End-to-end security for video distribution: the combination of encryption, watermarking, and video adaptation

Andras Boho, Glenn Van Wallendael, Ann Dooms, Jan De Cock, Geert Braeckman, Peter Schelkens, Bart Preneel, and Rik Van de Walle

In: Signal Processing Magazine, IEEE, 30 (2), 97-107, 2013.

End-to-end security for video distribution: the combination of encryption, watermarking, and video adaptation

Andras Boho[‡°], Glenn Van Wallendael[⋆°], Ann Dooms[*°], Jan De Cock[⋆°], Geert Braeckman[*°], Peter Schelkens[*°], Bart Preneel[‡°], and Rik Van de Walle[⋆°]

[‡] K.U. Leuven, ESAT/SCD (COSIC),
Kasteelpark Arenberg 10 (box 2446), B-3001 Heverlee, Belgium.
[⋆] Ghent University, ELIS - Multimedia Lab,
Gaston Crommenlaan 8 (box 201), B-9050 Ledeberg-Ghent, Belgium
[*] Vrije Universiteit Brussel (VUB), Dept. of Electronics and Informatics (ETRO),
Pleinlaan 2, B-1050 Brussels, Belgium.
[°] Interdisciplinary Institute for Broadband Technology (IBBT),
Gaston Crommenlaan 8 (box 102), B-9050 Ghent, Belgium.

Abstract

Abstract. Content distribution applications such as digital broadcasting, video-on-demand services (VoD), video conferencing, surveillance and telesurgery are confronted with difficulties - besides the inevitable compression and quality challenges - with respect to intellectual property management, authenticity, privacy regulations, access control etc. Meeting such security requirements in an end-to-end video distribution scenario poses significant challenges. If the entire content is encrypted at the content creation side, the space for signal processing operations is very limited. Decryption, followed by video processing and re-encryption is also to be avoided as it is far from efficient, complicates key management and could expose the video to possible attacks. Additionally, also when the content is delivered and decrypted, the protection is gone. Watermarking can complement encryption in these scenarios by embedding a message within the content itself containing for example ownership information, unique buyer codes or content descriptions. Ideally, securing the video distribution should therefore be possible throughout the distribution chain in a flexible way allowing the encryption, watermarking and encoding/transcoding operations to commute.

This paper introduces the reader to the relevant techniques that are needed to implement such an end-to-end commutative security system for video distribution, and presents a practical solution for encryption and watermarking compliant with H.264/AVC and the upcoming HEVC (High

Efficiency Video Coding) video coding standards. To minimize the overhead and visual impact, a practical trade-off between the security of the encryption routine, robust watermarking and transcoding possibilities is investigated. We demonstrate that our combined commutative protection system effectively scrambles video streams, achieving SSIM (Structural Similarity Index) values below 0.2 across a range of practical bit rates, while allowing robust watermarking and transcoding.

Commuting: a protection solution for an end-to-end video distribution system

In current video distribution scenarios, it is often hard for the content producers to keep track of the distribution of their content due to the number of middlemen in the value chain that sit between the content producer and the end consumer. Fig. 1 shows a typical end-to-end video distribution chain, where (possibly encrypted) video content is delivered by the content producer to the distribution network via a dedicated channel (e.g. a satellite channel) or video storage servers. Network providers or cable operators pick up the content and might want to optimize their bandwidth and the quality-of-service to the end users by transcoding the video stream. The classical example is the case in which the ultimate destination of the video is not known in advance and can vary from an HDTV to a cell phone. Similarly, when (re)distributing TV signals in a broadcast environment, transrating is used to steer the bit rate of individual channels before multiplexing, hereby keeping the total bit rate of the bundle of multiplexed TV channels constant. To deal with these varying transport net-

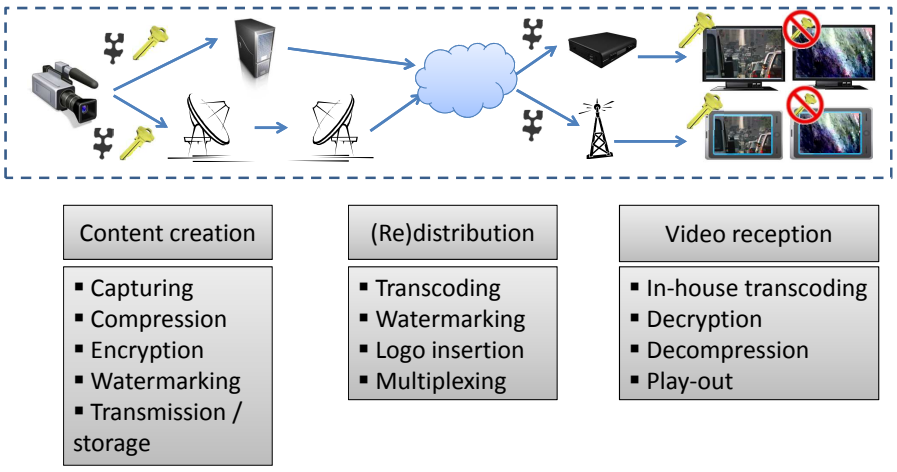


Figure 1: Example of a video distribution chain.

works and end user devices, appropriate video encoding technologies such as the

popular H.264/AVC standard and its proposed successor HEVC are required to handle the variable bandwidth conditions and error-prone network behavior.

Guaranteeing secure delivery of content to the consumer and beyond in such an heterogeneous environment therefore poses a number of practical hurdles. Not surprisingly, combined encryption and watermarking systems for the compressed domain are only sparsely covered in literature, e.g. [22, 27] for JPEG2000 images, [17] to adapt, encrypt and authenticate MPEG-21 & H.264/AVC video, whereas [26] is the most recent survey paper on protecting H.264/AVC video.

As indicated in the abstract, we thus need a flexible system that allows for commuting the encryption, watermarking and encoding/transcoding operations. Namely, the latter two should be (1) both applicable in the encrypted domain and (2) mutually compatible (i.e. transcoding shall not affect the watermarking and vice versa).

There are three approaches to realize requirement (1):

1. **Homomorphic encryption:** the data is fully encrypted and algebraic operations on the plaintext can be realized by performing a (possibly different) algebraic operations on the ciphertext, cfr. [12];
2. **Invariant encryption:** the data is fully encrypted but has invariant subsets (leaving room for signal processing thereon), e.g. the recent [23];
3. **Partial encryption:** only part of the data is encrypted (again leaving room for signal processing on the remaining set), cfr. [12].

Homomorphic encryption provides the most elegant solution, however, as explained in the tutorial paper [14], most efficient homomorphic schemes, e.g. [21], have a limited set of possible signal operations (the same holds for the invariant encryption approach), while current schemes that do offer a richer algebraic approach, e.g. [10], are not efficient. This basically prohibits the first two encryption systems to be used in transcoding scenarios. Regarding watermarking, in [11] the (multiplicative) homomorphic encryption properties of the RSA cryptosystem are combined with linear and additive watermarking algorithms in which the detection can be performed by correlation (for instance the so-called *spread spectrum technique* [7]), while in [13], it is shown that the commutativity of the encryption and watermarking operations can be weakened and an example for MPEG-2 video based on additive watermarking is presented and investigated.

