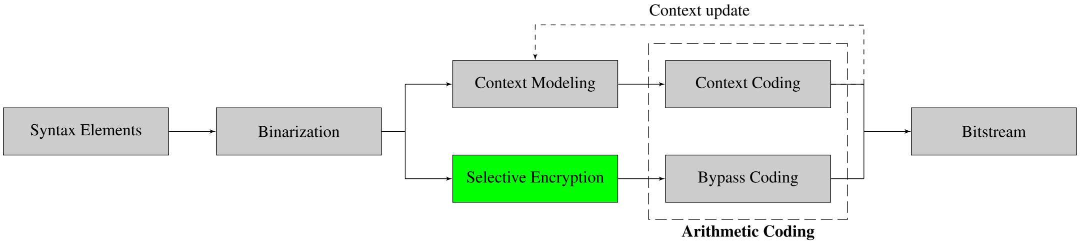


FIGURE 1. Overall architecture of the CABAC engine in VVC. The selective encryption block is illustrated in green.

and management of the encrypted data to be amendable to HEVC. This work relies on the AES algorithm for secure transmission of HEVC bitstream with 256 bits as key length. It collects sign bits of each transform coefficient of I frames until the collected signs reached 256 bits. However, it is not clear in the proposed algorithm whether the collected bits are used as key value or as state value since AES-256 state size is 128 and not 256. The proposed algorithm performs conventional AES encryption on the collected bits and swap the original sign bits by the encrypted ones. Finally, the Shamir's secret sharing (SSS) input parameters are collected from the non-zero alternating current (AC) coefficients of each transform block within the I frame. It is clear from the description that the proposed algorithm performs partial encryption algorithm. It is also important to note that the proposed solution is not format-compliant, none constant bitrate since it increases the bitstream size at least by 8%. Long *et al.* [13] presented a format-compliant encryption in order to secure HEVC streams in multimedia social networks. The presented algorithm is tightly integrated with the encoding/decoding processes. The presented work performs encryption in two steps. First, a stream cipher is used to encrypt sign of the nonzero TCs, and the first sign bit hiding of TCs. Second, based on a control factor, only one parameter from merging index, MV prediction index, sign of MV difference and reference frame index is encrypted. The presented research increases the bitrate, while it is format-compliant solution. Finally, the presented work was assessed regarding security and complexity which confirms the good security level and acceptable complexity overhead. Ahmed *et al.* [14] presented a new solution for efficient selective encryption based on the chaotic logistic map for HEVC. The presented solution encrypts the sign bit of the MV differences and the TCs. The encryption process is performed at the entropy coding stage of the HEVC encoding process. They focused on achieving a low complexity ciphering targeting real time applications, constant bitrate and format-compliant encryption. The presented work was compared with the solution proposed in [11] and the obtained results confirm the suitability for real time applications with an intermediate level of security. Peng *et al.* [27] presented a tunable selective encryption scheme for HEVC based on chroma Intra prediction mode (IPM) and TCs scrambling. The presented work has two security levels. The first one encrypts HEVC syntax elements including Luma IPMs, Chroma IPMs, the suffix part of the TCs, sign and value of the MV differences, merge index, advanced MV prediction, reference frame index, and sample adaptive offset (SAO) filter parameters. The second security level relies on edge extraction of each transform block. The transform block coefficients are scrambled to increase the security level only when the current transform block contains edges. Finally, the AES is used in CounTeR (CTR) mode in order to generate the pseudo-random number sequences. These sequences are used to encrypt all previously mentioned parameters with a simple eXclusive OR (XOR) operation. Xu [28] proposed to perform data hiding inside the selected encrypted bitstream of HEVC. The secret message is hidden using a quantized transform coefficient (QTC) modification technique. It only changes bits value based on the data hiding without changing the data size in bypass coding, which confirms that the obtained solution is constant bitrate and format-compliant. Since the used operation is an XOR, the extraction process of the hiding data can be achieved on both encrypted as well as original videos. Obtained results confirm the resilience of the presented work against replacement attacks. Moreover, the degradation on the video quality introduced by data hiding is negligible. However, the presented algorithm was not evaluated regarding important general video attacks such as UACI, NPCR, EDR, EQ, histogram analysis and key sensitivity attacks.

III. CABAC ENGINE IN VVC

The CABAC engine defined in VVC is similar to HEVC consisting of three main functions: binarization, context modeling and arithmetic coding [30]. The overall CABAC architecture is illustrated in Fig. 1. First, the binarization step converts syntax elements to binary symbols (bins). Second, the context modeling updates the probabilities of bins, and finally, the arithmetic coding compresses the bins into bits according to the estimated probabilities.

A. BINARIZATION METHODS

Six binarization methods are used in VVC, namely unary (U), truncated unary (TU), fixed length (FL), truncated binary (TB), truncated rice code with context p (TRp) and Exp-Golomb k-th order code (EGk). The U code represents an unsigned integer B with a binstring of length $B+1$ composed

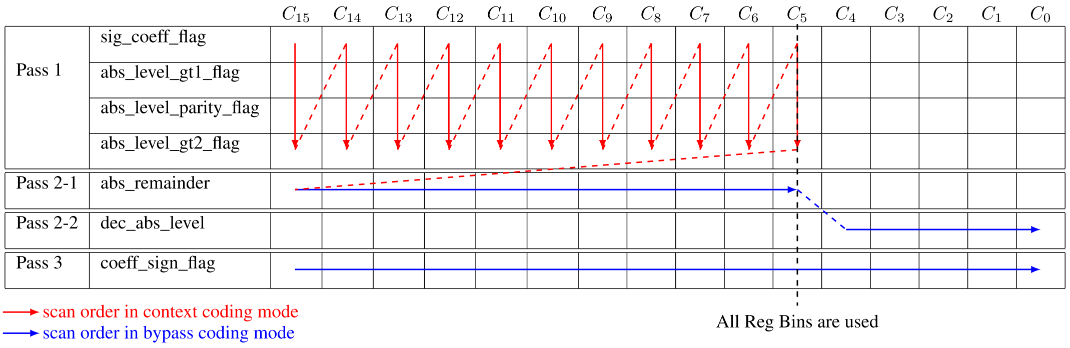


FIGURE 2. **Binarization of the transform coefficients (TCs) of a 4 × 4 sub-block in TC mode JVET-S2002 [29].**

of B 1-bins followed by one 0-bin. The TU code is defined with the largest possible value of the syntax element *cMax* ($0 \leq B \leq cMax$). When the syntax element value $B < cMax$, the TU is equivalent to U code, otherwise B is represented by a binstring of *cMax* 1-bins. The FL code represents a syntax element B with its binary representation of length $\lceil \log_2(cMax+1) \rceil$ with $\lceil x \rceil$ is the smallest integer greater than or equal to x. The TB code is similar to the FL code, except when the $cMax+1$ value is not a power of 2. In this case, let k be $k = \lfloor \log_2(cMax+1) \rfloor$ (with $\lfloor x \rfloor$ is the largest integer less than or equal to x). The first $u = 2^{k+1} - cMax$ elements are coded with a FL code of length k. The remaining $cMax+1-u$ symbols are shifted by u and coded by $k+1$ bins. The TRp code is a concatenation of a quotient $q = \lfloor B/2^p \rfloor$ and a remainder $r = B - q2^p$. The quotient q is first represented by the TU code as a prefix concatenated with a suffix r represented by the FL code of length p. The EGk code is also a concatenation of prefix and suffix. The prefix part of the EGk code is the U representation of $l(B) = \lfloor \log_2(\frac{B}{2^k}+1) \rfloor$. The suffix part is the FL code of $B + 2^k(1 - 2^{l(B)})$ with $cMax = k + l(B)$.

B. TRANSFORM COEFFICIENTS CODING

In this section, we describe the CABAC coding of the TCs. Similar to HEVC, VVC coefficients are either coded in regular TC mode or transform skip (TS) mode. In both modes, the transform block is first divided into sub-blocks.

