Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com

Transparent encryption with scalable video communication: Lower-latency, CABAC-based schemes☆

Mamoona N. Asghar[a], Rukhsana Kousar[a], Hooriya Majid[a], Martin Fleury[b,*]

[a] The Islamia University of Bahawalpur, Department of Computer Science & Information Technology and UCET, Pakistan
[b] University of Essex, Colchester, United Kingdom

ARTICLE INFO

Article history:
Received 21 July 2016
Received in revised form 10 November 2016
Accepted 21 February 2017
Available online xxx

Keywords:
B-frames
Scalable video streaming
Reduced encryption
Selective encryption
Transparent encryption

ABSTRACT

Selective encryption masks all of the content without completely hiding it, as full encryption would do at a cost in encryption delay and increased bandwidth. Many commercial applications of video encryption do not even require selective encryption, because greater utility can be gained from transparent encryption, i.e. allowing prospective viewers to glimpse a reduced quality version of the content as a taster. Our lightweight selective encryption scheme when applied to scalable video coding is well suited to transparent encryption. The paper illustrates the gains in reducing delay and increased distortion arising from a transparent encryption that leaves reduced quality base layer in the clear. Reduced encryption of B-frames is a further step beyond transparent encryption in which the computational overhead reduction is traded against content security and limited distortion. This spectrum of video encryption possibilities is analyzed in this paper, though all of the schemes maintain decoder compatibility and add no bitrate overhead as a result of jointly encoding and encrypting the input video by virtue of carefully selecting the entropy coding parameters that are encrypted. The schemes are suitable both for H.264 and HEVC codecs, though demonstrated in the paper for H.264. Selected Content Adaptive Binary Arithmetic Coding (CABAC) parameters are encrypted by a lightweight Exclusive OR technique, which is chosen for practicality.

1. Introduction

Scalable video communication [1] is a way of simplifying adaptation both to network conditions and capacity as well as to display device resolution and processing speed. As this paper discusses, by virtue of its layered structure it naturally supports transparent encryption, a form of encryption that hides access to high-quality enhancement layers (ELs) but allows a lower quality version of an original video stream to be visible. The Joint Video Team of the ITU-T VCEG and the ISO/IEC MPEG has standardized Scalable Video Coding (SVC), which is an extension of the H.264/Advanced Video Coding (AVC) standard [2]. H.264/SVC [3] permits the transmission and decoding of partial bit-streams to provide video services at various temporal, spatial and/or quality resolutions, which itself requires encryption transparency to allow access to those devices that decompose a bitstream. At the same time H.264/SVC preserves a reconstruction quality that is high enough relative to the rate of the partial bit-streams. Because the bitrate overhead for spatial scalability compared to single-layer H.264/AVC is at most 10% [3], commercial developers can be more confident that adopting this technology will not seriously handicap their application.

The trend towards scalable video has been maintained in the High Efficiency Video Coding (HEVC) standard codec [4] with two scalable extensions [5,6] available by July 2014. However, because of their limited deployment at the time of this research, this paper is confined to H.264/SVC. Nonetheless, because the encryption method analyzed in the paper operates on the Context Adaptive Binary Arithmetic Coding (CABAC) form of entropy coding (refer to Section 2.3) it can be converted [7] to work on HEVC. Despite positive SVC developments, including a software-based multi-endpoint video conferencing system [8], commercial developers must also be confident that the confidentiality of their content is protected on the public Internet, due to the risk [9] of illegal copying and redistribution. Hence, the topic of this paper is transparent encryption [10], which, as mentioned, is a commercially-aware form of encryption that heightens a viewer's interest with a debased quality version of the original but dampens any appetite for pirated copies because of the difficulty of extracting a high-quality version of the original video. Notice that transparent encryption is also known as perceptual encryption, with an analysis of such single-layer schemes and their desirable features in [11]. As remarked in [12], commercial applications of video communication often rely on encryption of a video stream that is only viewable by the end user upon payment for a decryption key. A variety of business models are opened up by the possibility of transparent encryption. For example, it offers a means to promote a service to viewers currently not subscribed to a service. Thus, if a viewer is interested in viewing a sports TV channel but is unsure if its contents meets their needs then a lower-quality version can be seen for free. If that viewer decides that they do wish to subscribe then they can purchase the key(s) to ELs. In fact, a differentiated service could be of-

? This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.

