A Cloud-based Transcoding with Partial Content Protection Scheme

Rizwan Shah[1], Muqaddas Hashmi[2], Mamoona Asghar[3]
Deptt. of Computer Science & IT, The Islamia University of Bahawalpur
Pakistan
rizali4u@yahoo.com
rizali4u@yahoo.com
mamona.asghar@iub.edu.pk

Martin Fleury
School of Computer Science and Electronic Systems
University of Essex, Colchester, United Kingdom
United Kingdom
fleury.martin55@gmail.com

ABSTRACT: *Multimedia content is now routinely distributed between devices across global networks. These devices differ in their video rendering capability in terms of frame rate, quality, and spatial resolution. To facilitate content exchange between such devices it is necessary to transcode the video format; otherwise no exchange can take place. To prolong battery life on mobile devices, transcoding may take place remotely on a cloud data center and in which case content protection is advisable. This paper presents an effective multimedia content protection technique that removes the need to decrypt the video prior to transcoding. It does this by partially encrypting the compressed video in such a way that it is decoder-format compliant. The demonstrated scheme allows the transcoder to transrate the video to a desired bit-rate without spending time in encryption/ decryption before decoding the video. In this way, the content and decryption keys are not exposed to third party software at the remote cloud data center and there is no need for complex key management software at the cloud. Consequently, the proposed scheme significantly simplifies cloud-based processing compared to previous schemes.*

Keywords: Cloud computing; content protection; Partial encryption; Transcoding; Video streaming

Received: 19 June 2016, Revised 21 July 2016, Accepted 14 August 2016

1. Introduction

A plethora of portable devices such as smart phones and tablets are multimedia-enabled and network connected. This is a very different scenario to that existing in traditional broadcast TV in which there is one fixed device, the television, with a single frame rate, 50 or 60 frame/s and usually a single spatial resolution, Standard Definition (SD) (for example, 640 × 480 pixels/frame). For digital TV, quality is determined by the allocated bitrate of the TV channel within the channel multiplex. Content protection for Pay-TV is through full encryption because it is unlikely that there will be any intermediate processing before the video signal enters a set-top box from the cable or satellite source.

In contrast, in networked video, including Internet Protocol TV (IPTV), there may well be a need to change the format of the compressed video signal according to the display or rendering capabilities of the target device. The quantization parameter (QP) determines the extent of compression (necessary to make bandwidth consumption manageable) which also impacts on the processing required at the target device. The frame rate is also governed by the device processing capability, particularly if a portable device needs to prolong its battery life before re-charging. The target device may have a spatial resolution as low as Common Intermediate Format (CIF) (352 × 288 pixels/frame) though portable devices now support SD, one of the High Definitions (HDs) (for example 1280 × 720 pixels/frame), or recently even Ultra HD (UHD). To cope with such diversity an intermediate transcoder is required within the network path of the video.

Video transcoding [1] is the process of converting a video from one format into another according to one or more of the following parameters: bit rate, frame rate, spatial resolution, encoding syntax, and sometimes changing the objects within a scene. In the experiments reported in this paper, transcoding to change the bitrate by altering the QP is reported, which was the original use of transcoding in broadcast systems.

There is a further gain from the ability to change the QP, other than alteration of the spatial resolution according to the target device s screen. This gain is the ability to charge for the quality of the delivered video, a service known as „pay-per-quality [2]. Because of that possibility, experiments in Section 5 are conducted with change of QP. Just as with Pay-TV, if a service is charged for there is a need to protect the content, which is usually achieved by encryption. However, significantly the scheme in [2] still requires decryption and re-encryption after transcoding, which significantly increases the complexity and transcoding latency. In this paper, we aim to remove those overheads.