1) VVC TC CODING MODE

The coefficients of each sub-block are encoded in three passes as illustrated in Fig. 2 for a 4 × 4 sub-block. The coefficients are processed in reverse diagonal scan order, as depicted in Fig. 3a. The first pass processes a group of flags until it reaches a limit of used bins specified by the standard. This maximum number of bins used in the first pass is computed with respect to the block size ($W_b \times H_b$) as follows $\lfloor (2^{\log_2(W_b)+\log_2(H_b)})\, 7/4 \rfloor$. Once this limit is reached, the second pass starts encoding the remainders computed from the coefficient value C as follows

$$abs\_remainder = \begin{cases} \left\lfloor \dfrac{|C|-4}{2} \right\rfloor & |C| \geq 4, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The coefficients of value lower than 4 are binarized by the flags in the first pass. The second pass relies on the TRp/EGk binarization, until the position of the last coefficient processed by the first pass is reached. Then, the *dec_abs_level* syntax element, computed by (2) for the remaining coefficients is bypassed and binarized also using the TRp/EGk binarization.

$$dec\_abs\_level = \begin{cases} V, & \text{if } C = 0, \\ |C|, & \text{if } |C| \leq V, \\ |C|+1, & \text{if } |C| > V, \end{cases} \quad (2)$$

the V constant is derived from a lookup table (LUT) *VArr* according to the state and the local absolute sum *LocAbsSum* computed for the current coefficient by (3). The V value updates the 0 coefficient value such that coefficients have smaller binarization when large coefficients are mixed with definite 0 values. The *LocAbsSum* is a saturated sum in the interval $[0, 31]$ of a set of neighboring coefficients S_1 illustrated in green in Fig. 4a

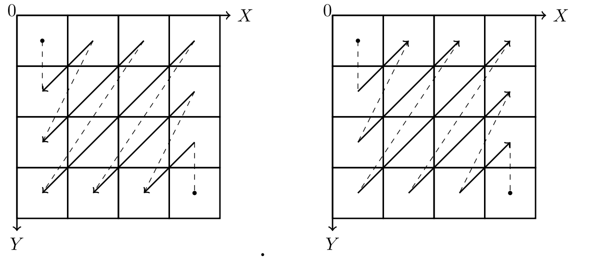
$$LocAbsSum = \left[\sum_{i \in S_1} |C_i| - 5\, BaseLvl \right]_0^{31}, \quad (3)$$

where *BaseLvl* is equal to 4 for the *abs_remainder* (Pass 2-1) and 0 for the *dec_abs_level* (Pass 2-2).

Finally, the third pass encodes the signs of the coefficients. We can notice that only the first pass relies on CABAC context coding and the last two passes perform bypass coding. The *abs_remainder* and *dec_abs_level* syntax elements are both binarized by a combination of TRp and in a special case EGk code. This binarization is presented in Section III-B3.

2) VVC TS CODING MODE

In transform skip mode, the coefficients of each sub-block are also encoded in three passes that process the coefficients in a simple diagonal scan order, as shown in Fig. 3b. The first pass mainly encodes all coefficients considered as significant (ie. $C \neq 0$) including its sign and parity. The second pass encodes more flags to check whether the coefficient is greater than a certain threshold. Finally, the third pass encodes the



(a) (b)

FIGURE 3. TCs scanning orders (3a) reverse diagonal scan order and (3b) regular diagonal scan order.

remainder coefficients greater than 10 using the TRp/EGk binarization of *abs_remainderTS*

$$abs\_remainderTS = \left\lfloor \frac{|C| - 10}{2} \right\rfloor. \quad (4)$$

The local absolute sum in the case of TS mode *LocAbsSumTS* is computed by (5) as follows

$$LocAbsSumTS = \left[\sum_{i \in S_2} |C_i| \right]_0^{31}. \quad (5)$$

It should be noted that the third pass relies on bypass coding.

3) BINARIZATION PROCESS

Algorithm 1 gives the binarization process of the *abs_remainder*. The *dec_abs_level* and *abs_remainderTS* syntax elements are also binarized by this algorithm. The TCs remainders are binarized using either a TRp code, introduced in Section III-A, or an EGk code limiting the maximal length of a binarization to 32 bits as presented in Algorithm 2. The selection between the two binarizations depends on a threshold value β defined in the standard as

$$\beta = BinReduc\ 2^{cRiceParam}, \quad (6)$$

where *BinReduc* is set to 5, and $cRiceParam \in \{0, 1, 2, 3\}$ is the rice parameter derived from a LUT *riceArr* according to the saturated local absolute sum *LocSumAbs* of previously coded coefficients computed by (3). Fig. 4a illustrates in green the set of coefficients S_1 used to derive the rice parameter of the current coefficient highlighted in yellow. Similarly, Fig. 4b presents the coefficients used in TS mode, where the rice parameter depends only on the top and left neighbor coefficients set S_2. When the remainder to encode is strictly below the threshold β, the TRp binarization is performed with $p = cRiceParam$. Otherwise, the limited EGk coding is applied.

Algorithm 2 shows that the maximum length of the prefix *maxPrefixLen* depends on the range of the transform coefficients $2^{log2TrRange}$ and *BinReduc*. To differentiate between the two binarizations at the decoder side, *BinReduc* is added to the prefix length when *Limited_EGk* is used. Then, a classical EGk binarization starts. However, if the computed prefix length *prefixLen* is equal to the maximal prefix length *maxPrefixLen*, the suffix length *suffixLen* is set to *log2TrRange*. Both codes are composed of a variable-length *prefix* and if exists, a fixed-length *suffix*. The prefix is coded using a U or TU code representation which implies that changing any bin will violate the decoder standard or change the bitrate. On the other hand, the suffix might be encrypted in format compliance and constant bitrate only when the *LocSumAbs* does not change the *cRiceParam* value of the neighbor coefficients.

Algorithm 1 *abs_remainder* Binarization

```
Input: $abs\_remainder$ is the unsigned integer to binarize
    $cRiceParam$ is the rice parameter
    $log2TrRange$ is the $log2$ of the TC range
    $BinReduc \leftarrow 5$ is the value used to determine the threshold
    between TRp and $Limited\_EGk$

    $\beta \leftarrow BinReduc\ 2^{cRiceParam}$
    if $abs\_remainder < \beta$ then
        TRp binarization with $p \leftarrow cRiceParam$
    else
        $Limited\_EGk(abs\_remainder, cRiceParam,$
            $BinReduc, log2TrRange)$
    end if
```

Algorithm 2 *Limited_EGk*(*abs_remainder*, *cRiceParam*, *BinReduc*, *log2TrRange*)

```
Input: $abs\_remainder$, $cRiceParam$, $log2TrRange$,
    $BinReduc$.

    $maxPrefixLen \leftarrow 32 - BinReduc - log2TrRange$
    $codeValue \leftarrow \left\lfloor \frac{abs\_remainder}{2^{cRiceParam}} \right\rfloor - BinReduc$
    $prefixLen \leftarrow 0$
    while $prefixLen < maxPrefixLen$
    and $codeValue > 2^{prefixLen+1} - 2$ do
        $prefixLen \leftarrow prefixLen + 1$
    end while
    if $prefixLen = maxPrefixLen$ then
        $suffixLen \leftarrow log2TrRange$
    else
        $suffixLen \leftarrow prefixLen + cRiceParam + 1$
    end if
    $totalPrefixLen \leftarrow prefixLen + BinReduc$
    $bitMask \leftarrow 2^{cRiceParam}$
    $prefix \leftarrow 2^{totalPrefixLen} - 1$
    $suffix \leftarrow codeValue - 2^{prefixLen} - 1$
    $suffix \leftarrow suffix\ 2^{cRiceParam}$
        $+abs\_remainder \mod bitMask$
    //where $a \mod n$ gives the remainder of the euclidean
    divison of $a$ by $n$
```

IV. PROPOSED VVC SELECTIVE ENCRYPTION

This section presents a new selective encryption scheme for VVC standard. The proposed selective encryption fulfills two

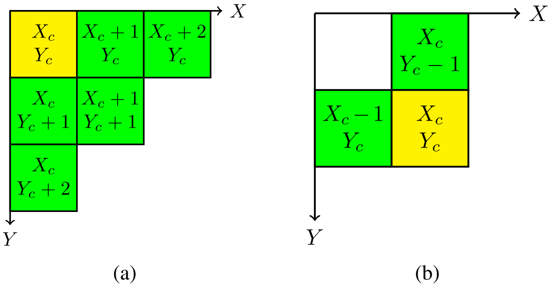


FIGURE 4. **TCs dependencies: coefficients highlighted in green are used to compute the local absolute sum of the current coefficient in yellow for (4a) TC mode and (4b) TS mode. S_1 and S_2 are two sets of green coefficients in (4a) and (4b), respectively.**

TABLE 1. **Encrypted syntax elements in the proposed VVC selective encryption solution, all these syntax elements are bypass coded.**

Coding block	Syntax elements	Binarization
Transform coefficients (TCs)	abs_remainder, dec_abs_level	TRp,EGk
	abs_remainderTS	TRp,EGk
	coeff_sign_flag	FL
	coeff_sign_flagTS	FL
Motion vector (MV)	abs_mvd_minus2	EGk
	mvd_sign_flag	FL
ALF Filter	alf_luma_fixed_filter_idx	TB
Inter Prediction	mmvd_direction_idx	FL
	merge_triangle_split_dir	FL
SAO Filter	sao_offset_sign	FL
	sao_band_position	FL
	sao_eo_class	FL
Intra Prediction	intra_chroma_pred_cand	FL

important features: standard format-compliant encryption (i.e. the bitstream must be decodable by any VVC decoder) and constant bitrate encryption (i.e. preserve the VVC compression efficiency). The proposed selective encryption is a reversible operation (i.e. it does not introduce loss of information when the secrete key is shared between the sender and the receiver). Therefore, PSNR computed after decryption process and without encryption process remains the same (i.e., no loss of information is caused by selective encryption).

