Multimedia Tools and Applications (2019) 78:21455–21484
https://doi.org/10.1007/s11042-019-7451-5

Effectiveness of crypto-transcoding for H.264/AVC and HEVC video bit-streams

Rizwan A. Shah[1] · Mamoona N. Asghar[1] · Saima Abdullah[1] · Martin Fleury · Neelam Gohar[3]

Received: 8 June 2018 / Revised: 22 January 2019 / Accepted: 6 March 2019 /
Published online: 22 March 2019

Abstract

To avoid delays arising from a need to decrypt a video prior to transcoding and then re-encrypt it afterwards, this paper assesses a selective encryption (SE) content protection scheme. The scheme is suited to both recent standardized codecs, namely H.264/Advanced Video Coding (AVC) and High Efficiency Video Coding (HEVC). Specifically, the paper outlines a joint crypto-transcoding scheme for secure transrating of a video bitstream. That is to say it generates new video bitrates, possibly as part of an HTTP Adaptive Streaming (HAS) content delivery network. The scheme will reduce the bitrate to one or more lower desired bit-rate without consuming time in the encryption/decryption process, which would be the case when full encryption is used. In addition, the decryption key no longer needs to be exposed at intermediate middleboxes, including when transrating is performed in a cloud datacenter. The effectiveness of the scheme is variously evaluated: by examination of the SE generated visual distortion; by the extent of computational and bitrate overheads; and by choice of cipher when encrypting the selected elements within the bitstream. Results indicate that there remains: a content; quantization level (after transrating of an encrypted video); and codec-type dependency to any distortion introduced. A further recommendation is that the Advanced Encryption Standard (AES) is preferred for SE to lightweight XOR encryption, despite it being taken up elsewhere as a real-time encryption method.

Keywords Content protection · H.264/AVC · HEVC · Selective encryption · Transcoding Transrating · Video streaming

✉ Martin Fleury
fleury.martin55@gmail.com

[1] Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

[2] Formerly School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

[3] Department of Computer Science, Shaheed Benazir Bhutto Women University, Peshawar, Pakistan

1 Introduction

Video content can be streamed to a device for viewing without the need for other than partial storage on the device, though the video stream is stored on a server and possibly encrypted as well. For example, using the most mature form [5] of HTTP Adaptive Streaming (HAS), that is HTTP Live Streaming (HLS), a client device can dynamically select from different representations of the stream according to available bandwidth. Each of these versions might be first transcoded [64] to one of (say) eight versions, with a ninth audio stream. Transcoding might be through changing the spatial resolution, the temporal resolution, or the bitrate, which in effect involves a reduction in the Signal-to-Noise Ratio (SNR), which is commonly interpreted as a reduction in video quality. This paper assesses the effectiveness of changing the bitrate, i.e. transrating, when that bitstream is encrypted. Transcoding may also involve changing the video format, for example between that of one codec to another, though this is usually not required for HAS and is not assessed herein. Prior to transcoding for HLS, the original video stream might be delivered to a Content Delivery Network (CDN) by Real-Time Messaging Protocol (RTMP). Such a CDN, might be part of a CDN as a Service (CDNaaS) [65] cloud-based offering. As remote processing is involved, additional security issues arise if content is not protected. Unfortunately, content protection through encryption results in additional latency at intermediate transcoders due to the need for decryption. The aim of this paper is to avoid that additional latency by allowing the video to be transcoded without decryption taking place. At the same time, the paper seeks to establish whether accelerated forms of encryption/decryption can further reduce the latency, without significantly affecting the content's security. This is a timely study, given the increased prevalence of transcoding in the mobile Internet, which was not around when transcoding was originally applied to statistical multiplexing of broadcast video.

Encryption allows protection of content against illegal access to a video server and also protects each video stream during transport across the Internet. Content protection is deemed necessary to ensure the commercial viability of a video streaming service because otherwise there would be no monetary incentive to make videos and distribute them. For example, the HLS specification [48] supports full encryption of video segments within a representation using Advanced Encryption Standard (AES) key length 128 and the Encrypted Media Extensions (EME) W3C specification describes key management for HTML5 video [17]. RTMPE also supports full encryption through the Rivest Cipher 4 (RC4) stream cipher. As RC4 is now considered cryptographically insecure by the Internet Engineering Task Force (IETF) [51] and, as a result, is not included in browsers such as Internet Explorer 11, alternatively, the RTMP video stream can be wrapped within a Transport Layer Security (TLS) session. However, these forms of full encryption are sometimes called naïve encryption because they do not exploit the features of video, essentially treating it as text. Consequently, it may result in a significant performance overhead, especially if software-only encryption occurs [14].

However, compared to full encryption, with selective encryption (SE) [40] not only is the amount of data encrypted reduced but some forms of SE are decoder format-compatible. If the form of SE is not format compatible, as occurred in [67] for the I-frame encryption option, then the transcoder has to be modified. As transcoder modification reduces the generality of the solution, requiring all transcoders to be modified, this paper only considers decoder compatible forms of SE. A similar observation applies to secured hardware transcoders such as that of the Secure Video Processor (SVP) alliance [55], which may also be costly compared to an

unsecured transcoder. Thus, it is possible to ensure the video is unwatchable owing to distortions, while at the same time permit bitrate transcoding without decryption. For example, in the RTMP delivery of the original video to a CDNaaS, were SE to be employed, there would be no intermediate storage of the decryption key, which is only held by the client device. As previously mentioned, the client device also does not store a streamed video, unless deserialization software has illicitly been installed. In addition, the selectively encrypted components can be encrypted by a standardized cipher such as AES operating in a streaming mode such as Cipher Feedback (CFB). After encryption, those encrypted components are normally replaced in the video bitstream in the interests of format compatibility.

Apart from its use in HAS, video content can be further compressed by transcoding to reduce the bit-rate of the video in order to match the capability of a user's device, such as its processing capacity, which will affect the viable display frame rate. Spatial resolution is the other main factor that can reduce the bitrate of the original video content. The resolution of a user's device may be as low as Common Intermediate Format (CIF), as high as 1280×720 pixels/frame (High Definition or HD), or even Ultra HD (UHD) resolution [1]. However, though spatial resolution switching does occur in HAS systems, in [58] it was recommended that the spatial resolution of the target device is first adjusted for, after which different quality representations are selected by a client device. In [23], quality switching through encoding was found to be the most common form of representation considered in research. Thus, this paper considers quality transcoding, especially as temporal switching, though effective in terms of the resulting Quality-of-Experience (QoE) has limited impact on the bitrate [58].

Transcoder banks are now common as intermediate devices sitting between mobile devices of various types, such as smartphones, tablets. They provide a way of mediating between high-quality source video, typically held at a server, and the processing capability of the target device, along with any bitrate restrictions on the network path to the device. Should these transcoders or other intermediate devices need to decrypt the video stream then the decryption key is exposed. There is also a key management overhead involved in supplying the key to any intermediate devices, not only to transcoders but devices that might insert logos or watermarks. Another application could be through video transcoding at a satellite and additionally video transcoding already takes place as part of the broadcast statistical multiplexing process.

Transcoding of the quality level is also common in digital TV broadcast systems for the purpose of statistical multiplexing TV programs onto the transmission channel. Larger values of the Quantization Parameter (QP) produce a more compressed version of the original video, so that increasing the QP through transcoding can additionally support a 'pay-per-quality' service, as originally described in [43]. Of course, the original compressed video must be encoded with a low QP or high quality, as it is impossible to increase the quality of a video through transcoding. However, for any such 'pay-per-quality' scheme, content protection through encryption is required. Because such a scheme requires decryption before and re-encryption after transcoding, the complexity and transcoding latency will be increased [16]. In this paper, we present a way of transcoding with selective content encryption which aims to reduce those overheads. One should also remark that in the video plus depth 3D video format [41], the depth information is stored as a conventional video, in addition to the normal 2D video. Therefore, the same method of crypto-transcoding could be applied to the depth video stream in the video plus depth format.

Raw video in YUV format is encoded (compressed) prior to encryption. Figure 1 shows a classical transcoding system, in which the video is fully decrypted before altering the QP and re-encrypting.

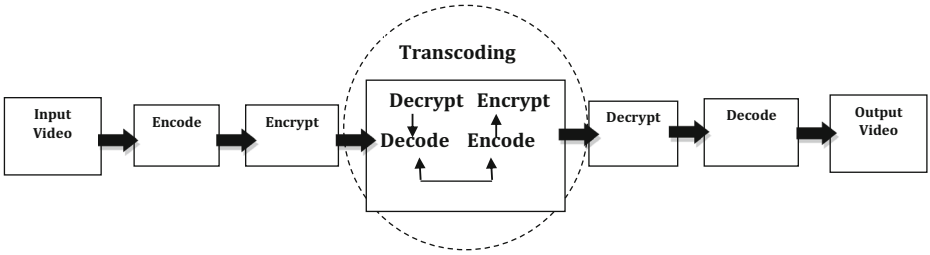


Fig. 1 Classical transcoding system

Though a number of transcoder designs [72] have sought to transcode without fully decoding the video, in order to decrease latency, some of these designs are vulnerable in one way or another to temporal error drift due to lack of synchronization between the original encoder and the remote decoder. In open-loop transcoder designs, because no reference is made to the frame store before re-quantization or transform coefficient pruning, temporal error drift builds up across a Group-of-Pictures (GoP), even though processing time is reduced. Closed-loop designs seek to approximate the cascaded architecture of Fig. 1 by reintroducing the transform stage; however, though drift is minimized, extra processing latency is re-introduced. Instead, the proposed crypto-transcoder, while avoiding temporal error drift, reduces the time previously taken-up by encryption and decryption. As a result, all processing takes place within the same transcoder block. Apart from reducing processing latency, decrypted video content is no longer exposed to other intermediate devices during a video stream's journey across the Internet. As the focus of this work is on encryption overhead rather than transcoder architectures, the use of alternative transcoder architectures will be the subject of future research. The need for the proposed scheme has increased. Since 2011, according to [20], the personalization phase of video delivery has occurred, by which cloud storage and cloud-based streaming of cached video [60] has become common after suitable representations have been generated by means of transcoder banks. Storage on local video servers in a way is even more vulnerable because professional security management may not be available, implying that encryption is also needed for non-cloud storage.

The key factor of the transcoding system represented in Fig. 1 is that there is a need to decrypt the video before transcoding is started and after transcoding to re-encode before performing re-encryption. As previously mentioned, in Fig. 1's classical system, there is a need to expose the video content during transcoding and the decryption keys are also exposed at the transcoder, as otherwise decoding will fail. In contrast to the classical system of Fig. 1, the crypto-transcoding scheme in this paper has the following research contributions to traditional secure transcoders:

1. The scheme presented in this paper employs a selective-encryption method that works only on uniformly-distributed selected syntax elements of the compressed video stream. Owing to this strategy, a selectively-encrypted video stream is decoder compliant, which means that, despite encryption, the video stream can be transcoded. As a result, there is no need to decrypt and then re-encrypt at the transcoder site.
2. The process of changing the QP otherwise works exactly as before. Finally, the decryption procedure is always performed by a target device and not at any intermediate point, where the content and/or decryption keys may be exposed.