As Section 2 further considers there are many transcoder designs, including those in the spatial and compression or frequency domains and those that seek to reduce drift, caused by a loss of synchronization between encoder and decoder due to the insertion of the transcoder. At a cost in computation time, a classical transcoding system fully decodes and re-encodes a video stream, in this paper s scenario by increasing the QP, which reduces the quality and, hence, the bitrate. Fig. 2 shows such a classical transcoding system, which has the merit that no drift can occur. A key feature of Fig. 1 is the need to decrypt the

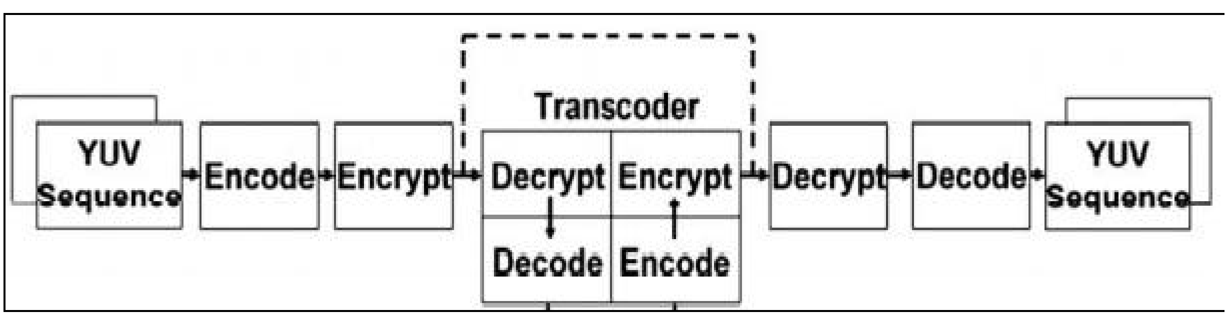


Figure 1. Classical transcoding system

video before decoding and then re-encrypting before re-encrypting. This, of course, leaves the content vulnerable to malware inserted within a remote transcoder. If full encryption is employed then there is no alternative to exposing the content and the decryption keys at the transcoder because otherwise decoding will fail.

This paper’s contribution is employing partial encryption (PE). In the form of partial encryption, only selected syntax elements of the compressed video stream are encrypted. In that way, the encrypted video stream remains decoder compliant so that changing the QP can still take place. In that way also, decryption need only take place after transport over a network to the target device. Moreover, PE only takes place at the final entropy coding stage of processing so that there is no additional bitrate overhead from encryption, as there would be from full encryption.

In the paper's proposed system, transcoding takes place 'on the cloud'. The intensive usage of portable, smart-devices has created a high demand for video transcoding. However, classical transcoding is a computational intensive process, involving both encoding and decoding. For devices that have limited battery capacity this task cannot possibly accomplished by themselves. Therefore, it is natural to move this processing to remote data centers that is through transcoding on the cloud. To establish the feasibility of this approach, that is partial encryption without the need for decryption at the cloud-based transcoder, this paper makes measurements of the computation involved, the file sizes, and the effect of choosing various syntax elements as part of the partial encryption process. There are enormous potential gains from employing PE without the need to decrypt on the cloud because there is no need to distribute the decryption keys and, consequently, there is no need to set up complex key distribution management.

2. Partial Encryption

PE disguises all of the content without completely hiding it, as full encryption would do at a cost in encryption delay and increased bandwidth. By reducing the quantity of data to encrypt, PE reduces the computation involved at the video source. However, not all types of PE can be recommended, because some forms of PE have weaknesses in terms of: confidentiality; introduction of additional bitrate overhead; and decoder compliance. By ensuring that all encryption takes place at the final entropy coding stage and ensuring that the statistical distribution of the encrypted syntax elements is not altered, it is possible to add no extra bitrate overhead, as we do in our method of PE [3].

Moreover, the PE method used in this paper operates on the Context Adaptive Binary Arithmetic Coding (CABAC) form of entropy coding so that it can work both with the H.264/AVC codec and the new codec standard from 2013 High Efficiency Video Coding (HEVC). The CABAC encoder has a number of parameters that can be encrypted, for example: Macroblock (MB) types; Coded Block Flag; Transform Coefficients (TCs); Motion Vector Differences (MVDs); delta quantization parameters (dQPs); and the numerical signs of TCs and MVDs. Not all of these syntax elements preserve decoder compliance and in this paper we select: the signs of MVDs, abbreviated to MV signs; and the signs of TCs, abbreviated in the results to Coeff. Signs. Combining MV signs and signs of TC preserves confidentiality and at the same time ensures decoder compliance. In that way it is possible to perform PE without decryption and encryption when transcoding takes place. Indeed, as the results of Section 5 show confidentiality is still preserved by this form of SE. The paper now discusses differing forms of transcoding, one of which was selected for these experiments because of its beneficial qualities.