The encryption is performed at the CABAC level of the encoder. Fig. 1 depicts in green the position of the selective encryption in the CABAC engine. The encryption is performed after the binarization process, and only a set of selected syntax elements, listed in Table 1 are ciphered. The encryption involves syntax elements from different coding tools including transform block, intra and inter predictions, and in-loop filters. This ensures the encryption of both intra (I) and inter (P and B) coded slices included in the VVC video sequence.

The syntax elements, listed in Table 1, have been selected based on following two criteria:

- The syntax element is bypassed: this restriction preserves the VVC coding efficiency.

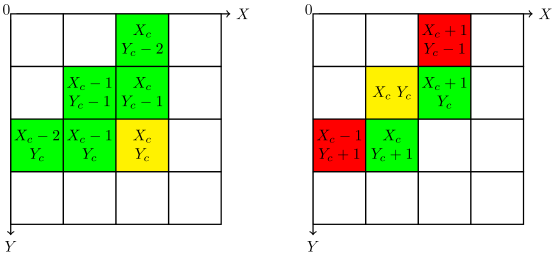


FIGURE 5. **The current coefficient (in yellow) is used to compute the local absolute sum of the coefficients highlighted in green for (5a) TC mode and (5b) TS mode. Coefficients highlighted in red are used along the current coefficient in the prediction of the parity of the coefficients in green. $\bar{S}_1$ and $\bar{S}_2$ are two sets of green coefficients in (5a) and (5b), respectively.**

- Changing any bin will not change how the bin-string is read by the decoder: this restriction ensures format-compliant encryption by excluding most of flags and syntax elements binarized by variable length codes.

The encryption of the most syntax elements listed in Table 1 is straightforward except the TCs that requires a specific processing to search for the encryptable bins. In the next section, we describe the encryption of the TCs since it is the most challenging syntax element to encrypt. The coding of the TCs introduces dependencies that need to be carefully addressed to perform format-compliant and constant bitrate encryption.

A. ENCRYPTION OF TRANSFORM COEFFICIENTS

This section presents how the TCs are encrypted. As explained in Section III-B3, the binarization of the TCs and especially the length of the suffix depends on the previously encoded TCs. Therefore, encryption that changes the value of the coefficients may introduce bitrate increase. Indeed, the binarization depends on a rice parameter *cRiceParam* $\in \{0, 1, 2, 3\}$ derived from previous TCs. This rice parameter defines the fixed length of the suffix and therefore it corresponds to the size of the encryptable bins. After an analysis of the binarization algorithm, multiple conditions ensuring constant bitrate have emerged and are presented below.

First, it is important to note that the coefficients are binarized in two different ways depending on whether they are processed by pass 2-1 or 2-2, as presented in Fig. 2.

- *The encryption **must not** change the parity of the coefficient:* changing the parity will result in changing the *state* of the CABAC context. The state is updated using the previous state value and the parity of the current coefficient.
- *The encryption **must not** change the rice parameter:* this will affect the bitrate.
- *The encryption **must not** change the V value for coefficients processed by pass 2-2:* changing the value of this parameter can result in changing the parity, and thus the CABAC context.

Considering those conditions, Algorithm 3 is proposed to identify the bins within the TCs that can be encrypted in constant bitrate and format compliance. The rice parameter of each TC is derived from a saturated absolute sum of the local neighborhood of the current TC. Fig. 5a depicts in green the set $\bar{S}_1$ of affected TCs if the current TC in yellow is modified by encryption. Therefore, for each affected coefficient, Algorithm 4 checks whether the changes in the local absolute sum will affect the context, the rice parameter and the V value. To make sure that the parity is not changed, the encryption excludes the least significant bit (LSB) of the suffix and will perform encryption only when the rice parameter is greater than 1 ($cRiceParam \in \{2, 3\}$).

The encryption of the coefficients in TS mode is similar. The main difference lies in how the rice parameter is derived. Fig. 5b shows the set $\bar{S}_2$ of affected coefficients in green when the current coefficient (in yellow) is modified by encryption. The current coefficient is binarized using a prediction based on the top and left coefficients. Therefore, the ciphered value of the current coefficient must remain lower or equal than the coefficient depicted in red in Fig. 5b according to the used prediction scheme.

Algorithms 3 and 4 are used to check that the encrypted bins of the current binarized coefficient are not affecting how the neighbor coefficients will be encoded. This enables defining the bins that can be encrypted in format-compliant and constant bitrate. The proposed solution is carried out as follows:

- Algorithm 3 checks at the binarization process whether the TC can be encrypted or not. The encryption is possible only when the absolute value of the coefficient is different from 0 ($absLevel \neq 0$) and the value of the derived rice parameter is above 1 ($cRiceParam > 1$).
- The algorithm then computes the minimum and maximum values of the encrypted remainder ($remMin, remMax$) of the current coefficient. The minimum ($absCMin$) and maximum ($absCMax$) absolute values of the coefficient are derived from their respective remainders.
- Algorithm 4 checks for all coefficients in $\bar{S}_1$ depicted in green in Fig. 5a (when they exist) whether ciphering the current coefficient (X_c, Y_c) will affect its neighbor coefficients (X_p, Y_p), the rice parameter or the V value. This operation is performed in five steps as follows:
    1) The algorithm computes a saturated absolute sum of the tested coefficient of coordinates (X_p, Y_p) ($AbsSumP1 = \sum_{i \in S_1} min(4 + |C_i| \mod 2, |C_i|)$) which is used for the context computation, with S_1 the set of neighbor coefficients of (X_p, Y_p). This operation is performed at the first pass (P1) to check the CABAC context change and set the *NoCtxChange* flag to **true** if the changes on the $AbsSumP1$ will not affect the context.
    2) The local absolute sum $TrAbsSumP21$ is then computed by (3) for the coefficient of coordinates (X_p, Y_p). The minimum possible value $TrAbsSumP21Min$ and the maximum value $TrAbsSumP21Max$ are also computed by (3) with $BaseLvl$ equals to 4.
    3) Then, the algorithm checks whether the rice parameter will be affected with the different computed sums in step 2 and sets a flag *NoRiceParChange* to **true** if the rice parameter remains unchanged with all possible tested conditions. I_R is a LUT containing, for each rice value, the interval in which the local absolute sum does not change the rice parameter.
    4) The parameter V is computed only in pass 2-2. At this fourth step, the algorithm checks for the processed coefficients if the parameter V remains unchanged to set the flag *NoVChange* to **true**. Similar to I_R, I_P returns, depending of the state and V values, the interval in which the local absolute sum does not change the V value.
    5) Finally, Algorithm 4 returns **true** when *NoCtxChange*, *NoRiceParChange* and *NoVC hange* are all equal to **true**.

Algorithm 3 $nbEncryptable = isEncryptable(X_c, Y_c, CoeffArr, nbEncryptable, bypass, V)$

```
Input: $(X_c, Y_c)$: the coordinate of the current pixel,
    $CoeffArr[][]$: the array of coefficient value,
    $nbEncryptable$: the number of encryptable bits to test,
    $bypass$: true if the current coefficient is bypass,
    $V$: of the current pixel, set to 0 if $bypass$ is false.
Output: the number of encryptable bits.

    $absLevel \leftarrow |CoeffArr[X_c][Y_c]|$
    if $absLevel \neq 0$ and $cRiceParam > 1$ then
        $absCMin, remMin \leftarrow computeMin(absLevel,$
            $nbEncryptable, bypass, V)$
        $absCMax, remMax \leftarrow computeMax(absLevel,$
            $nbEncryptable, bypass, V)$
        $encryptable \leftarrow$ not $(bypass$
            and $V \in [remMin, remMax])$
        for $p \in \bar{S}_1$ do
            $encryptable \leftarrow encryptable$ and
              $checkSumChange(X_p, Y_p,$
              $absLevel, absCMin, absCMax)$
        end for
        if not $encryptable$ and $nbEncryptable > 1$ then
            return $isEncryptable(X_c, Y_c, CoeffArr,$
              $nbEncryptable - 1, bypass, V)$
        else if $encryptable$ then
            return $nbEncryptable$
        else
            return 0
        end if
    else
        return 0
    end if
```