3. SE is applied at the entropy-coding stage of the encoder as a crypto-compression scheme, so that there is limited additional computational overhead from the encryption process.
4. Choosing SE rather than full encryption runs a risk that content may not be sufficiently distorted and, so in some way, expose the video content any way without the need for decryption. Thus, an additional, significant contribution of this paper is to assess the effectiveness of the format compatible and compression-friendly approach to encryption for secure transcoding. The structural distortion analysis of crypto-transrated videos is done through the objective quality metrics in the experiments.
5. For experiments, we have conducted estimates of: the computation involved, the file sizes, and the effect, as part of the selective encryption process, of choosing various syntax elements of a Context Adaptive Binary Arithmetic Coding (CABAC) entropy coder [75]. A part of that evaluation process was the performance of the H.264/Advanced Video Coding (AVC) standard [75] and the more recently standardized High Efficiency Video Coding (HEVC) codec [45]. The intention was not to compare their relative compression performance, which has already been extensively explored. However, much legacy video content remains in H.264/AVC format or even in MPEG-2 codec format. However, it is possible to use format transcoding between MPEG-2 to H.264/AVC coding [31]. Therefore, an important part of this paper's contribution is the implementation of the proposed scheme over both more recent standardized codecs under transcoding.
6. Usually encryption in the context of transcoders is performed through a full-strength cipher, often the Advanced Encryption Standard (AES) cipher, before appropriate replacement in the compressed video stream. Despite its security strength, the AES has complex rounds, which take up much computation. In this paper, two ciphers are tested for crypto-transcoding through selected entropy-coder syntax elements. In experiments, an Exclusive OR (XOR) cipher is tested for both the H.264/AVC and HEVC encoder and then the AES cipher is also utilized in a stream cipher mode over the HEVC encoder. For speed, it is possible that the XOR cipher might be used for that purpose, if the security can be enhanced by means of One Time Pad (OTP) session keys. The current paper, therefore, also considers that alternative too.

The remainder of this paper is organized as follow. Section 2, describes the process of the SE used in this paper, the available transcoder architectures, and other background necessary for an understanding of the paper. Section 3 is a review of related work research in the domain of secure transcoding for HEVC. Then Section 4 outlines the evaluation methodology employed. Subsequently Section 5, considers the effectiveness of crypto-transcoding through experiments to determine the combined effect of SE and transrating according to the visual effect or distortion and the computational and bit-rate overheads, when using either of the two standard codecs. Results taken with two ciphers are also included. A comparative analysis of previous schemes and the current one is also included. Finally, Section 6 draws some conclusions concerning this research.

2 Background

This Section provides basic brief introductions to the main components of the scheme. It is not intended to be a comprehensive or full description of these components. In particular, the following Section describes the SE method used in this paper.

2.1 Selective encryption

Chaotic-map based full encryption does not have the performance penalties of the naïve encryption schemes described in Section 1. However, if encryption occurs before transcoding takes place, the video must first be decrypted prior to processing, which implies that the decryption key is also exposed at a CDN server. As in the future CDN servers may be placed remotely on a cloud, there is a risk from third-party contractors, who may operate in data-centers outside the legal jurisdiction of the content owner. For live video processing or interactive video processing, the need to decrypt and then re-encrypt will cause a significant delay if full encryption is employed. It is also no longer possible to perform intermediate video processing of video chunks if full encryption has been performed. For example, it is no longer possible to insert logos or watermarks without first decrypting the video stream.

The XOR is widely used in chaotic schemes by the researchers. An interesting study is [52], which considers the use of XOR encryption both in chaotic stream ciphers and two other schemes. Chaotic encryption is designed to avoid the computational overhead of full encryption with a block-based cipher but as the authors of [52] indicate, it has shortcomings if XOR encryption is employed. However, unlike the current paper, SE is not considered. In [57] there is a comparison between: 1) an SE scheme using AES encryption; 2) an SE scheme using AES to generate pseudo-random numbers prior to XOR encryption; and 3) an SE scheme based on chaotic generation of a random stream of numbers prior to XOR encryption. The latter is shown to improve considerably in terms of processing time compared to option 2). However, the originators of Scheme 3) do not consider the impact of intermediate transcoding.

Instead, Selective Encryption (SE) [38], as used herein, provides a lightweight procedure for video content confidentiality, as it does not encrypt all video data but selects the most influential or most important syntax elements from the multimedia content (herein video) and then encrypts those elements. Because of the reduction of encrypted material, SE reduce computational overhead compared to full (or its subset naïve) encryption. Thus, this approach lends itself to real-time or interactive applications of video streaming such as video phone, video conferencing, and telemedicine. However, not all types of SE provide efficient or sufficient content protection. Some SE algorithms exhibit weaknesses in terms of: an additional bit-rate overhead; lack of decoder compliance; and insufficient confidentiality. Nonetheless, these weaknesses can be addressed by ensuring that SE is performed at the final stage of a hybrid video encoder [24] i.e. entropy-coding stage, and after ensuring that statistical distribution of encrypted syntax elements will not be altered [4]. Only then does SE become beneficial as no or limited extra bit-rate overhead occurs. Notice also that in [3] the resilience of the SE scheme in this paper has already been checked and analysed in respect to a variety of attacks.

The SE utilized in this paper works at the Context Adaptive Binary Arithmetic Coding (CABAC) form (entropy coding) so that the SE scheme can apply both to the H.264/AVC codec and to HEVC. Notice that, though there are some differences in the way CABAC is performed in HEVC compared to H.264/AVC, H.264/AVC methods of SE can be adapted to those of HEVC by the conversion methods of [59]. The CABAC encoder works on a number of parameters which potentially could be used in the encryption operation, these being the Coded Block Flag; the Motion Vector Differences (MVDs); the Macroblock (MB) types; the Transform Coefficients (TCs); the delta quantization parameters (dQPs); and the numerical signs of TCs and MVDs. However, not all the syntax elements mentioned above provide decoder compliance and so in this paper we select: the signs of TCs and the signs of MVDs,

which we abbreviate to MV signs. Due to this selection, the proposed scheme allows SE to take place without decryption and re-encryption when transrating takes place. One key determinant of whether a syntax element is suitable for selection is whether it is by-pass coded, i.e. whether or not it affects the context adaptation models. Elements, such as the above two, are selected because they do not affect the context models. In addition, signs can reasonably be expected to follow a Uniform distribution, before and after encryption. Therefore, in a long-term statistical sense there is no bitrate overhead from this type of SE, even though for particular video streams it may turn out that there is some overhead.

An alternative to decoder-compatible SE exists, which might permit transcoding without prior decryption. That is video can be encrypted prior to compression. However, unless specialist forms of encryption are deployed, encryption removes the correlation that compression exploits, resulting in a loss of compression efficiency. Permutation-based forms of encryption can preserve or even enhance the correlation within a video frame. On the other hand, in [62], which described its proposed security as 'reasonable', the method was confined to spatial-only codecs. In fact, other encryption-then-compression schemes such as [80], though they offer a solution to a need for intermediate processing, including transcoding, appear confined to spatially-coded images and may have weak compression performance [18].

2.2 Ciphers: Advanced encryption standard (AES) and real-time XOR

The symmetric encryption cipher AES [54] was chosen as one encryption option for the SE elements. AES has low memory requirements and has been designed to guard against timing attacks. The AES structure is shown in Fig. 2. The encryption procedure uses a set of especially derivative keys called round keys. The following AES steps are for the encryption of a 128-bit block:

1. Calculate a set of round keys from the cipher key.
2. Arrange a state array with the plaintext block data.
3. Add the initial round key to the preliminary state array.
4. Execute nine rounds of state operations.

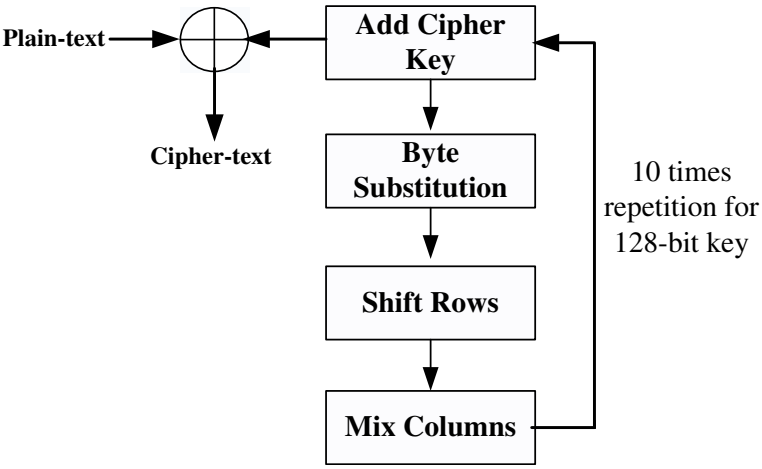


Fig. 2 AES structure to encrypt 128-bit plaintext block

5. Perform the tenth and last round of the state operation.
6. Duplicate the final state array as the outputted encrypted ciphertext.

As AES employs a single key with a limited key length, the efficiency is increased in terms of computational time, and memory consumption compared to asymmetric-key algorithms. Additionally, compared to prior standardized symmetric ciphers, AES provides a shield against many attacks [25].

The logical function XOR can optionally be applied as a symmetric cipher to the selected CABAC binary bins aggregated into 128-bit blocks. Its one-step operation results in rapid encryption. However, especially if the same key is repeatedly used, the security is weak, being vulnerable to a known-plaintext attack by XORing the plaintext with the ciphertext to output the key. The cipher is also vulnerable to flipping of the cipher text so that a valid but incorrect 'message' is generated. This effect is termed malleability, when decryption takes place. However, by means of a continually changing key generated with a Pseudo-Random Number Generator (PRNG), the confidentiality is greatly enhanced, if the initial seed can be securely distributed and the PRNG is 'sufficiently random'. Another possibility is to establish a long-term AES key through the Diffie-Helman key distribution protocol [63]. Thereafter, a session key, possibly for each video frame depending on the level of protection required, is then AES encrypted and included in the header of the XOR SE file. This form of lightweight encryption is also gaining prominence in lightweight encryption for smart grid applications, for example [37].

2.3 Transcoding

Video transcoding is the process of converting a compressed video from one form into another. Transcoding is performed on the basis of parameters such as the bitrate, frame rate, and spatial resolution. It is possible to convert from one codec standard to another, such as when a video encoded in a newer format, for example H.264/AVC, commonly employed for video over wireless, is converted to a legacy format such as MPEG-2, so that it can be broadcast over digital TV networks [42]. One of the main uses of transcoding continues to be reducing the bit-rate of a pre-compressed video stream according to the available channel bandwidth. Increasingly owing to the proliferation of networked devices, different clients may use different ways to access the Internet [76]. Each access network type has different channel characteristics such as bandwidths, bit rate errors and packet loss rates. At the user end, different networked devices including smartphones and desktop PCs are used for browsing the Internet. All end-user devices, including Set-Top Boxes, vary in terms of resources such as computing power and display resolution. To deliver video streaming data to users connected with different types of networks and having different terminals may need to be adapted dynamically at intermediate locations within a network [36]. Transcoding at intermediate relays has similar security implications to that when transporting encrypted source video in preparation for HAS representation generation within CDNs, described in Section 1. Encrypted video is also transcoded at various points during in-house production of films or TV programs, when the need for decryption and re-encryption is a considerable problem.