3. Background To Transcoding

Transcoders are frequently utilized for transrating (bitrate reduction), principally to match the bandwidth to the compressed video bitrate. There are four techniques [4] for performing bitrate reduction, while at the same time maintaining the spatio-temporal resolution. The first of these techniques is coefficient truncation. The input bitstream is parsed and higher frequency coefficients are removed to match the target bitrate. Re-quantization, the second of these techniques is the main way to perform bitrate control during encoding, by varying the quantization step size to match the target bitrate. The result is a higher compression ratio, caused by decreasing the number of representation levels of the transform coefficients. Another technique is re-encoding while at the same time reusing the motion vectors and mode decisions, which are embedded in the input bitstream. Compared to simply re-quantizing, this technique avoids drift because reference frames are reconstructed and the residual information (the frame difference data) is recompressed. To avoid too many additional calculations, no new motion estimations are made and no mode decisions (between intra- and inter coding) takes place. The fourth technique is an extension of re-encoding but this time the coding mode may be changed. This technique re-uses motion information but modifies coding modes to achieve an optimal coding mode decision based on the desired output bit-rate.

Adaptation to portable devices implies spatial resolution changes which transcoding can address [5], as well as altering the bitrate. Spatial domain transcoding involves a cascaded decoder-encoder pair as in the classical system of Fig. 1. The decoded output is down-sampled before input to the re-encoder. Input motion vectors are reused in the down converted video input and the new motion vectors are computed by means of a mapping function, which speeds up processing. Spatial resolution reduction provides high-quality, error drift-free transcoding. The main problem of down conversion in the frequency domain is to find efficient ways of merging four Inverse DCT (IDCT) into one DCT block. This can be achieved [6] only by utilizing the low-energy coefficients from the four original blocks to produce a new resized block. The associated motion vectors are also down scaled to meet the new block property.

In heterogeneous transcoding, format conversion takes place between different codec standards, such as MPEG-2 to H.263, or MPEG-2 to H.264/Advanced Video Coding (AVC). It can also be combined with some form of the homogeneous transcoding. Heterogeneous transcoding requires codec syntax conversion between the input and output standard. It may also change format parameters to match the target device s capabilities. Due to the asymmetry between the encoder and the decoder, heterogeneous transcoding greatly increases computational complexity [7]. Therefore, transcoding latency is increased over homogeneous transcoding, which implies that interactive applications such as video conferencing will be impacted.

There are several types of transcoding architecture [8]. The cascaded decoder-encoder pair is more costly in terms of computation. A reference frame or picture serves to minimize the difference between the input and output frames and, thus, to reduce error drift (refer to Section 1). Reference frames are stored in a decoded frame buffer.

The open-loop architecture is the fastest and the simplest method of video transcoding. In an open-loop architecture, the output is not measured. Feedback is also not compared to the input. In this architecture error drift is increased due to the removal of high frequency DCT coefficients from the residual information. Because of this, open loop transcoding is best confined to intra-coded frames (frames that only employ spatial coding).