Algorithm 4 *Encryptable* = *checkSumChange*(X_p, Y_p, *absLevel*, *absCMin*, *absCMax*);

```
Input: $(X_p, Y_p)$: the coordinate of the tested coefficient,
  $absLevel$: the absolute value of the coefficient $(X_c, Y_c)$ before encryption,
  $absCMin$: the minimal possible encrypted value of the coefficient $(X_c, Y_c)$,
  $absCMax$: the maximal possible encrypted value of the coefficient $(X_c, Y_c)$.
Output: true if ciphered value does not affect the context, the rice parameter and the $C_0$, false otherwise.
  //Step 1
  $AbsSumP1 \leftarrow \sum_{i \in S_1} \min(4 + |C_i| \mod 2, |C_i|)$
  $numPos \leftarrow \sum_{i \in S_1} E(C_i)$
  //where $E(x)$ returns 1 if $x \neq 0$, 0 otherwise
  $AbsSumP1min \leftarrow AbsSumP1$
    $- \min(4 + (absLevel \mod 2), absLevel)$
    $+ \min(4 + (absCMin \mod 2), absCMin)$
  $NoCtxChange \leftarrow \left\lfloor \frac{AbsSumP1+1}{2} \right\rfloor \geq 3$
    and $AbsSumP1 - numPos \geq 4$
    and $\left\lfloor \frac{AbsSumP1Min+1}{2} \right\rfloor \geq 3$
    and $AbsSumP1Min - numPos \geq 4$
  //Step 2
  $AbsSumP21 \leftarrow \sum_{i \in S_1} |C_i|$
  $AbsSumP21Min \leftarrow AbsSumP21 - absLevel$
    $+ absCMin$
  $AbsSumP21Max \leftarrow AbsSumP21 - absLevel$
    $+ absCMax$
  $TrAbsSumP21 \leftarrow [AbsSumP21 - 20]_0^{31}$
  $TrAbsSumP21Min \leftarrow [AbsSumP21Min - 20]_0^{31}$
  $TrAbsSumP21Max \leftarrow [AbsSumP21Max - 20]_0^{31}$
  $RiceParP21 \leftarrow riceArr[TrAbsSumP21]$

  $TrAbsSumP22 \leftarrow [AbsSumP21]_0^{31}$
  $TrAbsSumP22Min \leftarrow [AbsSumP21Min]_0^{31}$
  $TrAbsSumP22Max \leftarrow [AbsSumP21Max]_0^{31}$
  $RiceParP22 \leftarrow riceArr[TrAbsSumP22]$
  //Step 3
  $NoRiceParChange \leftarrow true$
  if $TrAbsSumP21Min \notin I_R[RiceParP21]$
  or $TrAbsSumP21Max \notin I_R[RiceParP21]$
  or $TrAbsSumP22Min \notin I_R[RiceParP22]$
  or $TrAbsSumP22Max \notin I_R[RiceParP22]$ then
      $NoRiceParChange \leftarrow false$
  end if
  //Step 4
  $NoVChange \leftarrow true$
  for $i \leftarrow 0$ to 2 do
    $currV \leftarrow VArr[i][TrAbsSumP22]$
    if $TrAbsSumP22Min \notin I_P[i][currV]$
    or $TrAbsSumP22Max \notin I_P[i][currV]$ then
        $NoVChange \leftarrow false$
    end if
  end for
  //Step 5
  return $NoCtxChange$ and $NoRiceParChange$ and $NoVChange$
```

The decoder performs inverse operations performed by the encoder for deciphering. The decoder first decodes the TCs and then it searches for the encryptable coefficients using Algorithms 3 and 4. Finally, the deciphering will process only the identified encrypted bins.

B. ENCRYPTION METHOD AND SYNCHRONISATION

To cipher the syntax elements of a variable length, a stream cipher is more suited for this application. As the minimum error propagation is one of the most desirable properties in video encryption, we use the AES algorithm in CTR mode as a pseudo-random number generator (PRNG) to encrypt the identified syntax elements. It is important to note that CTR counter value should not be reused, which is adopted in our solution [31]. Meanwhile, other stream ciphers such as Rabbit [32], light weight chaos-based (LWCB) stream cipher [33], HC-128 [34] or even block ciphers like AES in CFB mode, can be used as well. A stream cipher produces a cipher text C using a XOR operation between the plain text P and the output steam X_g produced by a PRNG,

$$C(P) = P \oplus X_g. \tag{7}$$

To revert the encryption, a XOR between the cipher text and the same PRNG output is performed. Thus, a perfect synchronization between the encoder and the decoder is required. Most of the syntax elements are systematically ciphered and do not dependant on the position or the context. However, the syntax elements associated to the TCs are ciphered only if they meet conditions previously described in Section IV-A. One of these conditions relies on the neighbor coefficients, implying that the last decoded coefficient needs to be deciphered first. To allow this behavior, for each significant coefficient, encryptable or not, the PRNG generates a sample equal to the size of the rice parameter, e.g. the maximum encryptable size. The unused samples are discarded to keep the encoder and the decoder perfectly synchronized. In CTR mode, one bit flipping caused by transmission errors will only affect one bit during the deciphering process which minimizes the error propagation.

V. RESULTS AND DISCUSSIONS

In this section, we first present the experimental setup, followed by an assessment of the video degradation introduced by the selective encryption, then a security analysis will be presented, and finally a complexity evaluation is provided.

A. EXPERIMENTAL SETUP

The experiments are carried-out under the CTCs of the VVC standard. The CTCs define several test video sequences of different resolutions, and five quantization parameters (QPs) are used QP $\in$ {17, 22, 27, 32, 37}. The proposed encryption solution is implemented in

TABLE 2. Video sequences defined by the JVET CTCs.

Class	Sequence name	Resolution	Frame count	Frame rate	Bit depth
A1	Tango2	3840×2160	294	60	10
	FoodMarket4	3840×2160	300	60	10
	Campfire	3840×2160	300	30	10
A2	CatRobot	3840×2160	300	60	10
	DaylightRoad2	3840×2160	300	60	10
	ParkRunning3	3840×2160	300	50	10
B	MarketPlace	1920×1080	600	60	10
	RitualDance	1920×1080	600	60	10
	Cactus	1920×1080	500	50	8
	BasketballDrive	1920×1080	500	50	8
	BQTerrace	1920×1080	600	60	8
C	RaceHorses	832×480	300	30	8
	BQMall	832×480	600	60	8
	PartyScene	832×480	500	50	8
	BasketballDrill	832×480	500	50	8
D	RaceHorses	416×240	300	30	8
	BQSquare	416×240	600	60	8
	BlowingBubbles	416×240	500	50	8
	BasketballPass	416×240	500	50	8
E	FourPeople	1280×720	600	60	8
	Johnny	1280×720	600	60	8
	KristenAndSara	1280×720	600	60	8
F	ArenaOfValor	1920×1080	600	60	8
	BasketballDrillText	832×480	500	50	8
	SlideEditing	1280×720	300	30	8
	SlideShow	1280×720	500	20	8

the VVC test model (VTM)[1] version 6.0. VTM is the reference software implementation of both encoder and decoder of the VVC standard. The coding configuration without encryption is referred to as the Anchor. The video sequences are encoded with encryption in random access (RA) coding configuration. This latter is the common coding configuration used in broadcast and over-the-top (OTT) applications with an Intra period of 32 frames. The complexity measurements are performed on a desktop computer equipped with an Intel i7-7700 processor running at 3.60 GHz on Ubuntu 18.04 OS.

B. VIDEO QUALITY AND ENCRYPTION SPACE

1) VIDEO QUALITY

The distortion introduced by the proposed solution on the test video sequences is assessed in this section. Three full-reference objective image and video quality metrics are computed on the encrypted video sequences with respect to the original. PSNR is used to evaluate the video quality based on the mean squared error computed over the frame pixels [35]. The PSNR is computed as a weighted sum of the PSNR scores of the three color components. SSIM explores the structural similarity between the original and the decoded frame. It is important to note that a SSIM value close to 1 refers to decoded frame of a similar quality as the original frame [36]. Finally, VMAF is a video quality metric that predicts the perceived quality score of a video sequence [37], where a score of 100 indicates a good perceptual video quality and 0 refers to a very low perceived video quality.

Table 3 presents the PSNR performance overall video sequences at the five considered QPs. We can notice that the PSNR drops at QP 17 from 44.77 dB in average to around 10.17 dB. The same PSNR values of encrypted videos are reached on different QPs values. This indicates that the proposed solution significantly decreases the objective quality of the encrypted video. *Campfire* and *SlideShow* encrypted video sequences have very low PSNR values. For *Campfire*, it can be explained by its texture and complex shapes associated to high motion that increase the encryption space and thus improving the quality of the encryption. Concerning *SlideShow*, shapes are less complex however the encryption is able to flip the colors causing noticeable quality degradation with lower PSNR scores than the average.

Table 4 presents the average SSIM scores for seven video classes at different QPs. The proposed encryption solution enables to reduce the SSIM from around 1 to 0.25. The obtained SSIM value confirms that the proposed solution introduces a drastic distortion on the structure information within the encrypted video frame. We can notice that SSIM scores of class E video sequences are higher than the average scores. These video sequences have low motion and less texture compared to other sequences. This improves the coding efficiency and decreases the performance of the selective encryption since less syntax elements are encrypted.

Finally, Table 5 presents the VMAF scores which also emphasize the large degradation of the subjective video quality as a result of using the proposed encryption solution.

Fig. 6 illustrates the frame #10 of *RaceHorcesC* video sequence decoded at five QP values with and without encryption. The visual quality of decoded encrypted video is very low making difficult to recognize objects and colors in the video frame at all QP values with PSNR scores around 11 dB.

2) ENCRYPTION SPACE

The computational time of encryption mainly depends on the encryption space of any ciphering process. However, the robustness and security level will be enhanced by increasing the encryption space. In selective encryption, the somehow robust encryption algorithm and low computational overhead as outcome of the encryption is the target. Table 6 presents the encryption space of the proposed encryption solution as the percentage of encrypted bits by syntax element on the whole bitstream. The quality degradation of selective encryption is achieved by ciphering only 26.66% and 15.42% of the bitstream at high and low bitrates, respectively. We can notice that the largest encryption space is enabled by the encryption of the TCs while the part of other syntax element less present in the bitstream remains negligible ($< 2\%$).