Transcoding is one way to handle and accomplish the conversion task, the other way being scalable video. However, there may be a bitrate overhead arising from scalable coding, though HEVC variants of SVC [11] have gone some way to address that issue by considering inter-

layer prediction. Encryption of H.264/SVC was considered in [4]. It is also possible [74] to apply a form of transcoding to encrypted scalable video after packetization. This works by employing the unencrypted packet headers to guide truncation of the encrypted packet payload. However, further consideration of transcoding of scalable video is beyond the scope of the current paper.2.4 Transcoding Architectures.

There are several types of transrating architectures [72] [6] [13]. The relative advantages/ disadvantages of these architectures are summarized in Table 1.The decoding-encoding cascaded architecture is the classical architecture, already described in Section1.

2.4 Standardized video codecs

As mentioned in Section 1, the scheme is suitable for both current video codec standards, the H.264/AVC standard [75] and the HEVC codec [45], the differences between which are summarized in Table 2, which is a modified version of that in [12]. Both of these codecs are standardized according to the bitstream format delivered to the decoder. As such they are suitable for use in consumer electronics devices. The HEVC standard is specialized towards HD and even UHD resolution video. As such it has introduced many coding refinements to achieve the required compression ratios to accommodate video streams in those formats across reduced bandwidth links. Though, the compression ratio can be increased by up to 50% by

Table 1 Comparison among different transcoding architectures

Parameters	Cascaded Decoding-Encoding	Open Loop	Closed Loop
Computational cost	This architecture is more costly in terms of computation overhead and processing units.	The open loop architecture is the fastest and the simplest means of video trans-coding.	This architecture approximates the cascaded decoding-encoding architecture by reintroducing a transform stage.
Error signal		In an open loop architecture, the output is neither measured nor fed-back for comparison with the input.	In a closed-loop architecture, an error signal is fed-back to minimize the temporal error drift.
Reference frames	A reference frame is used to minimize the difference between the input and output frame.	Reference frames from a frame buffer are not used in the processing.	For each reference frame, feedback (the difference between the actual frame and desired frame) is used to take corrective action.
Video drift	A reference frame is stored in a decoded frame buffer, which is utilized properly to remove temporal drift.	In this architecture temporal error drift is increased, particularly if high-frequency DCT coefficients are removed from the residual information.	Temporal drift is removed owing to the use of a reference frame and taking motion compensation as a linear function.
DCT/IDCT	Two pairs of DCT/IDCT are used.	The DCT/IDCT block is not used in this architecture.	A single pair of DCT/IDCT is used.
Best usage	This architecture can be used as a benchmark transcoder.	Its use straightforward and works best for intra-coded pictures. Latency is minimized.	This architecture is the best compromise for the video transcoding process, though additional latency occurs due to extra latency.

means of an HEVC codec that advantage comes with an increase in codec complexity, which in turn increases processing latency.

3 Related work

This Section considers the SE and possible transcoding of HEVC, as prior SE of H.264/AVC has been already considered in surveys such as [40] [35]. The Section additionally considers recent lightweight encryption schemes, as these have a bearing upon the investigation of XOR encryption of video in this paper.

The authors of [59] discussed how their prior H.264/AVC SE scheme, applied at the entropy coding stage to CABAC binstrings, could be adapted to HEVC, even though a somewhat different form of element coding (truncated Rice code instead of unary code) is employed in HEVC. In the HEVC version also, if real-time AES encryption was to be achieved, there was a need to concatenate elements to be encrypted so that their concatenated length was a power of two. In the prior H.264/AVC version as in the HEVC version, chosen elements of CABAC binstrings were encrypted with AES in CFB mode. Principally, bits of the

Table 2 Comparison between H. 264/AVC and HEVC standards after [12]

Category	H.264/AVC	H.265/HEVC
Names	MPEG 4 or H.264/AVC (Standardized in 2003)	MPEG-H, HEVC or H.265 (Accepted in Jan. 2013)
Key Improvement	• Designed to work with HD video stream delivery for online and Broadcast • 40–50% reduction of the bit rate compared to prior standards	• Aimed to deal with UHD, 4 k, 2 k for online and broadcast • 40–50% reduction of the bit rate at the same visual quality matched to previous standard (H.264/AVC)
Compression Model	Hybrid spatial-temporal prediction model • Flexible partition of Macro Block (MB), sub-Macro Block for MV prediction • 9 directional modes for intra prediction • 16 × 16 maximum Macroblock structure • Entropy coding by CABAC or lower complexity Context Adaptive Variable Length Coding (CAVLC) • ½ or ¼ pixel interpolation	Extended hybrid spatial-temporal prediction model • Introduced Coding Tree Units (CTUs) (Coding, Prediction and Transform Units (CU, PU and TU respectively) in quad-tree structure • 35 directional modes for intra prediction • Parallel processing architecture, enhancements in multi-view coding extension • CTU supporting larger block structure (64 × 64 pixels) with more variable sub-partition structures • Entropy coding is only Context Adaptive Binary Arithmetic Coding (CABAC)
Specification	• Support up to 4 K (4096 × 2304 pixels/frame) • Supports up to 59.94 fps • 21 profiles; 17 levels	• Support up to 8 K UHDTV (8192 × 4320 pixels/frame) • Supports up to 300 fps • 3 approved profiles, draft for additional 5; 13 levels
Drawback	Unrealistic for UHD content delivery owing to high bitrate requirements. Low frame rates unsuitable for higher resolutions.	Computationally expensive (~300% +) due to larger PUs and expensive Motion Estimation (intra prediction with more modes, asymmetric partitions in inter prediction, 1/8th pixel interpolation)

quantized transform coefficients (QTCs) and motion vector difference (MVDs) were encrypted. By careful choice of the encrypted CABAC parameters, it is possible to make the bitstream format compliant with limited or no impact on the bitstream size, as context modelling is not affected. As encryption does not extend over HEVC's entropy coding slices, potential parallel computing is also not affected. However, the method of [59] is reported by the authors to not be robust to compression domain processing, which includes some forms of transcoding. A minor issue that possibly could be rectified is that an update of the HEVC standard has meant that the method is no longer format compliant.

Hofbauer et al. [27] investigated transparent encryption of HEVC bitstream. Transparent encryption is a form of SE in which the viewer is able to partially view the content with a view to encouraging purchasing of a full quality version. The method worked by flipping the signs of AC transform coefficients signs. The percentage of bit flipping can be varied according to the desired level of transparency. Mid-range video quality, determined by QP, was evaluated. In fact, the authors conceded that the approach is unsuitable for high-quality video because, in this case, some blocks may not actually be transformed, resulting in no bits to flip.

To address issues with pioneering SE of HEVC CABAC elements, in [78], a somewhat different choice of coding elements was chosen, namely coeff_sign_flag, mvd_sign_flag, cu_qp_delta_abs and the suffix of abs_mvd_minus2. As before these bits are extracted, encrypted with AES after concatenation before placing the encrypted bits back in their original positions in the HEVC output bitstream. Though [78] demonstrates resilience against a replacement attacks, i.e. replacing bits known to be encrypted with other bits, and some other threats, within [78] the few pages available did not permit a full cryptanalysis. Preliminary analysis, did however suggest low computational overhead. Recent work [19] has also considered ways to protect the privacy information of individuals. One-way anonymity can be preserved is through a group signature mechanism and these mechanisms can be made flexible from a server's perspective. In [19], they are also made more flexible from the user's perspective.

The authors of [70] proposed an entropy-coding stage SE scheme that avoided affecting syntax elements which potentially might be manipulated by encryption. For example: splicing of video is suggests that Network Abstraction Layer (NAL) headers cannot be encrypted; motion information bits affect no-reference quality assessment [34]; compression domain insertion of watermarks [7] may be affected; and more specifically to the current paper, certain forms of compression domain transcoding [32] can be impacted. In fact, the possibility of combining more than one compression domain process, such as transcoding, watermarking, and encryption needs to be considered [7] [8]. In [70], no choice of syntax elements for SE is made but the trade-off between increase in bitrate for the same video quality and increase in confidentiality or security in general is analysed. For example, the impact of encrypting Sample Adaptive Offset (SAO) filtering parameters has very limited effect on the bitrate but can be thwarted if this form of in-loop filtering is turned off at the encoder.

In [9], the implementation of a symmetric transcoder which incorporates SE and is suitable for smartphones. By symmetric is meant that the transcoder will encrypt video output from an encoder or decrypt video arriving in encrypted form, similar to a cascaded transcoder. By selecting syntax elements that bypass context modelling at the entropy stage, as also used by prior SE schemes, the scheme avoids an increase in the bitrate even though SE has been applied. Because confidentiality may be weakened by only choosing bypass CABAC syntax elements in [10], the authors analyse the trade-offs if non-bypass (regular mode) elements are chosen. By choosing to encrypt the intra prediction modes used, the confidentiality is

significantly improved. As in [9] ciphering is embedded in the transcoder rather than the encoder to avoid the need to make SE encoder dependent. In a similar way, the decoder is made independent of decryption, which is performed in the transcoder. However, this appears to leave the video exposed if the symmetric transcoder is place in an intermediate network middlebox. Recently, [56] provides another analysis of CABAC syntax elements suitable for encryption, choosing the coeff_abs_level_remaining element. However, it is unclear what the impact on decoder format compatibility or bitrate overhead is or whether other elements could also be selected.

HEVC tiles allow a video sequence to be decomposed into autonomous rectangular areas. In the SE option of [22], the authors examined Region-of-Interest (RoI) encryption of tiles. It was found that this implied that motion vectors from non-encrypted tiles should not reference encrypted tiles. The result was some loss of rate-distortion performance to achieve RoI SE. However, propagation of encryption outside encrypted tiles was avoided.

Thomas al. [68] considered various secure transcoder systems according to the aims of the system. For example, if the transcoder only works on inter-coded frames (P- and B-frames) in the interests of accelerating transcoding, If full encryption of I-frames occurs then the Cascaded Pixel Domain Transcoder (CPDT) form of transcoding is handicapped by the lack of I-frame data, which the authors address by modifying the CPDT transcoder. SE, rather than full encryption, may also be performed only on intra-coded frames. However if traditional sign-bit encryption is performed then error drift through lack of synchronization between encoder and decoder can occur. Again the authors of [68] suggest a modified sign-bit encryption scheme to restore synchronization.

In [66], new ways of transcoding in the face of HEVC's quad-tree block structures are considered. Basically the proposed solution focuses on reducing the coding options employed. An implemented transcoder resulted in 80% less time for the transcoding process as compared to a conventional cascaded encoder decoder, see Fig. 1, but the coding performance reduced by up to 5%. In [50], computational scalable video transrating was investigated because of the high computation cost of HEVC cascaded transcoding. In this technique, information from the input bitstream prior to transrating was used to reduce the computation after bitrate scaling. This was done by altering the CU and PU structures so that the number of RD evaluations was reduced. Two methods of reducing subsequent computation were considered, namely top to bottom (T2B), bottom to top (B2T) processing of the CU structure. Machine learning techniques aided in this process. Furthermore, PUs were also reduced in number by manipulating information from the input stream.