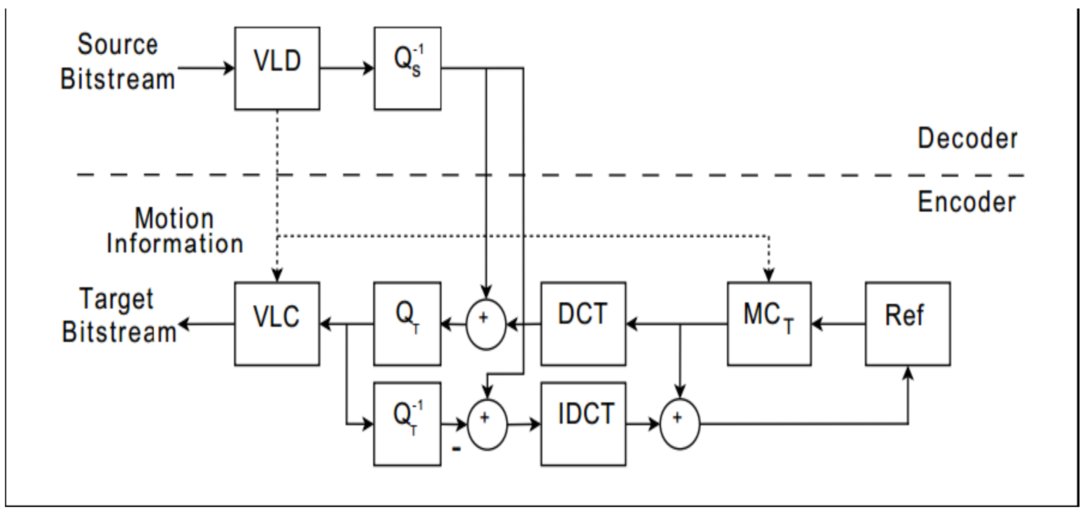


Figure 2. Closed-loop Architecture

A closed-loop architecture is an advanced form of a cascaded decoding and encoding architecture. The error signal, which is the difference between the input and the feedback, is fed to a controller to minimize the output error. For each reference input, feedback (the difference between the actual picture and the desired picture) acts to reduce error drift. Fig. 2 shows this architecture, with VLD/VLC = Variable Length Decoding/Coding (entropy coding), Q = quantization, and MC = Motion compensation.

4. Overview of Cloud Computing

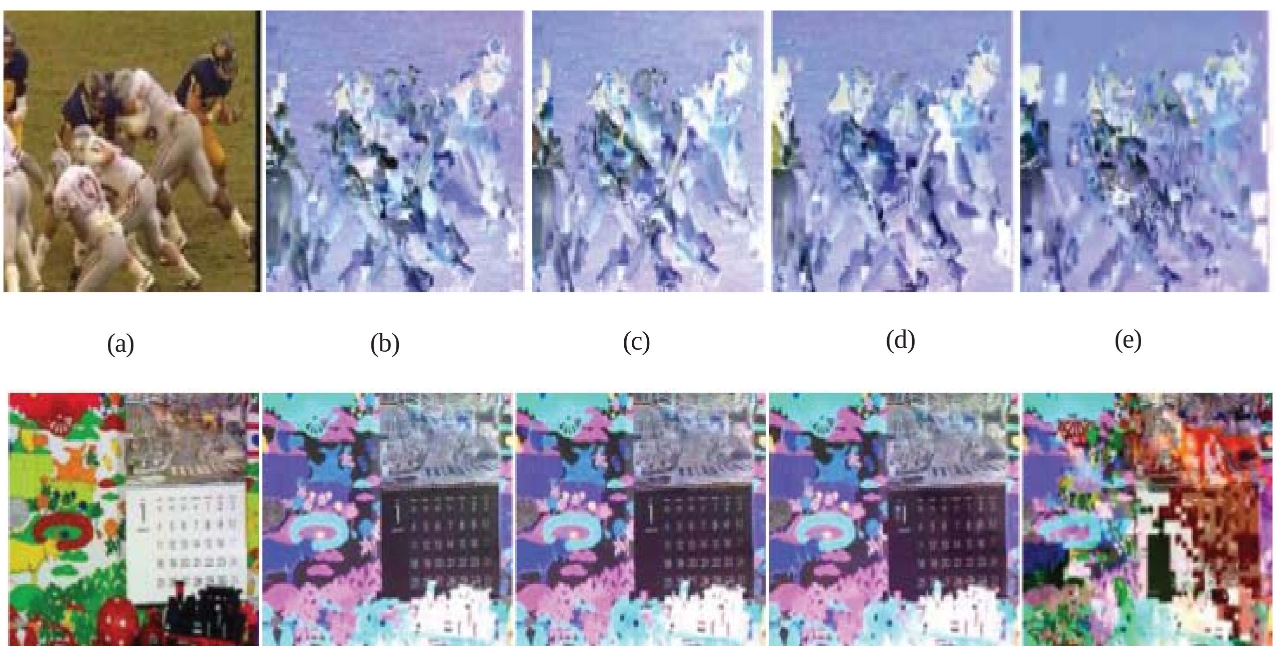
The proposed form of transcoding takes place within a cloud. Cloud computing is based on sharing resources over the public Internet instead of creating data centers locally. Cloud providers are normally large corporations that hire out these resources, servers and data storage, on a demand basis, without the need for purchase. Because these resources are accessible anywhere and at any time, they are referred to as a „cloud which pervades the Internet [9].