[1] VTM software github: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

TABLE 3. PSNR performance of the proposed selective encryption for all video sequences at five QPs. Anchor and ciphered configurations correspond to the video decoded without encryption and with selective encryption, respectively.

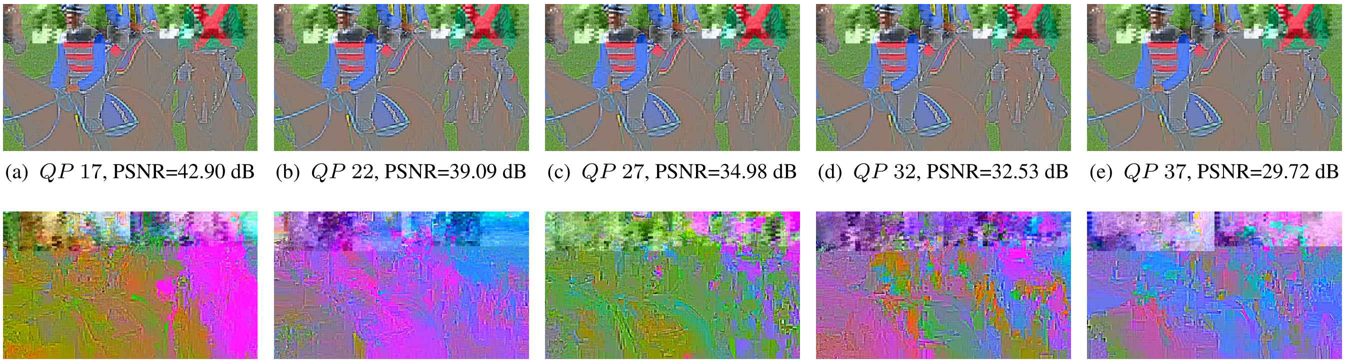
		$PSNR$ Scores (dB)									
		QP 17		QP 22		QP 27		QP 32		QP 37	
		Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered
A1	Campfire	44.02	4.58	39.78	4.69	37.64	4.60	36.54	4.38	35.18	4.37
	FoodMarket4	46.21	10.99	44.29	10.79	42.96	10.62	41.13	9.76	38.84	10.83
	Tango2	42.37	8.66	40.40	9.05	39.73	10.40	38.88	8.16	37.58	8.68
A2	CatRobot1	42.84	9.13	40.48	9.55	39.56	9.73	38.37	8.72	36.70	9.08
	DaylightRoad2	41.76	8.93	38.25	11.16	37.33	9.34	36.44	9.74	35.12	10.00
	ParkRunning3	47.51	11.24	43.83	11.52	39.85	10.56	36.58	12.00	33.60	11.10
B	BasketballDrive	42.08	13.17	39.55	12.95	37.98	11.90	36.34	11.26	34.40	11.63
	BQTerrace	42.76	10.22	37.66	10.23	35.56	10.28	34.29	10.15	32.78	9.82
	Cactus	41.54	10.04	38.74	10.23	37.24	9.48	35.59	9.12	33.51	8.99
	MarketPlace	43.52	9.55	40.96	9.10	38.84	8.22	36.72	8.61	34.44	8.67
	RitualDance	47.02	9.43	44.79	10.44	41.71	10.11	38.71	9.28	35.76	9.56
C	BasketballDrill	44.19	13.40	41.69	12.83	38.57	12.88	35.71	11.94	33.09	11.22
	BQMall	42.70	11.94	40.73	11.12	38.34	10.79	35.81	11.16	33.14	10.65
	PartyScene	42.13	12.02	39.05	11.78	35.73	11.34	32.76	11.21	29.96	11.33
	RaceHorsesC	43.13	11.53	39.58	11.58	36.48	11.74	33.80	11.64	31.15	10.97
D	BasketballPass	45.22	13.54	41.58	13.44	37.54	13.15	34.27	13.41	31.39	13.79
	BlowingBubbles	41.75	11.20	38.81	11.69	35.60	11.06	32.69	11.44	29.87	11.57
	BQSquare	42.10	8.89	38.69	9.37	35.41	8.78	32.70	8.93	30.22	9.62
	RaceHorses	43.66	11.75	40.14	12.11	36.55	12.05	33.25	11.74	30.31	11.56
E	FourPeople	44.92	9.91	43.38	9.72	41.81	8.73	39.82	8.93	37.30	8.70
	Johnny	44.99	9.54	43.53	9.26	42.38	8.70	40.94	8.58	39.01	8.48
	KristenAndSara	45.42	9.50	43.86	9.15	42.40	7.60	40.64	7.73	38.42	7.62
F	ArenaOfValor	46.94	11.15	43.63	10.78	40.28	10.67	37.45	9.38	34.79	9.46
	BasketballDrillText	44.31	12.52	41.73	12.01	38.52	12.10	35.60	11.24	32.89	11.21
	SlideEditing	54.72	9.69	51.35	10.49	47.31	10.15	43.25	10.72	38.92	9.31
	SlideShow	56.12	1.94	52.43	3.16	48.61	4.28	45.17	3.98	41.83	4.35
	Average	**44.77**	**10.17**	**41.88**	**10.32**	**39.38**	**9.97**	**37.05**	**9.74**	**34.62**	**9.71**

TABLE 4. Average SSIM performance of the proposed encryption solution for all video classes at five QPs.

	$SSIM$ Score									
	QP 17		QP 22		QP 27		QP 32		QP 37	
	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered
A1	1.00	0.27	1.00	0.22	1.00	0.23	0.99	0.21	0.99	0.22
A2	1.00	0.23	1.00	0.23	1.00	0.21	0.99	0.21	0.99	0.21
B	1.00	0.27	1.00	0.27	0.99	0.26	0.99	0.25	0.97	0.25
C	1.00	0.23	0.99	0.23	0.98	0.23	0.97	0.23	0.94	0.24
D	0.99	0.24	0.97	0.24	0.95	0.25	0.91	0.26	0.85	0.28
E	1.00	0.39	1.00	0.40	0.99	0.37	0.99	0.37	0.99	0.39
F	1.00	0.26	1.00	0.31	1.00	0.35	0.99	0.36	0.97	0.35
Average	**1.00**	**0.27**	**0.99**	**0.27**	**0.99**	**0.27**	**0.97**	**0.27**	**0.95**	**0.28**

TABLE 5. Average VMAF performance of the proposed encryption solution for all video classes at five QPs.

	$VMAF$ Score									
	QP 17		QP 22		QP 27		QP 32		QP 37	
	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered	Anchor	Ciphered
A1	99.47	25.53	99.03	28.84	96.70	30.23	92.31	29.84	84.86	28.19
A2	99.88	23.79	99.22	23.88	97.20	25.15	92.49	25.27	85.02	25.22
B	99.80	2.78	99.08	2.89	96.35	3.24	89.46	3.52	77.84	3.67
C	99.88	7.16	99.59	7.72	96.91	7.78	89.84	7.63	77.55	7.88
D	99.42	6.66	98.78	6.57	95.68	6.76	87.96	6.75	75.68	6.29
E	97.37	0.16	96.54	0.51	95.00	0.62	92.04	0.71	86.52	1.27
F	98.88	6.20	98.65	7.83	96.72	8.57	92.47	8.66	85.43	6.86
Average	**99.30**	**9.32**	**98.76**	**10.10**	**96.37**	**10.64**	**90.73**	**10.66**	**81.27**	**10.25**



(a) QP 17, PSNR=42.90 dB (b) QP 22, PSNR=39.09 dB (c) QP 27, PSNR=34.98 dB (d) QP 32, PSNR=32.53 dB (e) QP 37, PSNR=29.72 dB

(f) QP 17, PSNR=12.07 dB (g) QP 22, PSNR=11.73 dB (h) QP 27, PSNR=10.03 dB (i) QP 32, PSNR=10.68 dB (j) QP 37, PSNR=10.25 dB

FIGURE 6. Visual illustration of Frame #10 of *RaceHorsesC* video decoded without encryption (6a - 6e) and with selective encryption (6f - 6j) at five QPs.

TABLE 6. Encryption space in percentage (%) per syntax element at five QPs.