Requirement (2) can be met as long as the watermark can be embedded compliant with the compressed domain or survives (i.e. is *robust against*) transcoding. The latter is exactly where the two previously mentioned combined encryption and watermarking systems fail. This basically leaves us with partial encryption as the (current?) path to follow.

Note that in broadcasting systems, audio distribution needs to be considered as well. Typically, compressed audio and video signals are multiplexed into a single container, e.g. an MPEG Transport Stream. Such a container can provide additional metadata, synchronization, and error correction for the encapsulated

streams. The audio signals in the container can additionally be secured, e.g. by using partial encryption schemes such as in [25]. However, in this paper we will concentrate on video.

In the next section, we first give a survey of the transcoding methodologies and both protection techniques we envision before we introduce our novel H.264/AVC & HEVC format-compliant partial encryption and robust watermarking system for secure video distribution. In the performance demonstration section we show that our encryption method effectively scrambles video streams and illustrate the performance of watermark embedding before and during encoding, along with the effect of applying transcoding operations to reduce the bit rate of the encrypted video streams.

Secure video distribution in practice

Encryption, watermarking, and transcoding solutions are strongly dependent on the underlying video coding standards that are used for video transmission. Over the last two decades, significant efforts have been spent on defining efficient video coding specifications. This led to a number of successful standards, in particular MPEG-2, H.264/AVC, and HEVC. The first version of H.264/AVC was finalized in 2003 by the Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, and was extended with several annexes and profiles since then. H.264/AVC supports a wide range of applications, bit rates and resolutions, and its efficiency led to wide adoption in broadcasting, over-the-top video, and mobile video distribution. H.264/AVC achieves a bit rate reduction of about 50% when compared to MPEG-2 at a similar quality level [30]. The High Efficiency Video Coding (HEVC) standard, scheduled to be finalized in early 2013, provides another leap in coding efficiency (a further bit rate reduction of 50% is targeted at the same visual quality as H.264/AVC High Profile) [16].

In Fig. 2, a typical architecture of an encoder is shown. This encoding loop structure is common to most state-of-the-art video coding schemes, including H.264/AVC and HEVC.

First, the uncompressed video frame is predicted, using either temporal (motion-compensated) information based upon previously encoded frames (reference frame(s)) and/or spatially causal information from the currently encoded frame (i.e. intra-prediction). The *prediction residual* (i.e. the difference between prediction and original frame) is subsequently transformed and quantized, which enables lossy encoding, and the final bit rate is controlled by a rate-distortion optimization mechanism. The resulting quantized coefficients are (i) further entropy coded and packetized in a bitstream that contains other syntax elements such as motion vectors and prediction modes and (ii) inversely quantized, transformed, added to the prediction, loop filtered to remove disturbing block artifacts and finally stored as a new reference frame. The closing of the prediction loop in such a codec is necessary to guarantee synchronization of the reference frame between the encoder and decoder. In the illustrated coding architecture (Fig. 2), the potential encryption and watermarking locations are

indicated respectively by 'E' and 'W'.

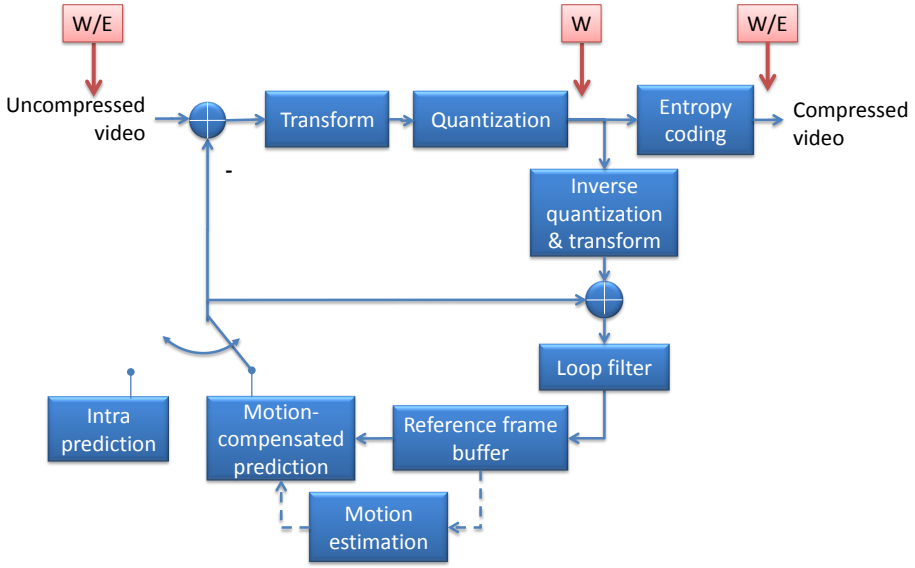


Figure 2: High-level encoder view.

One of the prerequisites for defining a format-compliant (partial) encryption algorithm that allows for operations such as watermarking and transcoding, is a classification of the data sets in the video bitstream, based on the knowledge of the video coding standard. Based on such a classification, an appropriate selection can be made of which data in the video bitstream is most suited for encryption, watermarking and transcoding. A similar strategy has been specified for the secured compression of JPEG 2000 images in [22]. In the case of H.264/AVC and HEVC, the streams roughly comprise information for so-called *prediction mode* signaling, *motion data*, and *residual (DCT) data*. The prediction modes give an indication of whether intra or inter-prediction is used, and which type and partitioning is used for each macroblock or coding unit. For inter-coded macroblocks, motion data is transmitted, consisting of reference picture indices and motion vector data. The residual data contains the prediction error after transformation and quantization. A suitable classification of bitstream elements will have an impact on the success of the encryption and the robustness of the watermarks.

Transcoding: classification

In general, transcoding aims at modifying the properties of a (video) bitstream, preferably with lower complexity than a combination of decoding and (time-consuming) encoding. Depending on the targeted application of the transcoder, we can distinguish between different adaptation operations, including *temporal*

(frame rate), *spatial* (resolution), and *bit rate transcoding* [29].

The most common type of transcoding operations for video streams is a reduction of the bit rate, also known as *transrating*, by reducing the precision of the information in the bitstream. Typically, this is achieved by increasing the quantization step size, called *quantization parameter* (QP), of the residual data (*requantization*) [8]. Another class of transrating techniques selectively removes residual coefficients from the bitstream (*dynamic rate shaping*) [9]. Both of these classes primarily target the residual data in the bitstream, while leaving other data unchanged. Note that when larger reductions of the bit rate are desired, not only residual, but also the motion data (such as motion vectors) can be adapted during transcoding.

A second type of adaptation is a reduction of the spatial resolution, which has a major impact on the bitstreams, and will change not only the residual data, but also the prediction modes and the motion data.