Because with the advent of HEVC, H.264/AVC standard compressed content may need to be converted to the new standard and in [21, 49] fast transcoders have emerged. However, because this paper is concerned with transrating rather than format transcoding, these transcoder designs are not considered further herein. Because also there is concern over the computation overhead from using HEVC standardized codecs, there have been numerous proposals as to how to reduce that overhead, such as [2, 15, 33, 46, 61, 77]. However, these are outside the scope of the present work.

In the view of the authors of [47], conventional encryption methods, such as full encryption by AES, are unsuitable for video data because of the computational overhead, even though major encryption providers such as Verimatrix & Widevine use AES encryption for satellite systems. The approach of [47] was to design a wavelet-based hardware codec that is made amenable to encryption, though in commercial usage, the lack of standardization. In [79], a hardware-based, cellular automata method of encryption was proposed. Because the

application was video surveillance, RoIs can be selected, reducing the computational overhead further. However, possible loss of compression efficiency caused by the need to treat encrypted RoIs separately and the risk from applying encryption to a limited area does not seem to have been checked. Nonetheless, the method is suitable for lightweight encryption of video collected in an Internet of Things (IoT) video surveillance setting and presented an alternative, lightweight method to the SE of this paper. In the context of IoT-based video surveillance, in [44] only key frames were encrypted with a chaotic encryption scheme. However, the weakness of some tests used to justify the security of such chaotic encryption schemes was questioned in [52], as was the relative encryption speed compared to conventional block-based encryption methods. Instead, the authors of [71], for a secure but lightweight IoT-based video streaming scheme, preferred to employ conventional block-based encryption. However, they reduced the message overhead through a lightweight transport-layer protocol into which they inserted a Hash-based Message Authentication Code (HMAC) authentication field. All the same, SE, as presented in the current paper, remains an effective lightweight encryption method, with advantages of lightweight chaos-based encryption. As developed in the current paper, it also supports intermediate processing, such as the transrating analysed herein.

4 Methodology

To remove error drift from the video stream, transcoding was by means of the closed-loop architecture of Table 1. Fig. 3 is a block diagram of the closed-loop architecture, i.e. presented for a generic transrater. We have proposed and implemented the crypto-transcoder in a closed-loop architecture. Therefore, it is worthwhile to explain this architecture here for newcomers to the field. The input bitstream, R_{in}, is first variable-length decoded (vld), allowing the embedded MB motion vectors (MVs) to be extracted. Inverse quantization, q^{-1}, of the residual transform coefficients then takes place using a QP, q_1, originally used by the encoder and placed in the compressed bitstream. A control, *ctrl*, places a desired new QP, q_2, ready for re-quantization before placing the residual coefficients within the output bitstream, R_{out}. The output bitstream is produced by variable length coding, *vlc*, that is using some form of entropy coding such as CABAC or CAVLC. However, prior to output of the bitsream, the process of motion compensation (*MC*), takes place based on the stored reference frame(s), which are

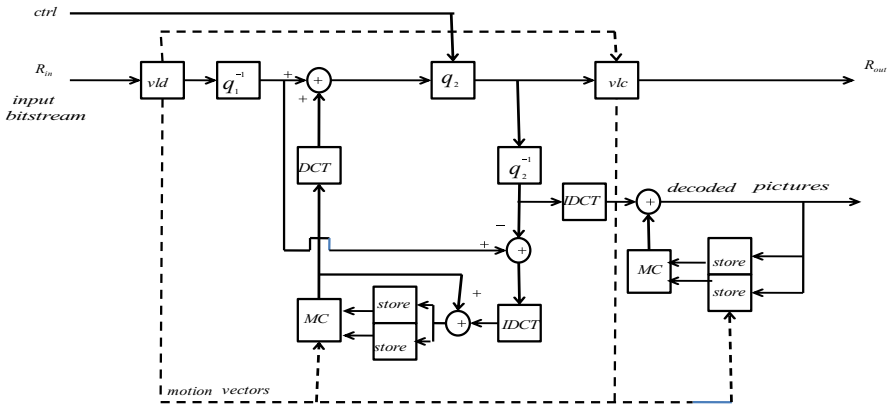


Fig. 3 Closed loop architecture of transcoders

adjusted for the changed QP. To do this requires spatial-frequency transforming motion-compensated MBs to allow partial synchronization at the decoder for the effect of transcoding at a different QP to the one originally used at the encoder. Thus the motion-compensation loop of Fig. 3, is the closed loop that gives this type of transcoder its name. Also shown in Fig. 3 is the path taken at an end decoder to retrieve the video. Because the computation involved in a closed-loop architecture transcoder is already substantial, it is unwise to increase the computation further by requiring decryption and re-encryption. From Fig. 3, it is apparent, that if multiple representations at different bitrates are to be generated, (say) for HAS, then motion information and de-quantized information from the decoder can be extracted once and those inputs to the encoder can be repeatedly used in the encoder, thus saving on computation. It should also be noticed that if the source bitstream was not already encrypted, e.g. not part of an encrypted bitstream delivered to a middlebox or cloud data-center, then the output could be selectively encrypted at the Variable Length Coding (VLC) stage of Fig. 3. In Fig. 3 DCT is used for Discrete Cosine Transform and IDCT is its inverse.

During motion compensation, typically, each 16 × 16-pixel MB is split up into 4 × 4 pixel blocks in H.264/AVC, while into 8 × 8 pixel blocks in HEVC. (In some cases in HEVC, an 8 × 8 pixel block is further split into 4 × 8 and 8 × 4 pixel block sizes as well.) After the decoding of each frame (and before moving on to the next frame) the Motion Vector (MV) parameters, being the vector components (MV_x, MV_y) and a pointer(s) to the reference frame (s) used, are extracted. These parameters define the motion compensation for each smallest possible block. To encode and encrypt the bitstream at a number of different QP levels, the stepwise procedure of the proposed scheme is described below and in Fig. 4, for the example case of crypto-transcoding to a set of QPs. The adopted algorithm shown in Fig. 4 assumes that the video is initially encoded at QP = 12, and then transcoded to a set of lower quality video streams, with QP = 24, 36, and 48, given that the range of QPs for H.264/AVC and HEVC is 0–51. In Fig. 4 ME represents motion estimation.

Step 1: Encode the raw video with the proposed H.264/HEVC crypto-entropy coder (modified CABAC) along with quantization at the initial QP value, herein QP = 12. Encrypt the selected parameters, i.e. signs of MVD and signs of residual texture information (TCs) of the outgoing H.264/HEVC bitstream.

Step 2: In the H.264/HEVC decoder, for every compressed H.264/HEVC encoded bitstream, the horizontal MVx and vertical MVy, along with a pointer(s) to the reference frame(s) used, are extracted from the smallest possible block of the H.264/HEVC bitstream and stored separately in a file. The decoding process is performed without decryption of any parameters and, thus, without the need to supply a key.

Step 3: Perform crypto-transcoding with the new QP value but without decryption of the video bitstream, according to the closed-loop architecture of Fig. 3. If necessary, repeat transcoding with one or more different QPs with the same extracted MV parameters from step 2.

Step 4: At the receiving end, the received video sequences after decryption and decoding will be able to be watched. Decoding with decryption follows the lower path shown in Fig. 4. (The optional verification step is added for clarity of understanding the whole scheme.)

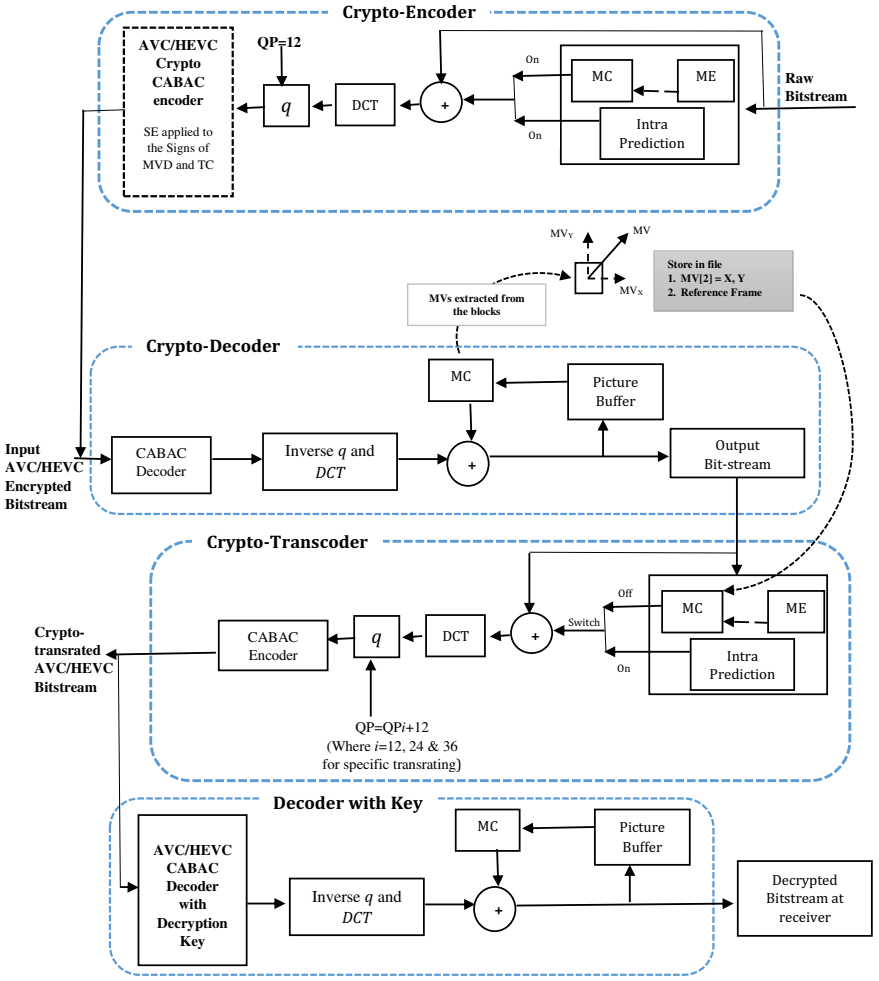


Fig. 4 Stepwise flow diagram of the proposed crypto-transcoding scheme

5 Evaluation

To perform experiments, H.264/AVC reference software JM 18.6 [30] and HEVC reference software HM-15.0 [29] were selected. The machine used for all the experiments was an Intel Core i3 Core 2 Duo (2.10 GHz) processor, with 6 GB RAM and Microsoft Windows 8.1 Professional installed as the operating system. Transcoding on this machine was conducted with the closed-loop architecture of presented in Fig. 4. As described in Section 1, video content for HAS can be transcoded into a number of qualities, which in turn determine the bitrate of the different representations available to a client by selecting from a manifest file. The quantization parameter (QP) normally determines the extent of compression (necessary to make bandwidth consumption manageable), which also impacts on the processing required at the target device. This assumes that Variable Bit Rate (VBR) video content is presented to the transcoder, with the result that constant quality representations result. Constant Bit Rate (CBR)

video, although it allows video storage to be planned has a disadvantage during transmission because if the content, for a given quality, does not require its bitrate then bandwidth wastage occurs. The well-known FFmpeg software, depending on the underlying encoder selected, allows constrained VBR to be output [53], avoiding a risk of a video stream temporally exceeding its average bitrate. However, in this paper for simplicity in making quality comparisons, VBR is assumed.