Syntax Elements	Encryption Space (%)				
	QP 17	QP 22	QP 27	QP 32	QP 37
alf_luma_filter_idx	0.01	0.02	0.04	0.06	0.06
sao_offset_sign	0.00	0.00	0.00	0.00	0.00
sao_band_position	0.01	0.01	0.01	0.01	0.01
sao_eo_class	0.01	0.02	0.01	0.01	0.00
mmvd_direction_idx	0.48	0.55	0.56	0.59	0.59
merge_triangle_split_dir	0.04	0.08	0.13	0.17	0.19
mvd_abs	0.26	0.48	0.76	0.98	1.14
mvd_sign	0.29	0.51	0.75	0.92	1.00
abs_remainder coeff_sign	25.44	21.39	17.15	14.33	12.22
abs_remainderTS coeff_signTS	0.06	0.06	0.05	0.03	0.01
intra_chroma _pred_candidate	0.07	0.10	0.14	0.16	0.18
Total	**26.66**	**23.21**	**19.61**	**17.24**	**15.42**

C. SECURITY ANALYSIS

In the previous section we only assess the visual degradation achieved by the proposed encryption. In this section we focus on the quality of the proposed encryption and its robustness against different types of attacks.

1) EQ ANALYSIS

The algebraic summation of differences between pixels distributions of the original frame $H(P)$ and the encrypted fame $H(C)$ is called EQ. This latter is computed as follows [20]

$$\mathrm{EQ} = \frac{\sum_{Z=0}^{2^d-1} |H_Z(C) - H_Z(P)|}{2^d}, \tag{8}$$

where $H_Z(C)$ is the total number of occurrences of byte Z in the ciphered frame C and $H_Z(P)$ is the total number of occurrences of the same byte Z in the original frame P. Finally, d is the bitdepth of the video. The higher EQ value is, the more secure is the selective encryption solution.

TABLE 7. Encryption Quality for CTCs video classes at five QP values.

	Encryption Quality (EQ)					
	QP 17	QP 22	QP 27	QP 32	QP 37	EQ_{max}
A1	8 661	8 260	8 573	7 947	8 535	16 200
A2	9 235	6 791	8 058	8 871	7 971	16 200
B	1 620	1 805	1 923	1 920	1 911	4 050
C	231	245	224	236	270	780
D	61	69	81	70	83	195
E	682	748	1 057	1 150	1 096	1 800
F	907	1 225	944	1 240	1 298	2 108
Average	**2640**	**2407**	**2603**	**2680**	**2652**	**5199**

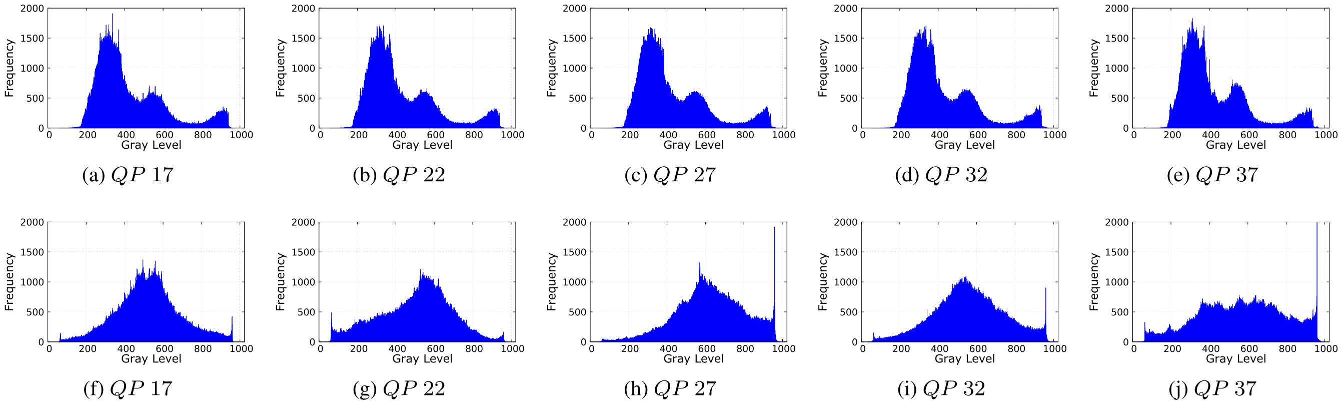
Table 7 presents the EQ values for all video classes at the five considered QPs. The presented values are the average EQ over encrypted frames and video sequences of each class. The EQ does not have a relative point for comparison. A derivation from (8) is proposed to compute the upper bound value of the EQ [10] as follows

$$\mathrm{EQ}_{max} = \frac{2\,W\,H}{2^d}, \tag{9}$$

where H and W are the video height and width, respectively and d is the bit depth. The upper bound value of the EQ is reached when the histograms of the two frames $H_Z(C)$ and $H_Z(P)$ are not overlapping.

The average EQ values are within the interval [2407, 2680] in average with a theoretical average upper bound of 5199.

The HEVC selective encryption solution proposed in [10] achieved an EQ value for *Kimono* video sequence is higher than 38.54% of its maximum EQ, and an EQ value for *PeopleOnStreet* video is higher than 40.92% its maximum EQ. The proposed solution of different videos at different configuration ranges between 46.29% and 51.56% of the maximum EQ which confirms that the proposed solution has a high security level regarding the encryption quality metric.



(a) QP 17 (b) QP 22 (c) QP 27 (d) QP 32 (e) QP 37

(f) QP 17 (g) QP 22 (h) QP 27 (i) QP 32 (j) QP 37

FIGURE 7. Histograms of frame #10 computed for the anchor (7a- 7e) and cipher (7f- 7j) of *RaceHorsesC* video at five *QP* values. The selective encryption solution significantly changes the pixels distributions as shown by the EQ metric.

2) HISTOGRAM ANALYSIS

Histogram of encrypted frame should be more uniform than original frame histogram in order to resist to statistical analysis based attacks [38], [39]. Fig. 7 illustrates the histograms of frame #10 of *RaceHorcesC* video sequence before and after selective encryption at five considered QPs. The histograms of the encrypted frames are completely different from the original frame histogram. In fact, the proposed encryption solution changes the distribution of the decoded pixels toward different pattern which is close to uniform distribution especially at lower bitrate (ie. high QP). We can also notice that in contrast to full encryption, it is difficult for constant bitrate and format compliant selective encryption to reach the uniform distribution of the histogram at all coding configurations and video contents.

3) EDGES AND STRUCTURAL INFORMATION PROTECTION

Edge detection enables assessing the ability of an encryption solution to hide the edge information in the encrypted frame. This section evaluates the ability of the proposed encryption solution to hide the edge in the encrypted video sequence. The EDR is computed by (10) [21], [40].

$$\mathrm{EDR} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}|P_{ED}(i,j) - C_{ED}(i,j)|}{\sum_{i=1}^{H}\sum_{j=1}^{W}|P_{ED}(i,j) + C_{ED}(i,j)|}, \qquad (10)$$

with P_{ED} and C_{ED} are the binary Laplacian of the decoded images with and without encryption, respectively. EDR takes values in the interval [0, 1], where a value close to 1 corresponds to a high edge hiding capability.

Table 8 presents the average values of EDR computed on the first 64 frames of the CTCs video sequences. The average EDR values are higher than 0.87 which shows the ability of the proposed selective encryption to hide edges and structural information in the encrypted frames. We can notice a lower EDR performance for class F video sequences. This class includes mainly screen content video sequences for which selective encryption is less effective to hide the structure of the edges.

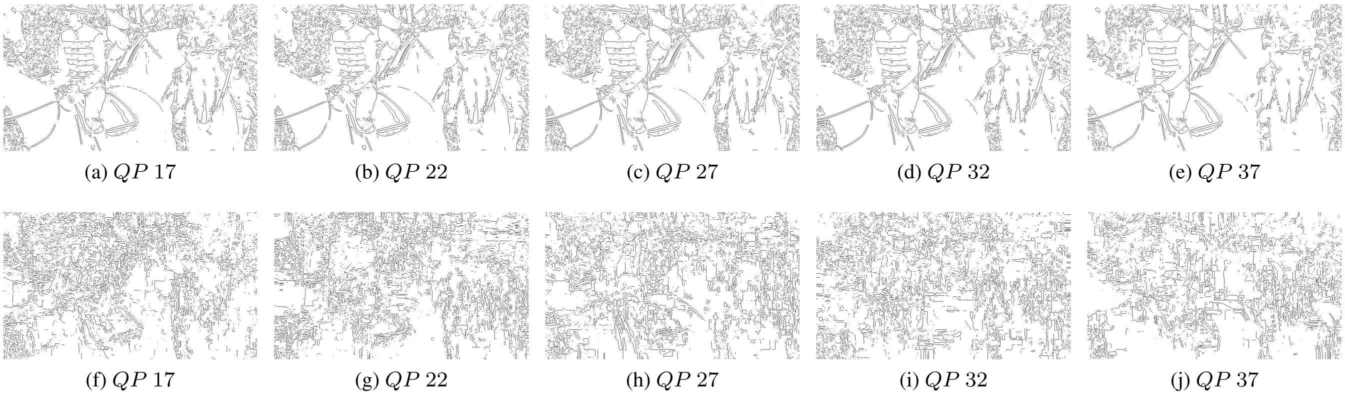
TABLE 8. Average EDR for CTCs video classes at five QP values.