Third, frame rate reduction can be achieved by dropping frames, e.g. by a factor of two. When using hierarchical coding patterns in H.264/AVC or HEVC, this can easily be achieved, including in the semi-encrypted domain. The *scalable video coding* (SVC) extension of H.264/AVC can be used to add intrinsic scalability to video streams, by using a layered approach during encoding. In this way, quality or spatial layers can be dropped from the SVC stream, and the resulting subset can be decoded independently, resulting in a lower-quality or lower-resolution version. In this way, transcoding operations are reduced to simple 'cut-and-paste' operations, and decoding/encoding algorithms are avoided altogether. In contrast to H.264/AVC, however, SVC has not made a breakthrough in the broadcast world. Given its high computational complexity (in particular at the encoder side) and its bandwidth overhead (the introduction of extra layers increases the bit rate compared to H.264/AVC [24]), broadcasters are not eager to replace their existing equipment with SVC-compatible hardware or software. Although SVC provides a legitimate solution for secure video distribution, we focus on solutions for prevalent standards such as H.264/AVC. The encryption and watermarking approaches discussed in this paper for H.264/AVC can be readily extended to SVC (similar to e.g. in [28]).

Encryption

Oceans of choices for video scrambling

As discussed earlier, we focus only on partial encryption techniques. Based on where the encryption takes place, partial encryption methods can be categorized as in [26]. *Encryption before compression* techniques are codec-independent (indicated by the first 'E' position in Fig. 2) such as pixel position permutation but lead to less compressible videos. However, it might be an applicable choice for region-of-interest encryption. *Bitstream oriented encryption* approaches are more straightforward and thus can preserve less functionality (second 'E' position in Fig. 2). They encrypt the whole encoded bitstream (naive approach) or only a fraction of it (e.g. headers, different frame types, or the NAL unit pay-

loads) which can still allow compliant adaptation, packetization or even lower quality playback in case of multi-layered SVC. The *compression integrated encryption* approaches are codec specific by nature. At the expense of some loss of cryptographic security, they can preserve useful functionality such as format compliance, transcodability, enabling watermark embedding and so on. Numerous approaches have been reported that scramble the signs and/or the levels of the residual DCT coefficients and the motion vector differences, or a subset of these. Encrypting the intra and inter prediction modes can also destroy the structure of the image to certain degree. Alternative approaches have been proposed to scan the DCT coefficients in a secret order and even the Variable Length Coding (VLC) tables have been scrambled. A detailed survey on the approaches and their provided functionality can be found in [26], whereas [15] presents all the necessary background information.

Stream and block ciphers

As format compliance and transcodability are strict requirements in this work, the bitstream can be only partially encrypted. *Symmetric stream-* as well as *block ciphers* are good candidates for this purpose. The former ones can encrypt arbitrary amount of bits, the latter ones are block-based. However, there are numerous modes of operation defined for block ciphers, some of which make them behave as a stream cipher. This way the *Advanced Encryption Standard* (AES) [18] can be used in our system which grants high cryptographic security and renders the key unrecoverable by typical attacks such as known-plaintext- or ciphertext-only attack.

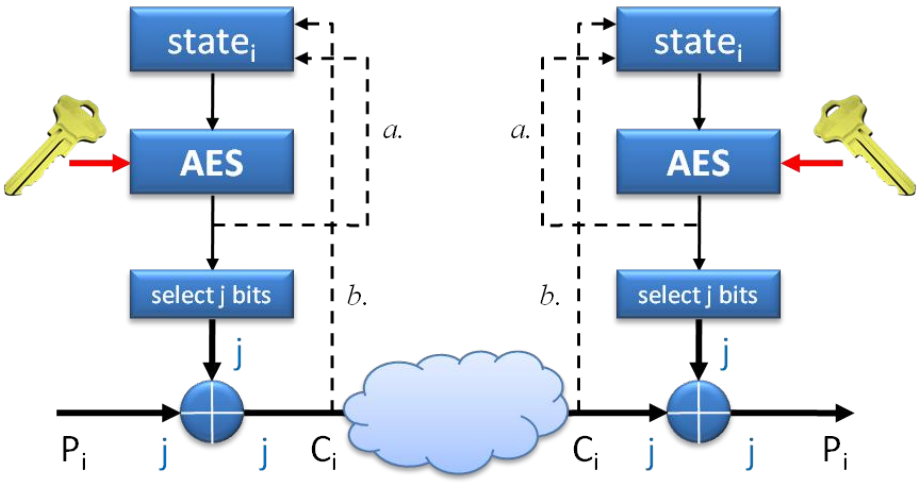


Figure 3: Encryption - decryption in: (a) Output feedback mode and (b) Cipher feedback mode.

Fig. 3 shows the principles of how the encryption works. The encryptable

data is considered as a continuous bitstream (P_i), each bit of which gets XOR-ed with a bit of a pseudo random sequence which is generated by a secure cipher (e.g. AES). If the same sequence is also generated at the decoder side and gets XOR-ed with the received ciphertext (C_i) then the two XOR operations cancel each other out, which renders the original plaintext. Since the pseudo random sequence depends on a key, the decryption is possible only for the entitled users. This key should be derived from the pre-shared long-term key and may change at an arbitrary interval. In this setup, we can request an arbitrary number of pseudo random bits (j) at each use which allows us to integrate the encryption part in a flexible way in the video codec wherever it is needed. Depending on what the input of the cipher is ($state_i$ in Fig. 3), there are several standardized modes of operation [19] that can be applied here: in *counter mode* a simple counter is fed to the AES which gets incremented after each AES call. In the *output feedback* and *cipher feedback* modes the output of AES or the ciphertext is used respectively. In the former two modes, the random sequence is completely independent from the data stream, thus even offline random sequence generation is possible, however, synchronization problems may occur. Cipher feedback mode is self-synchronizing but datastream-dependent.

Watermarking

Introduction to watermarking

Digital watermarking - the embedding of an imperceptible mark in the data - complements encryption in the sense that it can extend the protection of a multimedia item after its decryption. It allows the *embedding* of arbitrary information (*watermark*), indicated with a "puzzle piece" in Fig. 4, into digital media (images, video, audio) by applying imperceptible, systematic alterations to the data (*coverwork*) depending on a key, which is needed at the *detector*.

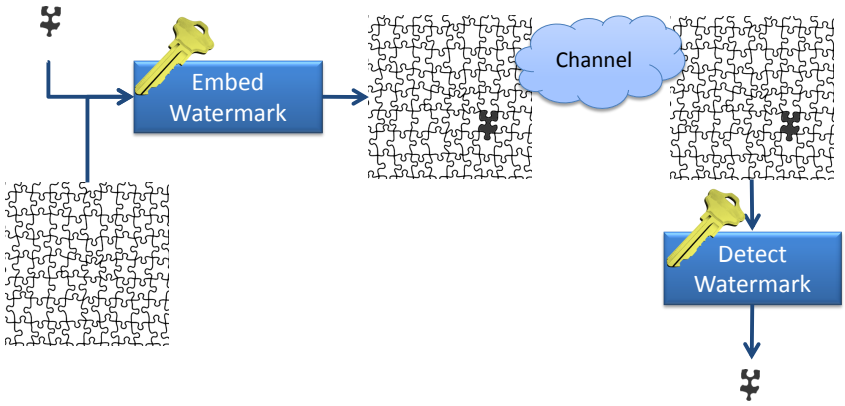


Figure 4: Watermark embedding and detection.