Two reference CIF (352 × 288 pixels/frame) video sequences, i.e. *Stefan* and *Mobile* (available from http://www2.tkn.tu-berlin.de/research/evalvid/cif.html) and an HD 720 (1280 × 720 pixels/frame) video sequence, i.e. *Four People*, were transcoded to different QPs (12, 24, 36 and 48). All tested videos are encoded with I, P and B frames with GOP of 16. A QP of 12 results in broadcast quality video, while if the QP is set to 48 the video is very compressed but equally the visual quality is very low in both an H.264/AVC and HEVC codec. Otherwise, the video configuration settings of the original test videos were retained.

Fig. 5 illustrates frames from the test video sequences with calculated average Peak Signal to Noise Ratio (PSNR), an objective measure of video quality measured in decibels (dBs) [28], and average Structural Similarity (SSIM) index [73], which aims to capture the human perceptual response on a scale (usually) of 0 to 1.

The above video sequences were firstly compressed by using lossless compression, by setting QP = 0, using either H.264/AVC or HEVC codecs. SE was applied during compression by selecting the TC and MV signs for AES encryption. Subsequently, the compressed versions of the sequences were transrated to different QP levels.

The following equations were used to calculate the various quantities reported. For the bitrate reduction in going from H.264/AVC to HEVC:

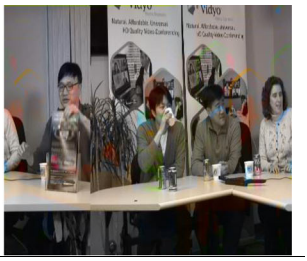
$$\Delta\, Bitrate = \frac{(Bitrate)_{HEVC} - (Bitrate)_{AVC}}{(Bitrate)_{AVC}} \times 100 \tag{1}$$

For visual evaluation after SE, the PSNR and SSIM index value were returned by eqs. (2) and (4) respectively:

$$\mathrm{PSNR} = 10.\log_{10} \frac{(2^{x}-1)^{2}}{MSE} \tag{2}$$

Stefan Video (frame # 13)
PSNR [Y=27.9, U= 46.8, V=41.9] dB,
SSIM = 0.9626

Mobile Video (frame # 54)
PSNR [Y=27.7, U= 40.5, V=37.5] dB,
SSIM = 0.9446



Four People (HD 720 video) (frame # 53)
PSNR [Y= 27.7 , U= 46.6 ,V= 54.21] dB,
SSIM = 0.9446

Fig. 5 Videos frames from the test videos (without crypto-transrating) showing PSNR and SSIM values for the luminance, Y, and chrominance, U and V, components

where MSE is the Mean Square Error between the reference video and the video of interest, with x being the bits per pixel, herein being eight to allow comparison between the two codecs.

The PSNR difference was calculated by eq. (3):

$$\Delta\, PSNR\,(Y) = \frac{(PSNR_y)_{HEVC} - (PSNRy)_{AVC}}{(PSNy)_{AVC}} \times 100 \tag{3}$$

and SSIM was calculated by eq. (4):

$$SSIM(a,b) = \frac{(2\mu_a\mu_b + c1)(2\sigma_{ab} + c2)}{(\mu_a^2\mu_b^2 + c1)(\sigma_a^2 + \sigma_b^2 + c2)} \tag{4}$$

where a, b are the two video frames being compared, with μ_a, $\mu_b$being average pixel value (intensity) within a, b respectively, with σ_a, σ_b being the variance within a,b respectively, and σ_{ab} being the covariance. Two variables, c1 and c2, are introduced to stabilize division with relatively small denominators [73].

Encoding time difference was found by eq. (5):

$$\Delta\, Enc.Time = \frac{(Enc.\ \ Time)_{HEVC} - (Enc.\ \ Time)_{AVC}}{(Enc.\ \ Time)_{AVC}} \times 100 \tag{5}$$

Apart from the comparison of video frame distortion by one of the two video quality metrics, PSNR and SSIM, two other processing steps were taken to determine the extent (or lack of distortion). The first of these was by edge-detection using a Laplacian filter [39] and the second of these was by pixelation [69], i.e. lowering of the resolution, of the frames. For both of these effects, an 8 × 8 pixel filter was applied to a video frame to obtain the effects.

5.1 Crypto-transcoding with AVC and HEVC

The results of crypto-encoding and then crypto-transrating according to the choice of codec and QP are shown and discussed in this Section. The results are taken according to the steps of the algorithm presented in the Methodology section (Section 4).

From Figs. 6 and 7, the reader will see that the content of all sample video frames is distorted to a greater or lesser extent after applying SE (Figs. 6, 7 (a1, a2)) and then transrated shown in Figs. 6 and 7 (b1, b2, c1, c2, d1 and d2). However, there are conspicuous visual differences between the outputs of the both codecs. In addition, some portions of the frames remain still visible, such as the static calendar in *Mobile* and the tennis court in *Stefan*. This is not surprising as the calendar and tennis court are largely static, whereas the method of SE works on changing values of the MV signs. If there is little motion the MVD will be small and the impact of encrypting the signs will be small. Examining the best quality at QP = 12 and the worst at QP = 48 of the sample *Mobile* frame, it is evident that the impact of SE along with transrating is greater on lower quality video. The same is also particularly apparent for the same QP levels of *Stefan*, especially when the original encoding was with the HEVC codec. This effect may be related to the TC sign encryption. At higher QPs, more of the TCs are reduced to zero and, hence, do not appear in the input to the CABAC engine. Therefore, the effect of flipping the signs of the fewer TCs that are present may have a greater impact. To verify the effectiveness of the crypto-encoding along with transrating over tested videos, video

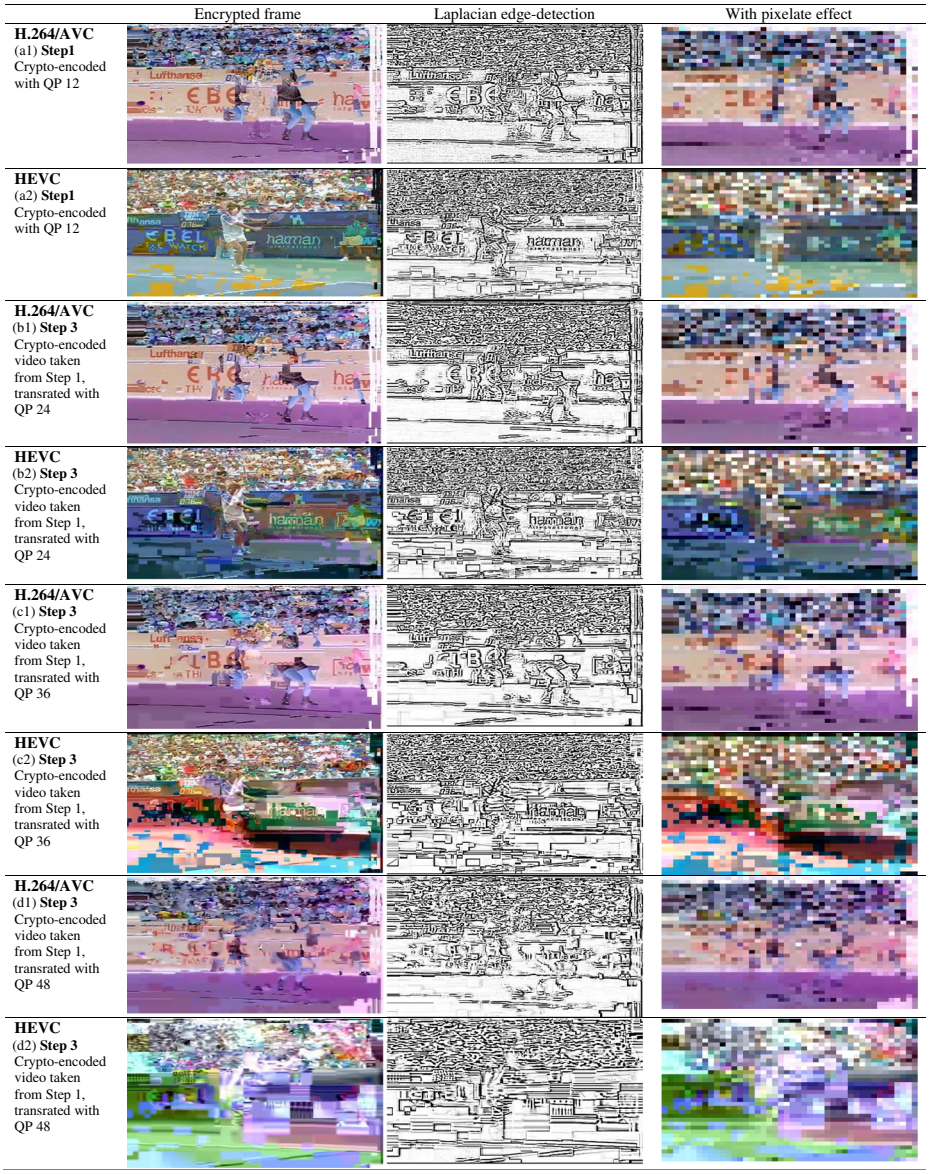


Fig. 6 Visual results for *Stefan* (frame # 13) after crypt-transrating with different QPs, with edge-detection and pixelation tests

structural distortion analysis is done with Laplacian edge detection or with the pixelate effect, as is demonstrated in Figs. 6 and 7. Notice that a recent way of testing the effectiveness of encryption is to apply edge detection to the distorted frame [26] and pixellation also serves to check the effectiveness of the encryption.