In early cloud computing, transcoding was performed by means of proxy servers [10]. In this scheme, a proxy server performs transcoding before streaming the required content to a client via a centralized control system. However, this scheme needs extensive resources to perform transcoding. As a result, a further scheme caches already transcoded videos on cloud [11]. The main saving from this method is that popular video is only transcoded once, though in a variety of formats. However, transcoding is still required in the first instance. Transcoding is conveniently performed on a cloud by Hadoop, which is a simplified form of parallel processing for large-scale data-centers. Hadoop is available either as part of a Platform as a Service (PaaS) cloud facility.

4. Findings

We use the closed loop architecture for transcoding (refer to Section 2.3), which is a good way to remove error drift from the video. However, because a closed loop transcoder is compute intensive the burden of computation should not be further increased by including a need for decryption and re-encryption. Therefore, PE is used, as it can be made decoder compliant. As mentioned in Section 1, this means that there is no need to decrypt the video before decoding. Importantly, the need to distribute decryption keys and expose them at the transcoder is also removed.

Experiments by us confirmed the latency introduced by a need to include decryption and re-encryption into the transcoding cycle. All the experiments were performed on an Intel Core I3 Core 2 Duo (2.10 GHz) processor with 6GB RAM. In a classical closed loop system, see Fig. 2, it takes 2,410 ms for decryption and 83,891 ms for encoding plus encryption. Therefore, the total time required for transcoding in a classical system is 2,410 + 83,891 = 86,301 ms for transcoding a 90-frame *Football* video. Therefore, around 2 × 2,410 ms is gained (assuming re-encryption takes around the same time as encryption), along with reductions in bandwidth. However, the most important security gain is that there is no longer a need to distribute and expose decryption keys on the cloud. Given that a cloud is managed by a third party, not necessarily the cloud infrastructure provider, there is a definite need to assure commercial content providers of the confidentiality of their video. For end users, their privacy is an issue, particularly with user-to-user video streaming.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 3. Visual results with PE on the Football and Mobile video clips transcoded with PE and encoded for 90 frames (I-, and P-frames) and QPs: 12, 24, 36 and 48. (a) Football video frame # 07 [Y=29.8, U=42.2, V=44] dB, SSIM = 0.9405. (b) Transcoded video with PE at QP 12 for Football with CABAC [Y=9.3, U=14.2, V=21.1] dB, SSIM = 0.4776. (c) Transcoded video with PE at QP 24 for Football with CABAC [Y=8.8, U=14.4, V=21.1] dB, SSIM = 0.4675. (d) Transcoded video with PE at QP 36 for Football with CABAC [Y=9.6, U=14.1, V=21.2] dB, SSIM = 0.6304. (e) Transcoded video with PE at QP 48 for Football with CABAC [Y=10.3, U=14.1, V=21] dB, SSIM = 0.6433. (f) Mobile video frame # 54 [Y=27.7, U=40.5, V=37.5] dB, SSIM = 0.9446. (g) Transcoded video with PE at QP 12 for Mobile with CABAC [Y=6.7, U=12.9, V=13.2] dB, SSIM = 0.0637. (h) Transcoded video with PE at QP 24 for Mobile with CABAC [Y=6.7, U=12.9, V=13.3] dB, SSIM = 0.0698. (i) Transcoded video with PE at QP 36 for Mobile with CABAC [Y=6.6, U=12.9, V=13.4] dB, SSIM = 0.0805. (j) Transcoded video with PE at QP 48 for Mobile with CABAC [Y=7.6, U=14.4, V=13.2] dB, SSIM = 0.0690.

Experiments were conducted on dissimilar video clips i.e. Football, Mobile, and News, with different QPs (QP = 12, 24, 36 and 48). A QP of 12 is broadcast quality, while a QP of 48 is of coarse quality in the H.264/AVC codec (as the maximum QP = 51). Fig 3 shows frames from two of the video sequences with average Peak Signal to Noise Ratio (PSNR), an objective measure of video quality measured in decibels (dBs) and Structural Similarity (SSIM) index, which aims to capture the human perceptual response on a scale 0 to 1. Fig. 3 also shows the effect of applying PE to the selected frames, illustrating the ability of PE to mask the content, making the clips unwatchable. Table 1 is a comparison of the different average PSNR (dB) of 90 frames at different QPs, i.e after transcoding to different qualities. The Table shows how the PSNR after PE is reduced to a low level according to the Y luminance component and the two chrominance components, U and V. In all cases, even when the components are separated out the average quality is poor, as a PSNR below 25 dB is usually rated as poor.