In a *blind detection* system, the decoding function takes the received (possibly attacked) watermarked signal and a key to produce an estimate of the

watermark. In the *non-blind or informed detection* system, the decoding function in addition has access to the original host signal, which increases detection performance, but creates a communication and storage burden in practice.

Research in watermarking emerged in the 1990s, and in the meantime numerous practical systems have been published and theoretical bounds have been achieved. An excellent in-depth overview on the theory and security aspects of watermarking systems in general can be found in the tutorial paper by Moulin and Koetter [20] and in the book [7].

Any watermarking scheme is subject to the trade-off between its perceptual impact, robustness against signal processing operations and/or malicious attacks and the amount of information (*payload*) that can be transmitted reliably within the coverwork.

Lattice Quantization Index Modulation

In this paper, we chose to employ *Quantization Index Modulation* (QIM) watermarking, introduced by Chen and Wornell in 1998 - a superior (substitutive) technique in an information-theoretical sense [5] for blind detection. The QIM-watermarking system is based upon a good choice of a set of quantizers, which allows one to vary from a so-called *fragile* (designed to be easily destroyed if the watermarked image is manipulated in the slightest manner), over *semi-fragile* (designed to degrade under "unwanted" attacks) to a *robust* (designed to resist attempts to remove or destroy the watermark) watermarking technique depending on the stepsize or *strength* parameter Δ.

In Fig. 5 (a), we depicted a scalar quantizer with stepsize $\Delta/2$, which is split into two shifted coarse quantizers with stepsize Δ in order to embed one bit in a (real) sample s taken from the coverwork. These samples can be (luminance) pixel values, however for the sake of imperceptibility and robustness it is advised to work in a transform domain, like the DCT or DWT, where one easily can select a range of coefficients with low visual impact that are less vulnerable under attacks. To embed a 0-bit, the sample s is quantized to the value associated with the nearest symbol with label 0 (represented by a circle), while for a 1-bit we move s to the value associated with the nearest symbol with label 1 (represented by a cross). Given a watermarked, possibly attacked, sample s, we detect the embedded bit as the label of the nearest symbol of the fine quantizer, what is referred to as a *minimum distance decoder*. A classical example of the fine quantizer is $\Delta\mathbb{Z}$, where we use the even multiples of Δ for the circle quantizer, whereas his coset - the odd multiples - plays the role of the cross quantizer.

Most of the time, scalar quantizers are employed because of their simplicity, although they are outperformed by lattice quantizers [20]. Recall that a *lattice* (in $\mathbb{R}^n$) is defined as a collection of vectors that are integral combinations of a set of basis vectors in $\mathbb{R}^n$. We point to [6] as the reference work on properties of lattices and associated quantizers. In Fig. 5 (b), the centers of the dotted hexagons form a so-called *hexagonal lattice* for which the ones indicated with 01 and 10 can be seen as basis vectors. Similarly to the scalar case, this fine lattice is split into four similar coarse hexagonal lattices: one is given by the

centers of the solid hexagons (such as 00), while the other three are shifted over the coset leaders indicated with 01, 10 and 11. The associated quantizers can now be used to embed two bits (resp. 00, 01, 10 and 11) into samples taken from $\mathbb{R}^2$.

It is easily seen that the robustness of a lattice QIM system depends on the distance between the coarse lattice coset leaders in which a lattice can be split, which on its turn impacts the perceptibility. The trade-off between robustness and perceptibility of lattice QIM is therefore related to the *sphere-packing problem* of lattices in Euclidean space. In [1], we developed a methodology to create parametrized lattice QIM systems based on so-called *self-similar* lattices, where we relate the *rate*, i.e. the number of embedded bits per vector of coverwork samples, with the number of cosets in which we can split the fine lattice through rotation and scaling. We employed, in particular, this technique to the *Gosset* lattice E_8, which is the subgroup of vectors in $\mathbb{R}^8$ for which the coordinates are all in $\mathbb{Z}$ or all in $\mathbb{Z} + \frac{1}{2}$ and their sum is even. This lattice is self-similar and optimal for the sphere-packing problem in 8 dimensions. We showed that the resulting lattice QIM system has high robustness at the cost of a low perceptual impact together with flexible payload possibilities.

Watermarking for the compressed domain

Concerning compressed-domain watermarking, there are basically three places in a video encoder loop (pictured in Fig. 2) to perform watermark embedding:

1. **Pre-encoding watermarking:** Watermarking can be applied prior to encoding, on the uncompressed data. Most image or video watermarking schemes operate in this way, in which case the video encoding is considered an attack which can harm the watermark. Depending on the quality of the encoded stream (determined by the quantization in the encoder), the watermark can be damaged or removed.
2. **Inter-encoding watermarking:** By adding the watermark in the encoder loop (indicated as the second 'W' position), the watermark can be inserted in (already quantized) data in the encoder, hereby exploiting properties of the encoded bitstream. In this case, the encoding itself is no longer an attack to the watermark (see e.g. [3] for a survey article on watermarking in the H.264/AVC compressed domain).
3. **Post-encoding watermarking:** Furthermore, the watermark can be added outside the encoder loop, either in the encoder, or at a later stage in the video distribution (e.g. as a transcoding step). Note that the addition of a watermark outside of the encoder loop has to be done cautiously, since changes caused by the watermark can accumulate over time and introduce drift throughout the video stream.

Obviously, the video quality will be affected proportional to the strength of the watermark, reducing the overall encoding performance irrespective to the location where the watermark is applied in the distribution chain.

A novel H.264/ AVC & HEVC format-compliant encryption and watermarking system

In case of partial encryption, there are a number of components in the bitstream that can be encrypted and watermarked. Fig. 6 gives a high-level perspective on our proposed combined protection system based on the inter-encoding watermarking scenario, which is novel according to the general methodologies described in the survey [26]. The input frames on the left side (either intra-coded (I) or inter-coded (P or B) frames) contain several data sets, which encompass parameters, prediction mode information (for intra or inter-prediction modes), motion vectors and residual data. The data sets that are affected by encryption are indicated with a "key" image, while the residual coefficients, indicated with a "puzzle piece", are watermarked.

We chose to encrypt those components that do not disable rate change. Encrypting the *intra-prediction modes* and the *sign bits of the DCT coefficients* takes care of all the intra-blocks whereas changing *inter-prediction modes* and *sign bits of the motion vector differences* scrambles inter-predicted parts. Sign bit encryption refers to a possible sign bit change whereas encrypting the modes implies changing the actual mode to another one without violating the semantics and bitstream compliance. As the four data sets are completely independent from each other, they can be selectively encrypted. For example, intra-prediction modes are interchangeable in general, but not all modes are available along the top and the left borders of each frame due to the lack of neighbors. In case of inter-prediction modes, only 8×16 and 16×8 partitions are interchangeable since the other partition types require a different number of motion vectors, hence changing them would lead to undecodable video. Due to the fact that encryption occurs in the final, output bitstream (outside of the encoding loop), no bit rate increase arises.