In Tables 3 and 4, it is proved from the output file sizes, HEVC encryption and subsequent transcoding generally results in greater compression at all QP levels. However, this comes at a cost in a considerable time spent in transrating of the encrypted content when HEVC is

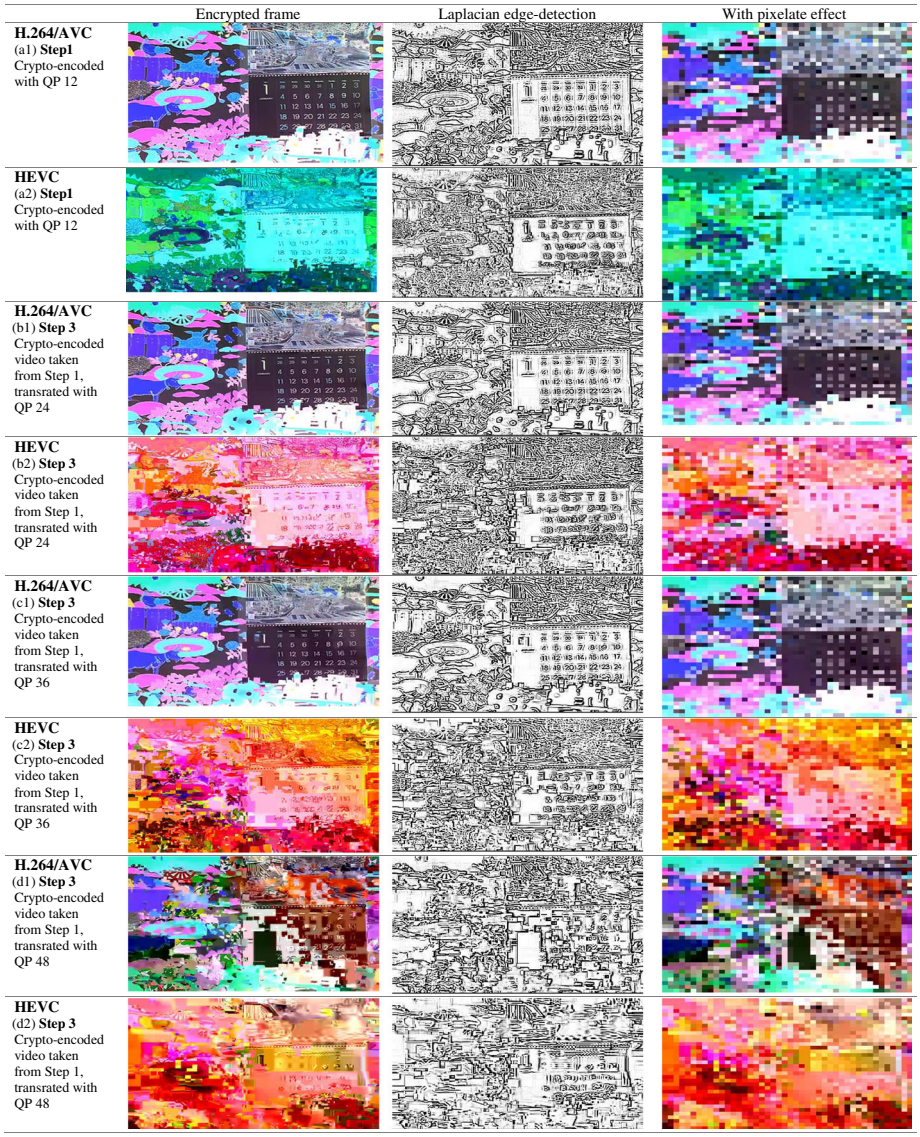


Fig. 7 Visual results for *Mobile* (frame # 54) after crypto-transrating with different QPs, with edge-detection and pixelation tests

Table 3 Crypto-Transrating results for the HEVC and H.264/AVC video codec with the *Stefan* video sequence

QP	Bit Rate (AVC)	Bit Rate (HEVC)	Δ Bit Rate (Avg. bit rate)	PSNR_Y(dB) AVC	PSNR_Y(dB) HEVC	Δ PSNR (Y) (dB)	Δ Enc. Time (ms)
12	1670 kb	1130 kb	- 32.34%	10.01	16.19	+61.74%	+81.79%
24	564 kb	375 kb	−33.51%	10.65	14.03	+31.74%	+33.09%
36	204 kb	114 kb	−44.12%	10.48	12.83	+22.42%	+14.93%
48	159 kb	29 kb	−81.76%	11.02	12.58	+14.16%	+9.18%

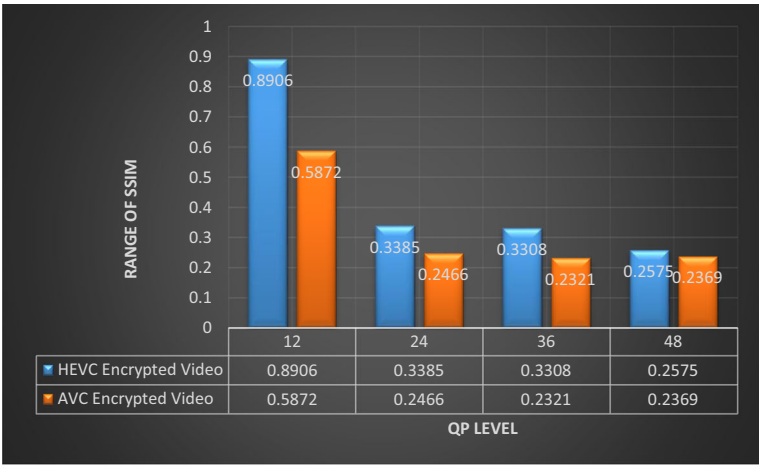
Table 4 Crypto-Transrating results for the HEVC and H.264/AVC video codec with the *Mobile* video sequence

QP	Bit Rate (AVC)	Bit Rate (HEVC)	Δ Bit Rate (Avg. Bit-Rate)	PSNR_Y(dB) AVC	PSNR_Y(dB) HEVC	Δ PSNR (Y) (dB)	Δ Enc. Time (ms)
12	1755 kb	1213 kb	−30.88%	6.7	16.5	+146.27%	+113%
24	691 kb	429 kb	−37.92%	6.7	16.7	+149.25%	+82.63%
36	252 kb	113 kb	−55.16%	6.7	12.9	+92.54%	+61.20%
48	178 kb	55 kb	−80.29%	7.1	9.1	+28.17%	+34.29%

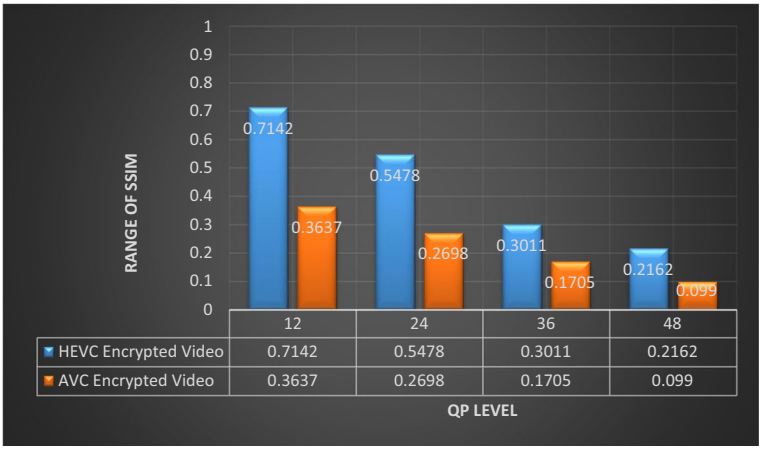
employed. From the two Tables, in PSNR terms, HEVC SE and subsequent transrating results in less distortion than when H.264/AVC is used. Recall that Tables 3 and 4's PSNR results are averaged across each of the two sequences compared and, thus, subjective impressions from Figs. 6 and 7 may not be confirmed for individual sample frames. Distortion increases for HEVC in going from low QP (high quality) to high QP (low quality), though this change is the combined effect of lower quality video at the higher QPs and the addition of SE. A surprising result, perhaps, is that the gain from HEVC rather than H.264/AVC transrating of encrypted content in terms of reduced bitrate is relatively larger at lower qualities. This is perhaps surprising because HEVC was designed for high resolution and high quality video, which is necessary as visual artifacts are otherwise more apparent at higher resolutions. Comparing between *Stefan* and *Mobile* video sequences, a similar pattern of results is evident. However, there are greater differences in quality between the codecs for *Mobile*, with greater spatial complexity than *Stefan*. It is more apparent from the SSIM index results in Fig. 8 that at QP level 12, SE by the given method is unsuitable when an HEVC codec is used because there is insufficient distortion across the sequences. For spatially more complex *Mobile* as well, if SE is followed by transrating, HEVC processing results in reduced distortion. Therefore, there is a content-dependent and quality-dependent effect when moving between H.264/AVC to HEVC when this type of processing takes place.

5.2 HEVC encoded crypto-transrating with XOR cipher and AES

The performance of crypto-transcoding with an XOR cipher is tested with H.264/HEVC encoders in Section 5.1 and presented in Figs. 6 and 7. In this Section, transrating after encryption with the state-of-the-art AES cipher is also tested. Fig. 9 shows the visual results of crypto-transrating with the two implemented ciphers i.e. XOR and AES, operating over the same syntax elements and encoded with the HEVC encoder on the HD720 *Four People* video. AES has many modes of operation. In the Cipher Feedback mode (CFB), AES works as a stream-cipher [3] and additionally benefits from a self-synchronization mechanism for real-time transmission. The visual results (from frame no. 53) indicate that the encryption performed using AES-CFB produces a strongly distorted frame, whereas applying an XOR operation results in insufficient distortion. Histogram analysis applied to crypto-transrated video with the two said ciphers is presented in Fig. 10. Figure 10 contains histograms of the red, green, blue and luminance values of each pixel of the tested video sequence. This Fig. 10 (c, e) indicates that the AES-CFB cipher results in an increase in entropy or randomness across the pixel values relative to using the XOR cipher. This entropy is evident in the spreading of more black and sharp colours across the video frames compared to the original histogram values prior to crypto-transrating of the *Four People* video, Fig.10 (a). This finding implies



(a)



(b)

Fig. 8 Average SSIM values for H.264/AVC and HEVC for a) *Stefan* and b) *Mobile* crypto-transrated videos

that, if the videos are selectively encrypted by AES, it is difficult to infer the presence of an object in any one of the R, G, B and luminance domains.

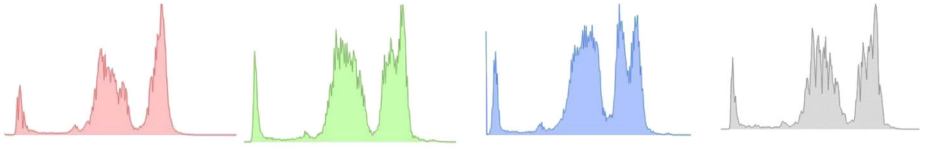
5.3 Comparative analysis

Table 5 is an analysis comparing the scheme described with previous secure transcoders. The crypto-transcoder has advantages over previously proposed schemes in [9, 16, 68], though to some extent these are more to do with the development of the HEVC codec than merits of prior schemes. The proposed crypto-transcoder scheme provides a greater compression rate due to the possible use of an HEVC codec, as well as preserving the whole video structure at transcoders. The proposed scheme also implements real-time transcoding as the authors did in [9, 16]. Meanwhile, when H.264/AVC is targeted, the computational overhead in terms of encryption with transcoding time appears to be lower than the schemes proposed in [9, 16], and [68].