	EDR				
	QP 17	QP 22	QP 27	QP 32	QP 37
A1	0.919	0.923	0.928	0.932	0.934
A2	0.890	0.896	0.900	0.904	0.906
B	0.890	0.893	0.896	0.899	0.904
C	0.879	0.877	0.877	0.876	0.874
D	0.876	0.887	0.890	0.880	0.881
E	0.866	0.864	0.856	0.856	0.844
F	0.813	0.780	0.762	0.741	0.762
Average	**0.875**	**0.873**	**0.871**	**0.867**	**0.871**

Fig. 8 illustrates the edges of the decoded frame #10 of *RaceHorsesC* sequence without encryption (first row) and with encryption at the second row for five different QPs. This figure clearly shows that the ciphered frames are noisy caused by high frequency structure introduced by the selective encryption. Therefore, structural information including edges in the ciphered frames are hidden and can be hardly explored by the EDR based attacks.

4) SENSITIVITY TO SECRET KEYS

Key sensitivity attacks are mainly based on the fact that the adversary tries to decipher the encrypted frames using a key close to the secret key used for encryption. The second adversary scenario is to guess the key if the encryption system provides information related to the used secrete key such as the sensitivity of the encryption regarding small change in the key. The proposed encryption algorithm should produce a completely different encrypted frame when a slight change (one bit change) on the used secrete key [41]. Evaluation of the system robustness against key sensitivity attacks can be assessed using many existing tools such as UACI and NPCR [42], [43]. To compute these metrics, one random key is generated K_1 and a key with only one bit difference K_2 is created. The two keys are then used to cipher the same frame of width W and height H and a bit depth d. The result will


(a) QP 17 (b) QP 22 (c) QP 27 (d) QP 32 (e) QP 37

(f) QP 17 (g) QP 22 (h) QP 27 (i) QP 32 (j) QP 37

FIGURE 8. Edge Detection on Frame #10 computed for the anchor (8a-8e) and ciphered (8f- 8j) *RaceHorsesC* video at five QPs.

TABLE 9. NPCR and UACI with two secret keys with 1-bit-difference.

	$UACI$ and $NPCR$					
	QP 17		QP 27		QP 37	
	$UACI$	$NPCR$	$UACI$	$NPCR$	$UACI$	$NPCR$
A1	25.52	99.62	24.68	99.77	31.18	99.90
A2	22.03	99.85	21.30	99.82	30.29	99.76
B	24.48	99.83	24.27	99.82	27.04	99.60
C	19.60	99.82	24.32	99.86	23.21	99.84
D	22.18	99.85	23.12	99.83	23.51	99.69
E	23.69	99.79	32.54	99.91	37.79	99.91
F	26.84	99.52	26.96	99.33	22.65	98.76
Ave.	**23.48**	**99.76**	**25.17**	**99.76**	**27.32**	**99.61**

create a ciphered frame C_1 using K_1, and C_2 using K_2. The UACI and the NPCR are defined as follow

$$\text{UACI} = \frac{1}{H\,W\,2^d} \sum_{i=1}^{H} \sum_{j=1}^{W} |C_1(i,j) - C_2(i,j)|\ 100\%, \tag{11}$$

$$\text{NPCR} = \frac{1}{H\,W} \sum_{i=1}^{H} \sum_{j=1}^{W} D(i,j)\, 100\%, \tag{12}$$

with:

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j). \end{cases}$$

The optimal NPCR and UACI values of a secure image encryption scheme against key sensitivity attacks are 99.58% and 33.46%, respectively [44].

Table 9 presents the obtained NPCR and UACI values on the CTCs for all video classes at three QPs. Here, it is important to note that the NPCR and UACI results of the selective encryption should not be analysed as in full image encryption. However, the obtained values can give an indication on the ability of the selective encryption to resist key sensitive and differential attacks. The average NPCR values at the three QPs for all classes are very close to the optimal value of a secure encryption scheme against key sensitivity attacks. Moreover, the average UACI values lie in the interval [19.60, 37.79] with an average value overall classes that converges to the optimal value of 33.64. This performance in terms of both NPCR and UACI proves the robustness of the proposed selective encryption solution with regards the key sensitivity and differential attacks.

5) BRUTE FORCE ATTACK

Brute force attack or exhaustive search attack performs testing all possible values of the used secret key in order to partially or completely break the cipher [45]. It is well-known, that any encryption algorithm with at least 128 bits as secret key is considered as resilient to brute force attack, which is the case for the used AES algorithm. In selective encryption, the total number of tries to correctly guess the selected encrypted bits should be at least 2^{128} tries in order to resist to brute force attack [46], [47]. Our proposed selective encryption algorithm relies on AES in counter mode as stream cipher with a secret key size of 128 bits. Moreover, the size of the encryption space is very large.

6) ERROR CONCEALMENT ATTACK

The error concealment attack is a kind of attack based on guessing the encrypted bits based on some assumptions. However, since the encryption space of a VVC video is large, the only scenario that the adversary can follow is to try replacing all encrypted bits with the same value (zero or one) and decipher the modified encrypted frame [48], [49]. In order to evaluate our proposed solution regarding error concealment attacks, all encrypted bits are replaced by zero and then PSNR, SSIM and VMAF are calculated again under the same CTCs. Table 10 gives the average PSNR, SSIM and VMAF scores of video sequences deciphered with replacement attack at the five QPs. The obtained quality scores are similar or even worst compared to encrypted video with AES generator presented in Section V-B. This confirms that the proposed selective encryption solution is robust against attacks based on replacement bits.

TABLE 10. Replacement attack, average PSNR, SSIM and VMAF score on CTCs.

QP	PSNR		SSIM		VMAF	
	Anchor	Replaced	Anchor	Replaced	Anchor	Repla.
17	44.85	5.87	1.00	0.29	99.21	6.77
22	41.98	5.80	0.99	0.30	98.47	6.67
27	39.42	5.90	0.99	0.31	95.78	7.16
32	37.02	5.80	0.97	0.32	89.88	7.29
37	34.54	5.92	0.95	0.32	80.21	6.93

TABLE 11. Deciphering time Δ_{SE} in second and deciphering overhead CO_{SE} in % on Intel i7-7700 processor at 3.6 GHz.

	$QP = 17$		$QP = 27$		$QP = 37$	
	Δ_{SE}	CO_{SE}	Δ_{SE}	CO_{SE}	Δ_{SE}	CO_{SE}
A1	1.356	3.540	0.091	0.518	-0.013	-0.147
A2	2.514	5.605	0.156	0.674	-0.001	-0.047
B	0.473	4.579	0.039	0.976	0.001	0.041
C	0.116	5.136	0.025	2.101	0.006	0.791
D	0.026	4.476	0.005	1.384	0.001	0.450
E	0.072	3.205	0.004	0.387	-0.002	-0.198
F	0.086	2.933	0.023	1.331	0.008	0.701
Average	**0.581**	**4.236**	**0.045**	**1.111**	**0.001**	**0.261**

D. COMPLEXITY ANALYSIS

The aim of this section is to assess the complexity of the proposed encryption solution. The complexity overhead is computed only for the decoder, since the encryption overhead is negligible with respect to the encoding time. The average decoding run time is computed based on 100 decodings without deciphering ($DecT_{Ref}$) and with deciphering ($DecT_{SE}$). The deciphering run time Δ_{SE} is computed as a difference between decoding times with and without deciphering $\Delta_{SE} = DecT_{SE} - DecT_{Ref}$, while the percentage of deciphering complexity overhead CO_{SE} is derived as follows

$$CO_{SE} = \frac{\Delta_{SE}}{DecT_{Ref}} 100\%. \tag{13}$$

Table 11 gives the deciphering time Δ_{SE} in second and the deciphering complexity overhead in percentage for all video classes at three QPs. The deciphering time does not exceed 3 seconds even for high bitrate and high resolution 4K videos of classes A and B. This corresponds to less than 6% of the total decoding time. The average deciphering overhead remains lower than 4.23% observed at high bitrate presenting more TCs to cipher. Finally, negative values at QP=37 are caused by the time precision error which is higher than the deciphering time.

VI. CONCLUSION

In this paper a new selective encryption solution for the VVC standard was proposed. This solution encrypts at the CABAC level a set of VVC syntax elements in format-compliant and constant bitrate. The coding of the TCs in VVC introduces several dependencies making constant bitrate encryption more challenging. We have proposed an original algorithm that analyses the coding dependencies of the TCs to determine the number and positions of encryptable bins for each coefficient. The proposed encryption solution was integrated in both encoder and decoder of the VVC reference software VTM 6.0. The quality of the encrypted video was assessed under the VVC CTCs with three objective quality metrics including PSNR, SSIM and VMAF. The low obtained quality scores clearly show the quality degradation enabled by the encryption. Security analysis was also conducted to asses the robustness against several attacks including statistical, key sensitivity and brute force attacks. Finally, the complexity overhead of the deciphering at the decoder side is estimated and remains lower than 6% of the decoding time confirming the lightweight advantage of the proposed encryption solution.

As future work, the proposed encryption solution will be implemented under real time software VVC encoder and decoder such as VVenC encoder and OpenVVC decoder.

REFERENCES

[1] *Specification for the Advanced Encryption Standard (AES)*, Standard 197, Federal Information Process, 2001.