For watermarking the residual DCT coefficients, we chose to employ our E_8-lattice QIM system, as it displays good robustness (needed for transcoding and possible other video processing attacks after delivery), low perceptibility (so that it hardly affects the quality of valuable content), while offering flexibility in payload (so that it can be adapted to a specific application in mind: copyright protection, traitor tracing, authentication, quality assessment [4] etc.). and blind detection (the original might not be at hand).

During transcoding, the encrypted data sets of the input video stream will be simply copied to the output stream without interfering with the encryption. The residual coefficients containing the watermark, however, will be affected by transcoding. Transcoding approaches will either *requantize* (leading to a coarser approximation of the coefficients) or *selectively remove* coefficients (by clipping e.g. high-frequency transform coefficients) to reduce the bit rate.

Performance demonstration

In this section, we demonstrate the feasibility and constraints of a system combining encryption, watermarking, and transcoding in an end-to-end video dis-

tribution system. Because the previously described encryption strategy only limits the design of the watermarking and transcoding algorithms and does not influence the performance of those techniques, encryption performance will be evaluated first. Then, the impact of video compression and transcoding on the watermark will be demonstrated.

To evaluate the performance of our implemented architecture, a sample set of 22 video sequences with varying properties (corresponding to the test set used in HEVC standardization [2], with sequences ranging from WQVGA to 2560×1600 resolution) were analyzed both visually and objectively after encryption, watermarking, and transcoding. The video streams were compressed at representative bit rates, in line with the coding conditions used in standardization.

Encryption

Encryption takes place during the encoding process on the video stream elements indicated in Fig. 6. In general, compression at a higher bit rate generates more residual data thus proportionally more elements to encrypt. The total amount of encryptable data varied between 19% and 50% of the bitstream in our test set. We used two objective quality metrics to measure the effectiveness of the partial encryption algorithm.

Peak signal-to-noise ratio (PSNR) is the most commonly used method to measure quality degradation. It is based on the mean of the squared difference of two images. Most sources in this field evaluate the encrypted videos based on comparing the PSNR values. However, a lower PSNR does not necessarily correspond to a more scrambled frame.

Although the Structural Similarity Index (SSIM) has been used less frequently in publications to assess the encryption performance, we have found this metric more meaningful than PSNR in our application. The SSIM index of two windows x and y (typically of size 8×8) is defined as:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \ , \tag{1}$$

where μ, σ are window averages and (co)variances respectively, making this technique less sensitive to noise. The C_i values refer to constants that depend on the bit-depth of the image. This metric was designed to take into account that spatially close pixels have strong dependencies which is also referred to as *structural information*. Since the structural difference of two images is exactly what we want to measure when assessing encryption techniques, we have decided to rely on this metric. Its output is a real number between 0 and 1, where a larger number means higher similarity. In our tests, the average scores showed that the structure of the frames could be sufficiently degraded in case of both codecs. H.264/AVC produced scores between 0.16 and 0.2, whereas the HEVC tests ended with even lower values that fell between 0.06 and 0.13 as seen in Fig. 7 (which shows average results for all test sequences).

Since every I/B/P block is affected in some way by the encryption, the variance of the resulting PSNR and SSIM values is consistently low throughout the whole video.

Although we present average scores in this work, it is important to mention that not every video can be degraded to the same extent. The quality of sharp, high-motion content becomes much more scrambled after encryption than that of low-motion video with a static background (e.g. in video conferencing or remote desktop scenarios).

Both H.264/AVC and HEVC encode only the residual (difference) between the actual and predicted pixel values (as was shown in Fig. 2). In case of accurate prediction, the energy contained in the residual is small, leaving limited room for encryption. The same holds for the motion vectors, where only the difference between the actual motion vector and a motion vector predictor is coded (derived from the motion vectors of neighboring blocks). Therefore, if there is only limited motion in the video, the magnitude of the difference is not large either. Hence, little change is induced when the encryption flips the motion vector signs, making the shapes slightly visible in extreme cases. The lower the quality of the encoded video, the more the frames get averaged out which blurs static backgrounds. Such flat areas require minimal data to encode, therefore minimal number of changes can be induced by encryption which might lead to information revelation. This phenomenon is demonstrated in Fig. 8, where the edge detected frames help to compare the output.

In general, it can be stated that for entertainment purposes scrambling performance of the described encryption system is more than adequate.

Watermarking and compression

In this demonstration, both watermarking before encoding and during encoding are investigated.

First, we applied the pre-encoding watermarking scenario, after which a video encoder compresses the video information with a certain quality loss. This quality reduction encompasses both video and watermark information loss, measured by the PSNR and the bit error rate (BER) for watermark detection, respectively.

We embedded 256 bits per frame in the mid-frequency DCT domain with varying strengths using the E_8-lattice at a rate of 4 bits per 8 coefficients. The compression impact is graphically represented in Fig. 9 for a representative sequence (the ParkScene test sequence with full HD (1080p) resolution, which contains a combination of panning and motion). The curves were obtained by coding at four different quality settings (QP values of 22, 27, 32, and 37). Similar results were obtained for the other test sequences.

It is clear that video compression at higher quality results in lower bit error rates for the watermark detection; intuitively, a lower watermarking strength results in higher bit error rates after compression. Note that when using error-correction codes (e.g. Turbo codes) a BER lower than about 10% may be successfully corrected at the cost of a larger payload (which is not a hurdle for the

DQP	H.264/AVC						HEVC					
	$\Delta = 6$		$\Delta = 12$		$\Delta = 18$		$\Delta = 6$		$\Delta = 12$		$\Delta = 18$	
	BER	PSNR	BER	PSNR	BER	PSNR	BER	PSNR	BER	PSNR	BER	PSNR
1	0.00	56.41	0.00	56.14	0.00	57.32	0.00	54.83	0.00	46.04	0.00	49.06
2	0.00	50.78	0.00	51.14	0.00	51.62	0.00	49.09	0.00	43.00	0.00	46.21
3	0.00	48.43	0.00	49.30	0.00	49.28	0.00	48.85	0.00	42.74	0.00	46.26
4	0.00	45.46	0.00	45.68	0.00	45.61	0.00	42.03	0.00	39.60	0.00	40.76
5	0.17	42.78	0.11	42.83	0.08	42.88	0.02	42.15	0.01	39.54	0.00	41.08
6	0.17	42.44	0.11	42.62	0.08	42.68	0.02	42.04	0.01	39.48	0.00	40.93
8	0.17	42.19	0.11	42.46	0.08	42.53	0.02	41.62	0.01	39.00	0.00	40.31
10	0.25	41.95	0.11	42.07	0.08	42.11	0.19	41.38	0.01	38.59	0.00	40.21

Table 1: Watermark bit error rate (BER) and PSNR [dB] results after transcoding with different $DQP = QP_{out} - QP_{in}$ values, applied to watermarking with different strengths ($\Delta = 6$, 12, or 18).

watermarking technique employed). Finally, we note that when the same video is compressed to the same quality (as measured by PSNR reduction) by both HEVC and H.264/AVC, we observe similar BER trends, but less watermarking information survives under HEVC. This is caused by the more advanced coding modes introduced by HEVC and the resulting higher decorrelation of the signal (or entropy reduction), which makes it more prone to bit error sensitivity.