Videos	**Without transrated luminance (Y)**	**Transrated with PE of Y**	**Without transrated chrominance U**	**Transrated with PE of chrominance U**	**Without transrated chrominance V**	**Transrated with PE of V**
Transcoded CIF videos at QP 12: CABAC-based scheme						
Football	34.0	9.3	41.6	13.1	44.6	21.6
Mobile	30.6	6.7	41.8	13.2	40.3	13.6
News	39.8	4.6	46.4	16.0	48.4	21.4
Transcoded CIF videos at QP 24: CABAC-based scheme						
Football	32.2	8.5	39.2	13.2	41.8	21.7
Mobile	28.9	6.7	38.4	13.1	37.2	13.6
News	37.6	4.6	42.8	16.1	44.4	21.5
Transcoded CIF videos at QP 36: CABAC-based scheme						
Football	30.2	9.8	36.5	13.3	38.9	21.8
Mobile	26.6	6.7	34.1	13.0	33.2	13.7
News	33.8	5.0	38.5	16.2	39.7	21.7
Transcoded CIF videos at QP 48: CABAC-based scheme						
Football	28.1	10.4	34.5	13.1	37.2	22.1
Mobile	23.9	7.1	31.8	14.2	30.8	13.3
News	29.7	5.1	35.9	16.3	37.5	22.0

Table 1. Comparison of PSNR of with and without transcoded YUV videos at different QPs. (Units of measurement are all dBs)

In more detail, our CABAC-based entropy coding scheme [3] was adopted to discriminate between the transcoded sequences after PE. In the experiments, the Group of Pictures (GOP) size was eight, with the H.264/AVC Baseline profile, which does not include B-frames. CIF resolution was employed throughout. Thus the video configuration was suitable for the lowest rated mobile devices in the marketplace today. See Section 2 for the PE configuration.

Fig. 4 demonstrates the relationship between QP and file size. The bar chat confirms that that there is an inverse relationship between QP value and file size, i.e. if the QP is increased the size of the video file will decrease and vice versa. Therefore, transcoding to a different QP, apart from changing the quality in a pay-per-quality scheme, also reduces the bitrate and, hence, the bandwidth consumption. There is a content dependency, as News at the same QP as the other two videos has a lower file size. The file sizes also only approximately change in the same way with a change in QP. In general, even for a few frames (90 in all) and a low resolution, the file sizes are considerable and encryption time would increase if full encryption were to be used.

However, for PE the encryption time is reduced. Even with PE the original encryption time at source cannot be ignored as Figs.

5, 6, and 7 shows respectively for the three test video sequences. These timings can give an indication of the timings