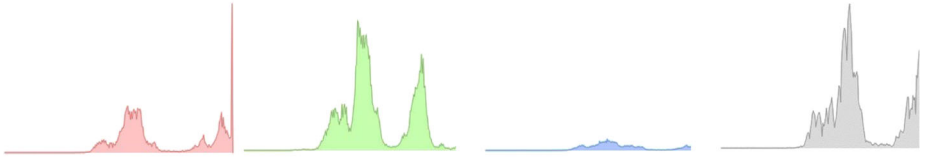
Fig. 9 Visual results for *Four People* (frame #53) after crypto-transrating through XOR and AES ciphers

6 Conclusions

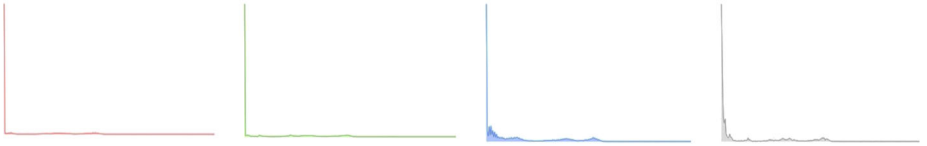
This paper implemented a joint crypto-transcoder with two widely deployed video codecs, H.264/AVC and HEVC. The main contribution has been to reduce the processing latency at intermediate transcoders arising from a need to encrypt video for reasons of content protection and possibly privacy protection at the user. The paper proposes that a suitable selective-encryption scheme is applied in that the encrypted video bitstream remains decoder format compatible. Because the selective encryption on carefully selected bin-strings reduces the computational overhead of encryption, there is a further gain, apart from the desired reduction



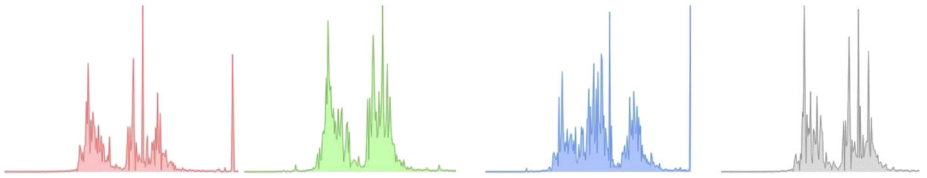
(a) RGB histograms & luminance histogram of original frame



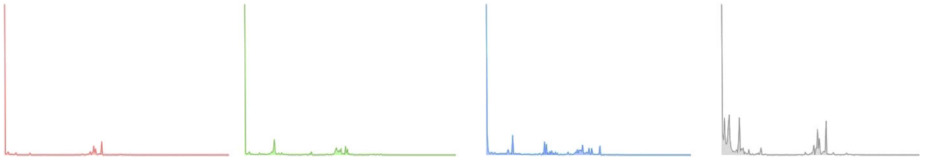
(b) RGB histograms & luminance histograms of video encrypted with XOR at QP 12



(c) RGB histograms & luminance histogram of video encrypted with AES (CFB) at QP 12



(d) RGB histograms & luminance histograms of video encrypted with XOR and then transrated with QP 48



(e) RGB histograms & luminance histograms of video encrypted with AES (CFB) and then transrated with QP 48

Fig. 10 Histogram analysis of *Four People* video after crypto-transrating with XOR and AES ciphers and encoding with an HEVC encoder

in latency, in terms of an overall bitrate reduction. The experiments performed have shown that transcoding of HEVC video produces better results in terms of reduced file sizes. However, if one considers the computation cost of performing joint crypto-transcoding, then HEVC results in greater transcoding times. Therefore, if transcoding is likely to be used then it is better to use SE rather than full encryption, unless the application is (say) military, legal, or medical. From experiments, the gains from using SE are substantial in terms of limited bitrate overhead and (according to the machine employed) several seconds saved in encoding alone, even for short video sequences. For longer video streams, including broadcast TV and films, the savings in bit-rate and computation time will be considerable. If reduced bandwidth is not a priority then the H.264/AVC codec remains viable, unless a hardware HEVC codec is available. From comparisons of visual distortion in the paper, it is apparent that there is a content dependency and QP-dependency when transrating to lower bitrates, in the sense that SE may contribute reduced distortion at higher bitrates. In going from H.264/AVC to HEVC, it is also likely that

Table 5 Comparison with some previous secure transcoding schemes

Parameters for comparison	(Díaz-Sánchez et al. 2016) [16]	(Boyadjis et al. 2014) [9]	(Thomas et al. 2010) [68]	Proposed crypto-transcoder scheme
Video structure preservation at decoder	Yes: Full encryption produced the actual video structure at decoder.	Yes: Preserved the video format with no bit-rate over-head	Yes: Preserved the cipher synchronization with no loss of compression performance at decoder	Yes: The scheme provides a greater compression rate, as well as preserving the video structure at the decoder
Real-time transcoding	Yes, by sharing load on multiple machines	Yes	No	Yes
Codec standard	H.264/ SVC	H.264/AVC and HEVC	H.264/AVC	Hybrid model: works both for H.264/AVC and HEVC
Encryption type	Full encryption: It proposed a distributed encryption and flexible key management, which facilitates content filtering, key extraction and content decryption at the receiver	Selective Encryption was based on symmetric ciphering and managed by AES	Introduced two methods of SE: **1.** On the basis of full I frame encryption; and **2.** On the basis of sign bits of motion vectors and transform coefficients	SE based on the arithmetic signs of motion vectors difference and the signs of texture data (TC)
Computational overhead	High due to full encryption.	Low	High in Scheme 1. Low in Scheme 2 but produced error drift after transcoding	Low for H.264/AVC, medium for HEVC due to video encoding time

there will be less distortion at the same QP. In fact, it may be better to avoid HEVC joint crypto-transcoding at the lowest QPs, i.e. for broadcast-quality video. The paper also considers whether a low-complexity cypher is worth considering because of a further reduction in latency at the server and client ends. However, from cipher comparisons between AES and XOR encryption of the selected elements in the video stream, XOR appears insufficient, despite interest in lightweight encryption using XOR for smart grid applications.

Future work will consider how to select suitable syntax elements according to the type of content and the expected transcoded quality. The aim will be to distort those features of a video frame that some encrypted syntax elements do not presently have an impact upon. In that sense, an intelligent SE scheme will adaptively apply encryption to elements within the compressed video stream. At the same time, any such system should preserve decoder format compatibility and minimize any increase in bitrate. There is also scope for performing a number of other statistical tests to confirm the results, such as through correlation-coefficient analysis as an alternative to histogram analysis, and finding the video encryption quantity as an alternative way to investigate the QP-dependency of encryption. Rate-distortion analysis, such as through the well-known Bjøntegaard-Delta metric, is an alternative way of examining the bitrate overhead from encryption.

References

1. Adeyemi-Ejeye AO, Alreshoodi M, Al-Jobouri L, Fleury M, Woods J (2017) Packet loss visibility across SD, HD, 3D, and UHD video streams. J Visual Commun Image Rep 45:95–106
2. Ahn S, Lee B, Kim M (2015) A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding. IEEE Trans Circ Syst Video Technol 25(3):422–435
3. Asghar MN, Ghanbari M, Fleury M, Reed MJ (2014) Confidentiality of a selectively encrypted H.264 coded video bit-stream. J Visual Commun Image Rep 25(2):487–498
4. Asghar MN, Ghanbari M, Fleury M, Reed MJ (2015) Sufficient encryption based on entropy coding syntax elements of H.264/SVC. Multimed Tools Appl 74:10215–10241. https://doi.org/10.1007/s11042-014-2160-6
5. Bing B (2016) Video over wireless. McGraw-Hill Educ, New York, NY
6. Bjork N, Christopoulos C (1998) Transcoder architectures for video coding. IEEE Trans Consumer Electron 44(1):88–98
7. Boho A et al (2013) End-to-end security for video distribution: the combination of encryption, watermarking, and video adaptation. IEEE Signal Process Mag 30(2):97–107
8. Boho A, Van Wallendael G, Dooms A, De Cock J, Schelkens P, Prenel B, Van de Wall R (2013) End-to-end security for video distribution. IEEE Sig Process Mag 30(2):97–107
9. Boyadjis B, Perrin ME, Bergeron C, Lecomte S (2014) A real time ciphering transcoder for H.264 and HEVC streams. Proc IEEE Int Conf Image Process: 3432–3434
10. Boyadjis B, Bergeron C, Pesquet B, Dufaux F (2017) Extended selective encryption of H.264/AVC (CABAC) and HEVC encoded video streams. IEEE Trans Circ Syst Video Technol 27(4):892–906
11. Boyce JM, Ye Y, Chen J, Ramasubramonian AK (2016) Overview of SHVC: scalable extensions of the high efficiency video coding standard. IEEE Trans Circ Syst Video Technol 26(1):20–34
12. Brorsoft Studio (2017) Video Convertor documentation. http://www.brorsoft.com/video-format/h265-vs-h264.html
13. Chang SF, Vetro A (2005) Video adaptation: concepts, technologies, and open issues. Proc IEEE 93(1):148–158
14. Choon LS, Samsudin A, Budiarto R (2004) Lightweight and cost-effective MPEG video encryption. Proc Info Commun Technol: 525–526.
15. Correa G, Assunção P, Volcan Agostini L, da Silva Cruz L (2015) Fast HEVC encoding decisions using data mining. IEEE Trans Circ Syst Video Technol 25(4):660–673
16. Díaz-Sánchez D, Sánchez-Guerrero R, Aries P, Almenarez F, Marín A (2016) A distributed transcoding and content protection system: enabling pay per quality using the cloud. Telecomm Syst 61:59–76
17. Dorwin D, Smith J, Watson M, Bateman A (2017) Encrypted media extensions W3C recommendation. W3C
18. Eletftheriadis A, Batra P (2006) Dynamic rate shaping of compressed video data. IEEE Trans Multimed 8(2):297–314
19. Eom S, Huh J-H (2018) Group signature with restrictive linkability: Minimizing privacy exposure in ubiquitous environment. J Ambient Intell Humanized Comput: 1–11
20. Fan Q, Yin H, Min G, Yang P, Luo Y, Lyu Y, Huang H, Jiao L (2018) Video delivery networks: challenges, solutions and future directions. Comput Elec Eng 66:332–341
21. Fang J, Chen Z, Liao T, Chang P (2014) A fast PU mode decision algorithm for H.264/AVC to HEVC transcoding. Proc 4th Int Conf Computer Sci Info Technol: 215–225
22. Farajallah M, Hamidouche W, Déforges O, Assad SE (2015) ROI encryption for the HEVC coded video contents. Proc IEEE Int Conf Image Process: 3096–3100
23. M.-N. Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake, (2014) Quality of Experience and HTTP Adaptive Streaming: A review of subjective studies. 6th Int Workshop on Quality of Multimedia Experience: 141–146
24. Ghanbari M (2003) Standard codecs: image compression to advanced video coding. IET
25. Hamdan O, Zaidan BB, Zaidan AA, Jalab HA, Shabbir M, Al-Nabhani Y (2010) New comparative study between DES, 3DES and AES within nine factors. J Comput 2(3):152–157
26. Hamidouche W, Farajallah M, Ould-Sidaty N, El Assad S, Déforges O (2017) Real-time selective video encryption based on the chaos system in scalable HEVC extension. Sig Proces Image Commun 58:73–86
27. Hofbauer H, Uhl A, Unterweger A (2014) Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption. Proc IEEE Int Conf Acoust Speech Signal Proc: 1986–1990
28. Huyn-Thu Q, Ghanbari M (2008) Scope of validity of PSNR in image/video quality assessment. Electron Letts 44(13):800–801
29. K. Suehring (HM software coordinator) (2018) HEVC reference software", [online] https://hevc.hhi.fraunhofer.de/, accessed on 22nd Jan. 2019, 2018