In a second experiment, inter-encoding watermarking is applied, indicated as the second 'W' position in Fig. 2. When applying a watermark during or after the compression process there is no negative impact of the compression itself on the watermark. The watermark still slightly reduces the picture quality, but this time the compression does not form an attack on the watermark. Possible bit errors can only be introduced when transcoding operations are applied afterwards. Similar to the previous experiment, we embedded 256 bits per frame with varying strengths, this time on the transformed and quantized residual data. Fig. 10 shows the impact on the rate-distortion performance by inserting watermarks with an example strength of 18 in both H.264/AVC and HEVC. For the highest rate point, a maximum quality loss of about 0.6 dB in PSNR is obtained for H.264/AVC. Because the watermarking process at this location is 'in the loop', rate-distortion optimization (RDO) will keep the introduced loss low, by carefully selecting blocks which are affected to a minimal extent by the watermark. Note that this is not the case when introducing the watermark at the third 'W' position (outside the loop, or after encoding). In this case, the locations for watermark insertion have to be carefully evaluated, since they can have a significant impact on the bit rate, or introduce drift in the video stream when errors in the bitstream accumulate.

Impact of transcoding

After encryption and watermark insertion, we subjected the bitstreams to a transcoding process, where the residual data in the bitstreams was parsed, requantized with a coarser quantization step size, and (entropy) coded again in the output video bitstream. This resulted in a lower bit rate, and unavoidably a

lower quality of the output streams. Previous research has indicated that drift caused by requantization of intra-coded blocks in the bitstreams has a major impact on the quality of the transcoding video [8]. For this reason, we apply requantization only to inter-coded macroblocks or coding units. Depending on the sequence, bit rate reductions of approximately 5-40% were tested, which corresponds to realistic transcoding scenarios.

Note that in all cases the encryption was untouched by the transcoding process, supporting the commutative property of our combined system. The impact on the watermark is illustrated in Table 1, showing that watermark embedding in HEVC is less sensitive to transcoding than watermarking in H.264/AVC. This is explained by the more efficient prediction modes and the RDO process of HEVC, which is highly selective in the locations in which watermarking bits can be embedded. Since watermarking is applied in the loop, the RDO process of the encoder will only decide to insert bits in regions that contain more residual energy (and larger coefficient magnitudes). Accordingly, since HEVC has more advanced prediction algorithms than H.264/AVC, fewer bits can be potentially embedded, but they will more easily survive requantization attacks.

Conclusions

End-to-end video security introduces several challenges that can be tackled when tailoring cryptography and signal processing operations to each other. We presented the use of partial encryption techniques in a trade-off between security and preserved functionality.

The proposed encryption of a combination of data sets in H.264/AVC and HEVC achieves consistently low SSIM values throughout the encrypted video streams, showing the effectiveness of the scrambling operation. Nonetheless, when considering the encryption, a few elements affect its performance such as homogeneous backgrounds and the absence of motion. In certain applications (e.g. video conferencing) these factors cannot be eliminated so somewhat lower security can be granted. However, the proposed system provides ample security in large application areas such as video broadcast and pay-per-view services.

Two important signal processing operations in secure video distribution chains (watermarking and transcoding) were shown to be commutative with the partial encryption scheme. The additional watermarking protection layer offers enough flexibility to be applied before or during the encoder/encrypting loop due to the more than satisfying trade-off between robustness, perceptibility and payload of the E_8-lattice based QIM-watermarking system we employed. A limited overhead in rate-distortion performance is induced for watermarking in the compressed domain.

The effect of transcoding on embedded watermarks was demonstrated for both H.264/AVC and HEVC, which shows that typical bit rate adaptations can be performed with limited impact on the BER of the watermark.

Acknowledgements

This research was supported by the ongoing WET project of Fonds Wetenschappelijk Onderzoek (FWO). Some of our results were achieved within the context of the AQUA and OMUS projects of the Interdisciplinary Institute for Broadband Technology (IBBT) and the DaVinci project of IMPact.

References

[1] D. Bardyn, A. Dooms, T. Dams and P. Schelkens. "Comparative study of wavelet based lattice QIM techniques and robustness against AWGN and JPEG attacks". Proc. 8th Int. Workshop on Digital Watermarking. Vol. 5703, pp. 39–53, 2009.

[2] F. Bossen. "Common test conditions and software reference configurations", ITU-T SG16 WP3 (VCEG) and ISO/IEC JTC1/SC29/WG11 (MPEG) doc. JCTVC-H1100, San Jose, CA, USA, February 2012.

[3] S. Bouchama, H. Aliane and L. Hamami. "Watermarking Techniques Applied to H264/AVC Video Standard", International Conference on Information Science and Applications (ICISA), pp.1-7, 2010.

[4] G. Braeckman, A. Barri, G. Fodor, A. Dooms, J. Barbarien, P. Schelkens, A. Bohó and L. Weng. "Reduced Reference Quality Assessment based on Watermarking and Perceptual Hashing", Sixth International Workshop on Video Processing and Quality Metrics for Consumer Electronics, Scottsdale, Arizona (USA), January 2012.

[5] B. Chen and G.W. Wornell. "Quantization Index Modulation: a class of provably good methods for digital watermarking and information embedding". IEEE International Symposium on Information Theory, p. 46, June 2000.

[6] J. H. Conway and N. J. A. Sloane. "Sphere Packings, Lattices and Groups". Springer, New York, 1999.

[7] I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker. "Digital Watermarking and Steganography", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[8] J. De Cock, S. Notebaert, P. Lambert and R. Van de Walle. "Requantization Transcoding for H.264/AVC Video Coding", Signal Processing: Image Communication, vol. 25, no. 4, pp. 235-254, April 2010.

[9] A. Eleftheriadis and P. Batra. "Dynamic rate shaping of compressed digital video". IEEE Transactions on Multimedia. Vol. 8 (2), pp. 297–314, April 2006.

[10] C. Gentry and S. Halevi. "Implementing Gentrys fully-homomorphic encryption scheme", Cryptology ePrint Archive, Report 2010/520, 2010, http://eprint.iacr.org/.

[11] K. Gopalakrishnan, N. Memon and P.L. Vora. "Protocols for watermark verification". IEEE Multimedia, Vol. 8 (4), pp. 66–70, October 2001.

[12] J. Herrera-Joancomartí, S. Katzenbeisser, D. Megías, J. Minguillón, A. Pommer, M. Steinebach and A. Uhl. "ECRYPT European Network of Excellence in Cryptology, first summary report on hybrid systems, D.WVL.5", 2005.

[13] S. Lian. "Quasi-commutative watermarking and encryption for secure media content distribution". Multimedia Tools Appl., 43(1), pp. 91–107, 2009.

[14] R.L. Lagendijk, Z. Erkin and M. Barni. "Encrypted Signal Processing for Privacy Protection". IEEE Signal Processing Magazine, January 2013.

[15] S. Lian. "Multimedia Content Encryption: Techniques and Applications". Auerbach Publications, 2009.

[16] B. Li, G. J. Sullivan and J. Xu. "Compression Performance of High Efficiency Video Coding (HEVC) Working Draft 4". IEEE International Symposium on Circuits and Systems (ISCAS), May 2012.