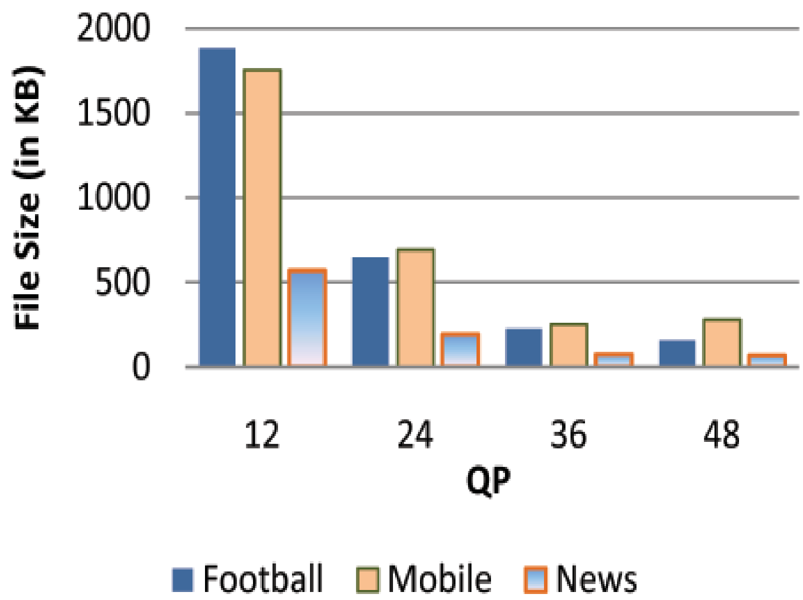


Figure 4. Comparison of file size with respect to QP

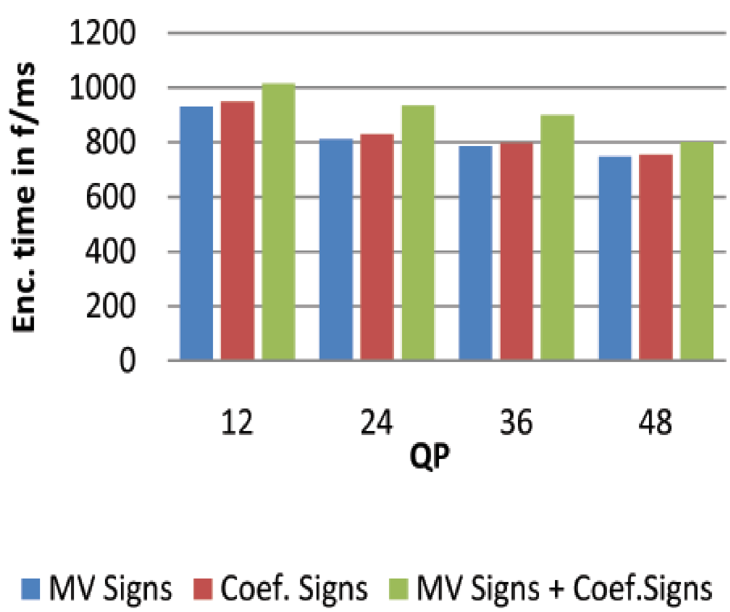


Figure 5. Encoding times for the Football video clip according to target QP and PE method

were full decryption and encryption to be used at the transcoder. Recall that in the proposed scheme, no decryption and re-encryption takes place at the transcoder. The timings are shown according to whether just one or both the selected syntax elements were encrypted. In Fig. 3 both the syntax components were used, whereas employing just one of the elements would significantly reduce the ability of the PE scheme to mask the video content. Another factor brought out by the Figs. is that encryption time decreases somewhat with QP.

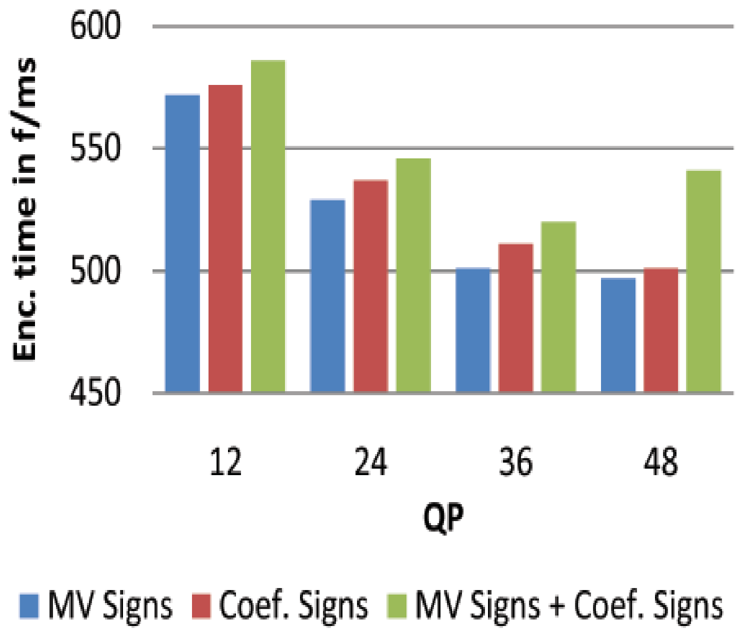


Figure 6. Encoding times for the Mobile video according to target QP and PE method