30. K. Suehring (JM software coordinator) (2018) H.264/MPEG-4 AVC reference software", [online] https://avc.hhi.fraunhofer.de/, accessed on 22nd Jan. 2019, 2018
31. Kim D, Jeong J (2006) Fast transcoding algorithm from MPEG2 to H.264. Proc Pacific Rim Symp Image Video Technol: 1067–1074
32. Kim M, Lee H, Sull S (2011) Efficient transform domain transcoding: intra frame of H.264/AVC to JPEG. IEEE Trans Consum Electron 57(3):1362–1369
33. Lee J, Kim S, Lim K, Lee S (2015) A fast CU size decision algorithm for HEVC. IEEE Trans Circ Syst Video Technol 25(3):411–421
34. Lin X, Ma H, Luo L, Chen Y (2012) No-reference video quality assessment in the compressed domain. IEEE Trans Consum Electron 58(2):505–512
35. Liu F, Koenig H (2010) A survey of video encryption algorithms. Comput Sec 25(1):3–15
36. Liu Y, Geurts J, Point JC, Lederer S, Rainer B, Muller C (2013) Dynamic adaptive streaming over CCN: A caching and overhead analysis. Proc IEEE Int Conf Commun: 3629–3633
37. Liu Y, Cheng C, Gue T et al (2016) A lightweight authenticated communication scheme for smart grid. IEEE Sensors J 26(3):836–842
38. Lookabaugh T, Sicker DC (2004) Selective encryption for consumer applications. IEEE Commun Mag 42(5):124–129
39. Marr D, Hildreth E (1980) Theory of edge detection. Proc R Soc Lond B 207:187–217
40. Massoudi AA, Lefebvre F, De Vleeschouwer C, Macq B, Quisquater J-J (2008) Overview on selective encryption of image and video: Challenges and perspective. EURASIP J Info Sec (5): 1–18
41. Merkle P, Wang Y, Müller K, Smolic KA, Wiegand T (2009) Video plus depth format for mobile 3D services. Proc IEEE 3DTV Conf: 1–4
42. Moiron S, Faria S, Assunção P, Silva V, Navarro A (2007) H.264/AVC to MPEG-2 video transcoding architecture. Proc Conf Telecomm: 449–452
43. Morrison DG, Nilsson ME, Ghanbari M (1994) Reduction of bit-rate of compressed video while in its coded form. Proc 6th Int Workshop on Packet Video: 392–406
44. Muhammad K, Hamza R, Ahmad J, Lloret J, Wang H, Baik SW (2018) Secure surveillance network for IoT systems using probabilistic image encryption. IEEE Trans Indust Inform 14(8):3679–3689
45. Ohm J, Sullivan G, Schwarz H, Tan T, Wiegand T (2012) Comparison of the coding efficiency of video coding standards — including high efficiency video coding (HEVC). IEEE Trans Circ Syst Video Technol. 22(12):1669–1684
46. Pan Z, Kwong S, Sun M-T, Lei J (2014) Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC. IEEE Trans Broadcast 60(2):405–412
47. Pande A, Mohapatra P, Zambreno J (2013) Securing multimedia content using joint compression and encryption. IEEE Multimed Mag 20(4):50–61
48. Pantos R, May W (2011) HTTP live streaming. Internet Draft
49. Peixoto E, Izquierdo E (2012) A complexity-scalable transcoder from H.264/AVC to the new HEVC codec. Proc IEEE Int Conf Image Process: 737–740
50. L. Pham Van, J. De Praeter, G. Van Wallendael, S. Van Leuven, J. De Cock and R. Van de Walle, "Efficient bit rate transcoding for high efficiency video coding", IEEE Trans Multimed, vol. 18, no. 3, pp. 364–378, 2016.
51. Popov A (2015) Prohibiting RC4 cipher suites. RFC 7465, Internet Eng. Task Force (IETF)
52. Preishuber M, Hütter T, Katzenbeisser S, Uhl A (2018) Depreciating motivation and empirical security analysis of chaos-based image and video encryption. IEEE Trans Info Forensics Sec 13(9):2137–2150
53. Robitza W (2017) FFmpeg VBR settings. [online] https://slhck.info/video/2017/02/24/vbr-settings.html, accessed on 16th Nov. 2018, 2017
54. Rothke B (2007) A look at the advanced encryption standard (AES), 6th edition. Info Sec Manag Handbook: 1151–1158, CRC Press, Boca Raton, FL
55. Safavi-Naini R, Shepard NP (2010) Digital rights management. In: Rosenburg B (ed) Handbook of financial cryptography and security. Chapman and Hall/CRC, Boca Raton, FL, pp 193–220
56. Saleh MA, Tahir NMD, Hashim AH (2018) Fast selective encryption for video stream of high efficiency video coding standard. J Theor Appl Info Technol 96(20):6806–6816
57. Sallam AI, El-Rabaie EM, Faragallah OS (2018) Efficient HEVC selective stream encryption using chaotic logistic map. Multimed Syst 24(4):419–437
58. Seufert M, Egger S, Slanina H, Zinner T, Hoβfeld T, Tran-Gia P (2015) A survey on quality of experience of HTTP adaptive streaming. IEEE Commun Surv Tutor 17(1):469–492
59. Shahid Z, Puech W (2014) Visual protection of HEVC video by selective encryption of CABAC binstrings. IEEE Trans Multimed 16(1):24–36
60. Shen B, Lee S, Basu S (2004) Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks. IEEE Trans Multimed 6(2):375–386

61. Shen L, Zhang Z, Liu Z (2014) Effective CU size decision for HEVC intra-coding. IEEE Trans Image Process 23(10):4232–4241
62. Socek D, Kalva H, Magliveras SS, Marques O, Culibrk D, Furht B (2006) A permutation-based correlation-preserving encryption method for digital videos. Proc Int Conf Image Anal Recogn: 547–558, .
63. Stinson D (2006) Cryptography: theory and practice. CRC Press, Boca Raton, FL
64. Sun H, Chen X, Chiang T (2005) Digital video transcoding for transmission and storage. CRC Press, Boca Raton, FL
65. Taleb T, Ksentini A, Jäntti R (2016) Anything as a service, for 5G mobile systems. IEEE Netw 30(5):84–91
66. Tamanna S (2013) Transcoding H.265/HEVC", M.S. thesis, BTH, Karlskrona, Sweden
67. Thomas NM, Lefol D, Bull DR, Redmill D (2007) A novel secure H.264 transcoder using selective encryption. IEEE Int Conf Image Process: 85–88
68. Thomas N, Redmill D, Bull D (2010) Secure transcoders for single layer video data. Signal Process: Image Commun 25(3):196–207
69. Umbaugh SE (2010) Digital image processing and analysis: human and computer vision applications with CVIP tools (2nd ed.). CRC Press, Boca Raton, FL
70. Van Wallendael G, Boho A, De Cock J, Munteanu A, Van de Walle R (2013) Encryption of high efficiency video coding with video adaptation capabilities. IEEE Trans Consum Elec 59(3):634–642
71. Venčkauskas A, Morkevicius N, Bagdonas K, Damaševičius R, Maskeliunas R (2018) A lightweight protocol for secure video streaming. Sensors 18(5)
72. Vetro A, Christopoulos C, Sun H (2003) Video transcoding architectures and techniques: an overview. IEEE Signal Process Mag 18(2):18–29
73. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
74. Wee S, Apostopoulos G (2001) Secure scalable video streaming for wireless networks. Proc Int Conf Acoust Speech Signal Process: 2049–2052
75. Wiegand T, Sullivan GJ, Bjøntegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circ Syst Video Technol 13(7):560–576
76. Xin J, Lin CW, Sun MT (2005) Digital video transcoding. Proc IEEE 91(1):84–96
77. Xiong J, Li H, Wu Q, Meng F (2014) A fast HEVC inter CU selection method based on pyramid motion divergence. IEEE Trans Multimed 16(2):559–564
78. Yang M, Zhuo L, Zhang J, Li X (2015) An efficient format compliant video encryption scheme for HEVC bitstream. Proc IEEE Int Conf Progress in Informatics and Computing: 374–378
79. Zhang X, Seo S-H, Wang C A lightweight encryption method for privacy protection in surveillance videos. IEEE Access 6:18074–18087
80. Zhou J, Liu X, Member OCA, Tang YY (2014) Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation. IEEE Trans Info Foren Sec 9(1):39–50

Rizwan Ali Shah received his Master degree of Computer Science from the Virtual University of Pakistan and the M. Phil degree in Computer Science, with the major in Computer Network Security, from the Department of

Computer Science & IT (DCS&IT), The Islamia University of Bahawalpur (IUB), Pakistan. Currently, he is an enrolled PhD Student in DCS & IT, IUB. He is an active Research member of the Multimedia Research Group in DCS & IT, IUB. He now works as a Computer Instructor in Federal Government Educational Institutes (Cantt. /Garrison). He has more than 5 years of teaching and R&D experience. He has also published an International Conference paper. His research interests include security aspects of multimedia (audio and video), compression, encryption, secure transcoding and secure multimedia transmission.

Mamoona N. Asghar received her bachelors degree in computer science from the Islamia University of Bahawalpur, Punjab, Pakistan, and her masters degree in Computer Science with the major in computer networks security, from the International Islamic University, Islamabad, Pakistan. She has earned her PhD degree with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK. She graduated in her doctoral degree in 2013. Currently, she is working as an Assistant Professor with the Department of Computer Science and Information Technology (DCS & IT), The Islamia University of Bahawalpur (IUB) Punjab, Pakistan. She has more than 12 years of teaching and R&D experience. She has published several ISI indexed journal articles with numerous International conference papers. She is also working as reviewer of well-known journals (including IEEE Transactions of Dependable and Secure Computing and IEEE Transactions on Information Forensics and Security) and conferences. She has received a Competitive Research Grant awarded by the Higher Education Commission (HEC) Pakistan under National Research Program for Universities (NRPU) 2016. She is also the Head of ongoing projects in the Multimedia Research Group in DCS & IT, IUB. Her research interests include security aspects of multimedia (image, audio and video), compression, encryption, steganography, secure transmission, video quality metrics, and key management schemes for standard and scalable video.

Saima Abdullah received her Ph.D. from the Department of Computer Science and Electronic Engineering of the University of Essex, UK. She is currently working as an assistant professor at the Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Pakistan. Her main research interests include wireless networks/communications, future Internet technology and network performance analysis. She has published around ten papers in the above research areas. She also serves as a reviewer of international journals. She is a member of the Multimedia Research Group in DCS & IT, and has been working on efficient and secure communication of multimedia data over future generation network technologies.

Martin Fleury holds a degree in Modern History (Oxford University, UK) and a Maths/Physics-based degree from the Open University, Milton Keynes, UK. He obtained an MSc in Astrophysics from Queen Mary College, University of London, UK in 1990 and an MSc from the University of South-West England, Bristol in Parallel Computing Systems in 1991. He subsequently gained a PhD in Parallel Image-Processing Systems from the University of Essex, Colchester, UK. He worked as a Senior Lecturer at the University of Essex, where until 2018 he remained a Visiting Fellow and Lecturer. He is now a free-lance consultant. Martin has authored or co-authored around two hundred and ninety articles and book chapters on topics such as document and image compression algorithms, performance prediction of parallel systems, software engineering, reconfigurable hardware, and vision systems. His current research interests are video communication over wireless networks, video quality assessment, and multimedia security. He has published or edited books on high-performance computing for image processing and peer-to-peer streaming.

Neelam Gohar is an Assistant Professor, the Coordinator of Advanced Studies and is in-charge of the Department of Computer Science, Shaheed Benazir Bhutto Women University Peshawar. She completed her PhD from the University of Liverpool, UK in 2012. Her research areas are security aspects in artificial intelligence, multi-agent decision problems, computational social choice theory and voting systems.