[17] T. Lookabaugh and D.C. Sicker. “Selective encryption for consumer applications”. IEEE Communications Magazine. Vol. 42(5), pp. 124–129, May 2004.

[18] NIST, Advanced Encryption Standard (AES) FIPS Publication 197, November 2001.

[19] NIST, Recommendation for Block Cipher Modes of Operation: Methods and Techniques Special Publication 800-38A, December 2001.

[20] P. Moulin and R. Koetter. “Data-Hiding Codes”. Proceedings of the IEEE. Vol. 93 (12), pp. 2083-2126, December 2005.

[21] P. Paillier. “Public-key cryptosystems based on composite degree residuosity classes”, in Advances in Cryptology EUROCRYPT 99. 1999, vol. 1592 of Lecture Notes in Computer Science, pp. 223238, Springer- Verlag. optimization”. IEEE Transactions on Circuits and Systems for Video Technology. Vol. 18 (6), pp. 746-755, June 2008.

[22] P. Schelkens, A. Skodras and T. Ebrahimi. Eds. “The JPEG 2000 Suite”. Hoboken, NJ: Wiley, 2009.

[23] R. Schmitz, S. Li, C. Grecos and X. Zhang. “A New Approach to Commutative Watermarking-Encryption”, 13th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security (IFIP CMS 2012), September 3-5, 2012, Canterbury, UK, Lecture Notes in Computer Science by Springer, 2012.

[24] H. Schwarz, D. Marpe and T. Wiegand. “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”. IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[25] A. Servetti and J. C. De Martin. “Perception-Based Partial Encryption of Compressed Speech”. IEEE Transactions on Speech and Audio Processing, vol. 10, no. 8, November 2002.

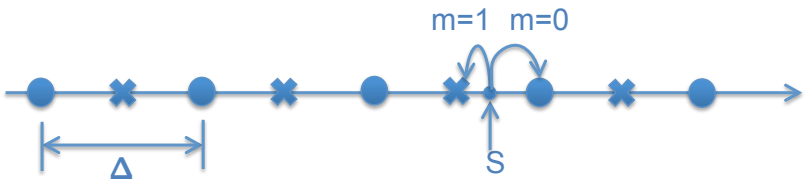
[26] T. Stütz and A. Uhl. “A Survey of H.264 AVC/SVC Encryption”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 3, March 2012.

[27] A.V. Subramanyam, S. Emmanuel and M.S. Kankanhalli. “Robust Watermarking of Compressed and Encrypted JPEG2000 Images”, IEEE Trans. on Multimedia, Vol. 14, No. 3, pp. 703- 716, June 2012.

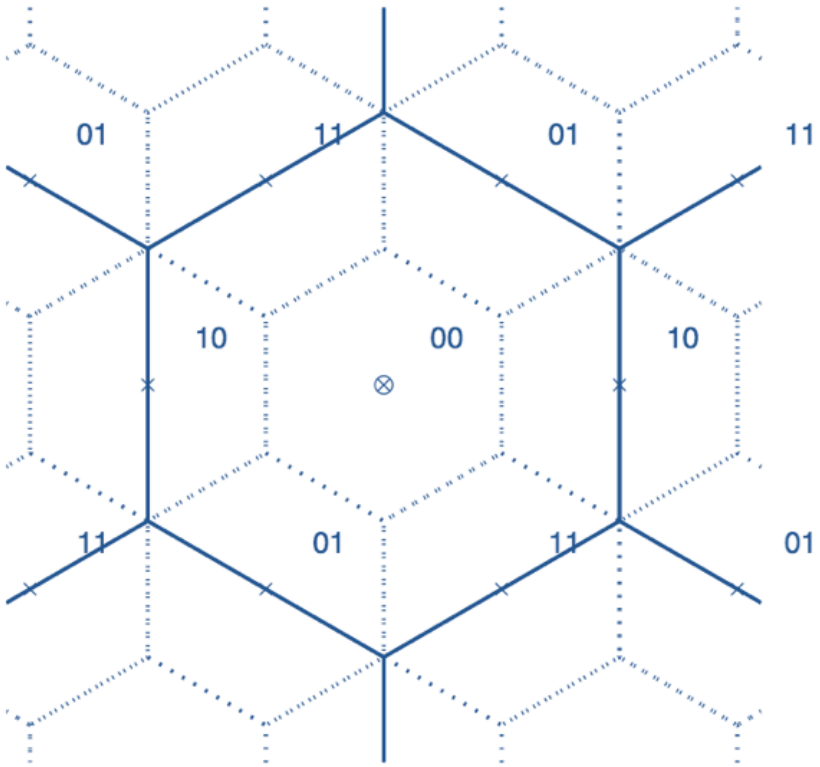
[28] N. Thomas, D. Bull and D. Redmill. “A novel H.264 SVC encryption scheme for secure bit-rate transcoding”. Proc. 27th Picture Coding Symposium (PCS), pp. 157–160, May 2009.

[29] A. Vetro, C. Christopoulos and H. Sun. “Video transcoding architectures and techniques: an overview”, IEEE Signal Processing Magazine, vol. 20, no. 2, pp. 18-29, Mar. 2003.

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



(a)



(b)

Figure 5: Quantizers. (a) Two shifted scalar quantizers with strength Δ to embed one bit in a sample s. (b) Hexagonal lattice with 4 coset leaders.

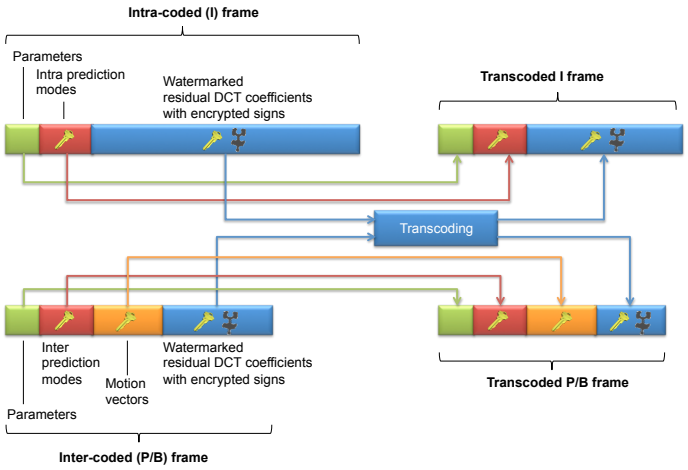


Figure 6: Interaction of transcoding with encrypted and watermarked bitstreams (for intra and inter-coded frames).