[2] M. Van Droogenbroeck, "Partial encryption of images for real-time applications," in *Proc. IEEE Signal Process. Symp.*, vol. 1, no. 2, Apr. 2004, pp. 11–15.

[3] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia Syst.*, vol. 15, no. 4, pp. 243–270, 2009.

[4] J.-L. Liu, "Efficient selective encryption for JPEG 2000 images using private initial table," *Pattern Recognit.*, vol. 39, no. 8, pp. 1509–1517, 2006.

[5] 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.

[6] S.-W. Park and S.-U. Shin, "Efficient selective encryption scheme for the H.264/scalable video Coding(SVC)," in *Proc. 4th Int. Conf. Netw. Comput. Adv. Inf. Manage.*, Sep. 2008, pp. 371–376.

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

[8] M. Farajallah, W. Hamidouche, O. Deforges, and S. E. Assad, "ROI encryption for the HEVC coded video contents," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3096–3100.

[9] B. Boyadjis, C. Bergeron, B. Pesquet-Popescu, and F. Dufaux, "Extended selective encryption of H.264/AVC (CABAC)- and HEVC-encoded video streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 892–906, Apr. 2017.

[10] W. Hamidouche, M. Farajallah, N. Sidaty, S. El Assad, and O. Deforges, "Real-time selective video encryption based on the chaos system in scalable HEVC extension," *Signal Process., Image Commun.*, vol. 58, pp. 73–86, Oct. 2017.

[11] G. Van Wallendael, A. Boho, J. D. Cock, A. Munteanu, and R. Van De Walle, "Encryption for high efficiency video coding with video adaptation capabilities," *IEEE Trans. Consum. Electron.*, vol. 59, no. 3, pp. 634–642, Aug. 2013.

[12] V. A. Memos and K. E. Psannis, "Encryption algorithm for efficient transmission of HEVC media," *J. Real-Time Image Process.*, vol. 12, no. 2, pp. 473–482, 2016.

[13] M. Long, F. Peng, and X. Gong, "A format-compliant encryption for secure HEVC video sharing in multimedia social network," *Int. J. Digit. Crime Forensics*, vol. 10, no. 2, pp. 23–39, Apr. 2018.

[14] A. I. Sallam, E.-S. M. El-Rabaie, and O. S. Faragallah, "Efficient HEVC selective stream encryption using chaotic logistic map," *Multimedia Syst.*, vol. 24, no. 4, pp. 419–437, 2018.

[15] W. Hamidouche, T. Biatek, M. Abdoli, E. Francois, F. Pescador, M. Radosavljevic, D. Menard, and M. Raulet, ''Versatile video coding standard: A review from coding tools to consumers deployment,'' *IEEE Consum. Electron. Mag.*, early access, Jan. 21, 2022, doi: 10.1109/MCE.2022.3144545.

[16] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, ''Overview of the versatile video coding (VVC) standard and its applications,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[17] C. Bonnineau, W. Hamidouche, J. Fournier, N. Sidaty, J.-F. Travers, and O. Deforges, ''Perceptual quality assessment of HEVC and VVC standards for 8K video,'' *IEEE Trans. Broadcast.*, pp. 1–8, 2022.

[18] N. Sidaty, W. Hamidouche, O. Déforges, P. Philippe, and J. Fournier, ''Compression performance of the versatile video coding: Hd and UHD visual quality monitoring,'' in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.

[19] C. E. Shannon, ''Communication theory of secrecy systems,'' *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[20] H. E. H. Ahmed, H. M. Kalash, and O. S. F. Allah, ''Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images,'' in *Proc. Int. Conf. Electr. Eng.*, Apr. 2007, pp. 1–7.

[21] N. Taneja, B. Raman, and I. Gupta, ''Chaos based partial encryption of SPIHT compressed images,'' *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 9, no. 2, pp. 317–331, Mar. 2011.

[22] Y. Wu, S. Member, P. Joseph Noonan, L. Member, S. Agaian, and S. Member, ''NPCR and UACI randomness tests for image encryption,'' *Cyber J., Multidisciplinary J. Sci. Technol.*, vol. 1, no. 2, pp. 31–38, 2011.

[23] M. A. Taha, W. Hamidouche, N. Sidaty, M. Viitanen, J. Vanne, S. El Assad, and O. Deforges, ''Privacy protection in real time HEVC standard using chaotic system,'' *Cryptography*, vol. 4, no. 2, p. 18, Jun. 2020.

[24] W. Hamidouche, M. Raulet, and O. Déforges, ''4K real-time and parallel software video decoder for multilayer HEVC extensions,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 169–180, Jan. 2016.

[25] A. Shamir, ''How to share a secret,'' *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[26] V. Vijayalakshmi, L. Varalakshmi, and G. F. Sudha, ''Efficient encryption of intra and inter frames in MPEG video,'' in *Proc. Int. Conf. Netw. Secur. Appl.* Chennai, India: Springer, 2010, pp. 93–104.

[27] F. Peng, X. Zhang, Z.-X. Lin, and M. Long, ''A tunable selective encryption scheme for H.265/HEVC based on chroma IPM and coefficient scrambling,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2765–2780, Aug. 2020.

[28] D. Xu, ''Data hiding in partially encrypted HEVC video,'' *ETRI J.*, vol. 42, no. 3, pp. 446–458, Jun. 2020.

[29] S. J. Kim Chen and Y. Ye, ''Algorithm description for versatile video coding and test model 10 (VTM 10),'' JVET, New York, NY, USA, Tech. Rep. S2002, Jul. 2020.

[30] V. Sze and M. Budagavi, ''High throughput CABAC entropy coding in HEVC,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.

[31] H. Lipmaa, P. Rogaway, and D. Wagner, ''Comments to NIST concerning AES modes of operations: CTR-mode encryption,'' in *National Institute of Standards and Technologies*. Princeton, NJ, USA: Citeseer, 2000.

[32] M. Boesgaard, M. Vesterager, and E. Zenner, ''The rabbit stream cipher,'' in *New Stream Cipher Designs* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2008, pp. 69–83.

[33] G. Gautier, M. L. Glatin, S. E. Assad, W. Hamidouche, O. Déforges, S. Guilley, and A. Facon, ''Hardware implementation of lightweight chaos-based stream cipher,'' in *Proc. Int. Conf. Cyber-Technol. Cyber-Syst.*, Porto, Portugal, Sep. 2019, pp. 1–5.

[34] H. Wu, ''The stream cipher HC-128,'' in *New Stream Cipher Designs* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2008.

[35] S. Winkler and P. Mohandas, ''The evolution of video quality measurement: From PSNR to hybrid metrics,'' *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 660–668, Sep. 2008.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ''Image quality assessment: From error visibility to structural similarity,'' *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[37] R. Rassool, ''VMAF reproducibility: Validating a perceptual practical video quality metric,'' in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2017, pp. 1–2.

[38] R. Qumsieh, M. Farajallah, and R. Hamamreh, ''Joint block and stream cipher based on a modified skew tent map,'' *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 33527–33547, Dec. 2019.

[39] M. Lewis-Beck, *Data Analysis: An Introduction*, no. 103. Newbury Park, CA, USA: Sage, 1995.

[40] N. Tanejaa, B. Ramanb, and I. Gupta, ''Selective image encryption in fractional wavelet domain,'' *AEU-Int. J. Electron. Commun.*, vol. 65, pp. 338–344, Apr. 2011.

[41] N. K. Pareek, V. Patidar, and K. K. Sud, ''Diffusion–substitution based gray image encryption scheme,'' *Digit. Signal Process.*, vol. 23, pp. 894–901, Mar. 2013.

[42] Y. Wu, J. P. Noonan, and S. Agaian, ''NPCR and UACI randomness tests for image encryption,'' *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecommu.*, vol. 1, pp. 31–38, Apr. 2011.

[43] E. Biham and A. Shamir, ''Differential cryptanalysis of DES-like cryptosystems,'' *J. Cryptol.*, vol. 4, no. 1, pp. 3–72, 1991.

[44] F. Maleki, A. Mohades, S. M. Hashemi, and M. E. Shiri, ''An image encryption system by cellular automata with memory,'' in *Proc. 3rd Int. Conf. Availability, Rel. Secur.*, Mar. 2008, pp. 1266–1271.

[45] B. A. Forouzan,'' *Cryptography & Network Security*. New York, NY, USA: McGraw-Hill, 2007.

[46] A. Said, ''Measuring the strength of partial encryption schemes,'' in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, p. 1126.

[47] S. Choi, ''Privacy-preserving H.264 video encryption scheme,'' *ETRI J.*, vol. 33, no. 6, pp. 935–944, Dec. 2011.

[48] T. Stütz and A. Uhl, ''On JPEG2000 error concealment attacks,'' in *Proc. Pacific–Rim Symp. Image Video Technol.*, Cham, Switzerland: Springer, 2009, pp. 851–861.

[49] F. Dufaux and T. Ebrahimi, ''Scrambling for privacy protection in video surveillance systems,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1168–1174, Aug. 2008.