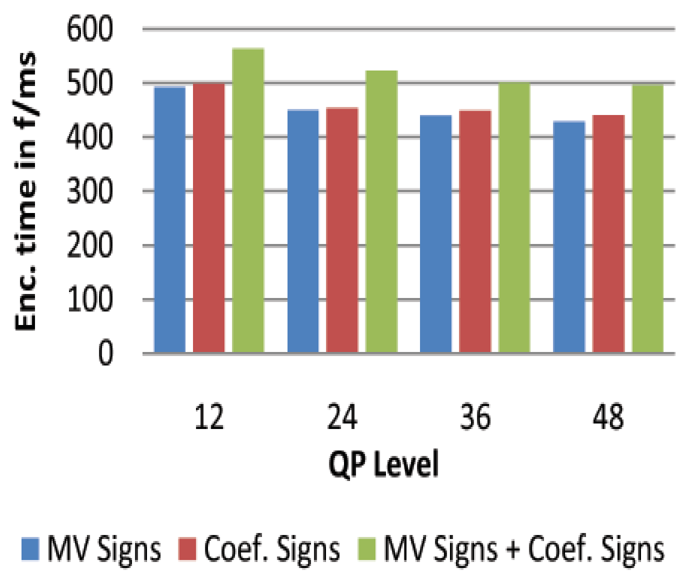


Figure 7. Encoding times for the News video according to target QP and PE method

5. Conclusion

The proposed cloud-based transcoding scheme avoids the principal weakness of previous schemes, the need to decrypt and re-encrypt after transcoding has taken place. This was achieved by decoder-compliant PE, which means that the transcoder no longer needs to decrypt the input video stream before decoding. Of course, an encryption scheme, including full encryption, would cause the decoder to crash if the stream was not format compliant. The PE scheme, based around the CABAC coder as it is, is suitable both for the widely deployed H.264/AVC codec and the more efficient HEVC, which Currently does not have a low-energy hardware implementation. Initial results have shown the masking behavior of the PE method and the operating parameters of the scheme in terms of file sizes and encoding time. Future work should compare full encryption and key management overheads with the PE-based, decoder-compliant scheme. Spatial resolution transcoding, as well as transrating, can be investigated.

References

[1] Vetro, A., Christopoulos, C., Sun, H. (2003). Video transcoding architectures and techniques: An overview, p. 18-29, *IEEE Signal Process. Mag.*, March

[2] Díaz-Sánchez, D., Sánchez-Guerrero, R., Aries, P., Almenarez, F., Marín, A. (2016). A distributed transcoding and content protection system, p. 59-76, Telecomm. Syst., vol. 61

[3] Asghar, M.N., Ghanbari, M., Fleury, M., Reed, M.J. (2012). Efficient selective encryption with H.264/SVC CABAC bin-strings, p. 2645-2648, *In*: Proc. of the IEEE Int. Conf. on Image Processing

[4] Morrison, D. G., Nilsson, M. E., Ghanbari, M. (1994). Reduction of bit-rate of compressed video while in its coded form, *In*: Proc. of the 6th Int. Workshop on Packet Video, p. 392-406

[5] Yin, P., Vetro, A., Lui, B., Sun, H. (2002). Drift compensation for reduced spatial resolution transcoding, p. 1009-1020, *IEEE Trans. Circuits Syst. Video Technol.*, 12 (11).

[6] Assunção, P.A.A., Ghanbari, M. (1998). A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams, p. 953-967, *IEEE Trans. Circuits Syst. Video Technol.*, 8 (8).

[7] Shanableh, T., Ghanbari, M. (2000). Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats, p. 101-110, *IEEE Trans. Multimedia*, 2 (2).

[8] Youn, J., Xin, J., Sun, M.T. (2000). Fast video transcoding architectures for networked multimedia, p. 25-28, *In*: Proc. of the IEEE Int. Symp. Circuits and Systems, vol. 4,

[9] Larus, J. R. (2011). The cloud will change everything, p. 1-2, Proc. of the Int. Conf. Architectural Support for Programming Languages and Operating Systems, March

[10] Bharadvaj, H., Joshi, A., Auephanwiriyakul, S. (1998). An active transcoding proxy to support mobile web access, *In*: Proc. of the IEEE Sym. on Reliable Distrib. Syst.

[11]Shen, B., Lee, S., Basu, S. (2004). Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks, p. 375–386, *IEEE Trans. on Multimedia*, 6 (2).