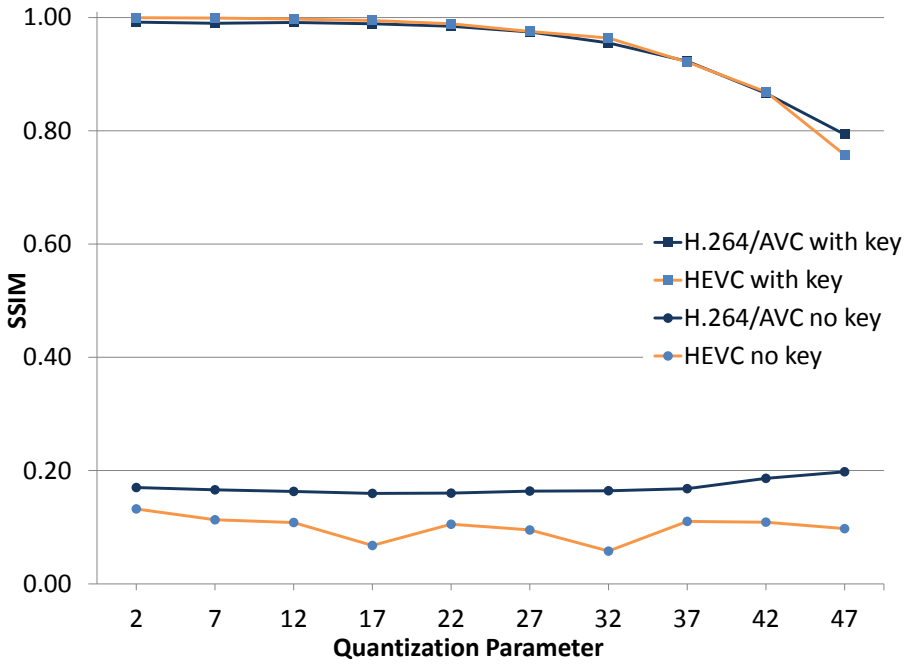


Figure 7: Average SSIM scores

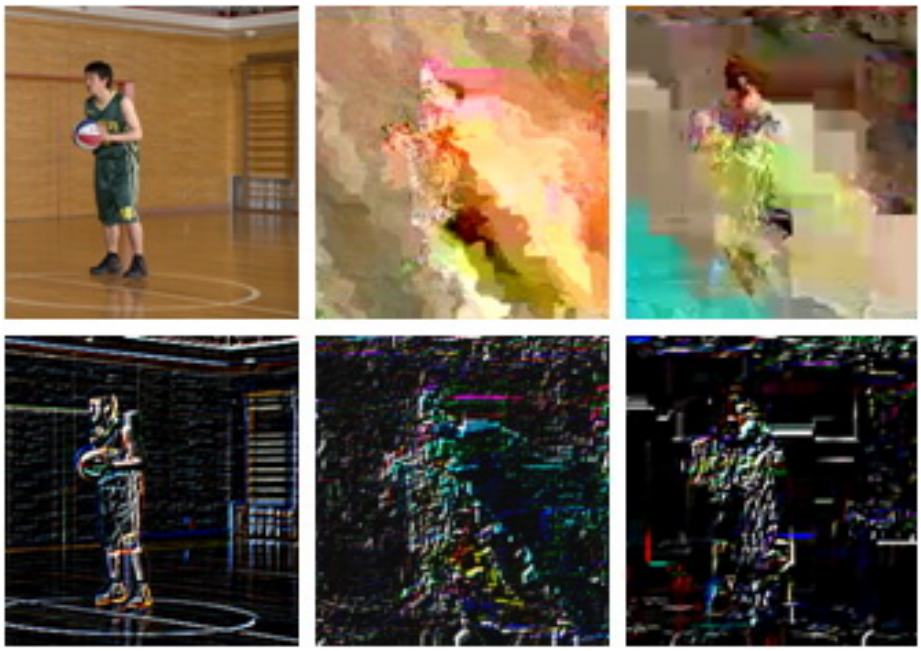


Figure 8: Comparison of the original (left) and the encrypted videos at QP=12 (middle) and QP=42 (right) along with their corresponding edge detected versions (bottom).

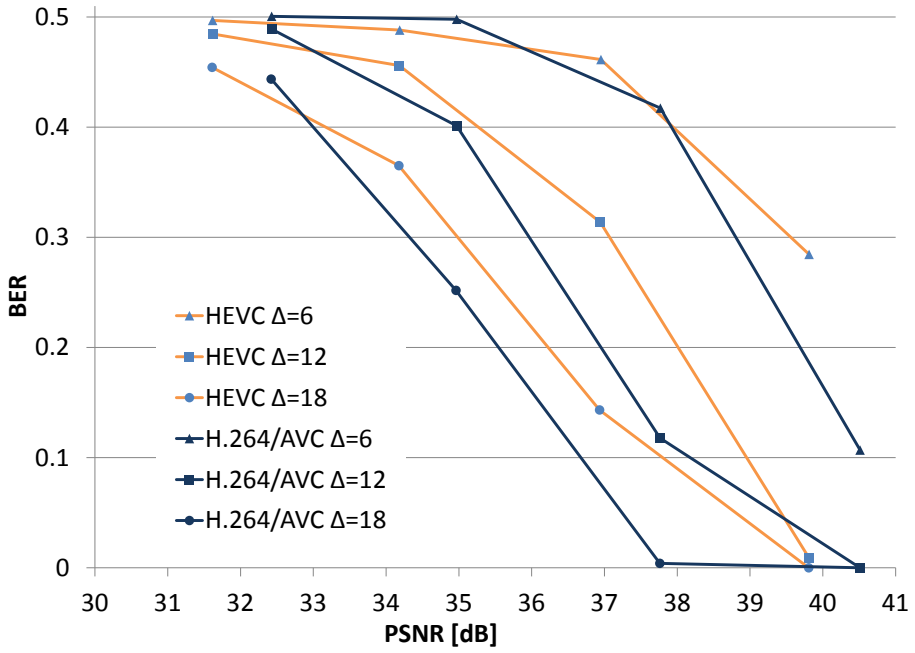


Figure 9: Bit error rate of the watermark detection with an indicated strength (Δ) of 6, 12, or 18 after H.264/AVC or HEVC compression.

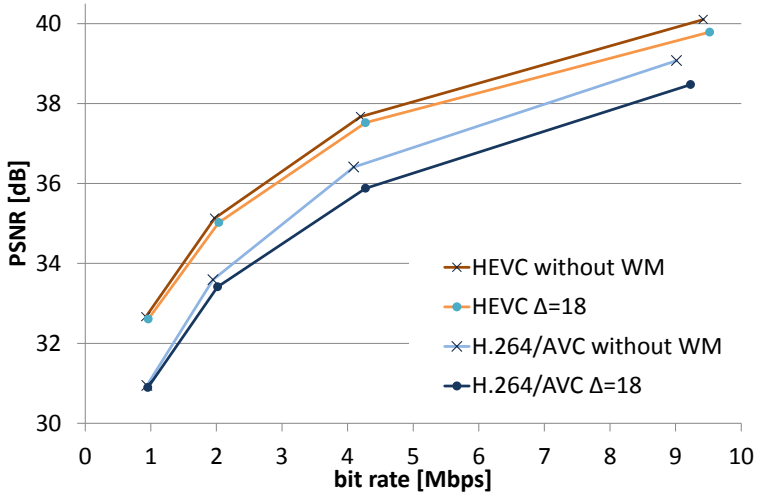


Figure 10: Compression efficiency (rate-distortion) results for H.264/AVC and HEVC watermarking with an example strength (Δ) of 18.