Email address: fleum@essex.ac.uk (M. Fleury)

http://dx.doi.org/10.1016/j.jvcir.2017.02.017
1047-3203/© 2016 Published by Elsevier Ltd.

fered which allows grades of service according to which access keys are purchased. The business model exploits the perceived desire to view higher quality (Signal-to-Noise Ratio (SNR)) video, which is not distorted in any way. Different spatial (or temporal) resolution videos could be sent as tasters but these are intended to be at a lower SNR to satisfy the business model. It is anticipated that a viewer would soon become tired of watching video with frames marred by any distortion, given the right subscription levels. Thus, even if a viewer set the decoder to extract only the base layer, it would remain distorted at whatever spatial or temporal scalability the viewer had the rights to.

The scalable structure of an H.264/SVC video stream is contained within the Network Abstraction Layer (NAL) unit headers that are output by an encoder to encapsulate the compressed content. Subsequently Media Aware Network Elements (MANEs), which are usually untrusted devices because of the expense and/or inconvenience of making them tamper proof, are able to discard those partial bitstreams that are unsuitable for a target device, without the need to decrypt the compressed content. Consequently, full encryption including NAL unit headers within scalable video streams is actually harmful to scalability [12]. In our solution to this requirement, we also provide adaptation-transparency, allowing scalable layers to be discarded by a MANE if they are not needed by a target display device. This further type of transparency is achieved by ensuring that the encryption is decoder format compliant, because a MANE must partially decode the stream in order to discard parts of it. By confining our encryption to the entropy coding stage of a codec, the bitstream statistical characteristics are able to be maintained. By also choosing to encrypt only those elements that during entropy coding will not impact the statistics, the bitstream remains the same size. There is a further form of transparency that allows a transcoder to alter the quantization parameter (QP) and subsequently re-scale the transform coefficients to reduce quality. This form of transparency may even be provided for scalable video that already can have quality scalability built into its ELs. The reason for this variant of transparent encryption is that the quality of the layers can be retrospectively adjusted by a transcoder. As the syntax elements selected in our encryption schemes do not include the QP, transcoder transparency follows.

For some legal and military applications, full encryption without regard to the internal structure of the compressed video contents is desirable but for other commercial applications, in order to meet real-time constraints, selective (partial) or transparent encryption of the video [13] may well be more appropriate, depending on the application. Notice that the set of selective encryption (SE) methods contains transparent encryption as a subset. Provided the compressed video statistics are maintained then there will be no bandwidth overhead arising from SE, unlike its full encryption counterpart. Likely commercial applications of transparent encryption are pay-per-view videos, pay-TV, and video-on demand. In terms of content protection, transparent encryption is preferable to hardware scrambling, many types of which have been broken. It should be borne in mind that encryption is the first line of content defense but it is not the only means: compliancy rules within device licenses [9] are one form of legal protection; and fingerprinting of video (embedding of identifiers into the compressed bitstream) [14] is a means of tracing and revoking illicit copies [15]. In fact, in [9], pushing the argument further, encryption is identified as another means by which content access can be licensed. Without encryption there might be nothing to license.

Due also to a requirement to display video at rates of 60 frames per second (fps) or more for higher-definition video, processing of live and interactive video streams needs to be expedited. By reducing the amount of data to encrypt, SE reduces the computation involved at the video server. All the same, not all types of SE can be recommended, which is why the form of encryption should be carefully considered both in respect to confidentiality but also in respect to various side effects that may arise. For example, in [16] the method of [17] for scalable video ELs, which scrambles the scan pattern of transform coefficients prior to encryption, is said to introduce about 17% bitrate overhead because the statistical properties of the scan order are disrupted.

In commercial applications, transparent encryption offers a means to promote a service to viewers currently not subscribed to that service. Transparent encryption enables soft video degradation but should not permit access to a better quality version of the video through a replacement or reconstruction attack [18] or another such attack depending on the form of SE. If a viewer is interested in viewing a sports channel but is unsure if its contents meet their needs then a lower-quality, preview version can be viewed. If that viewer decides that they do wish to subscribe then they can purchase the full-resolution service, whereupon, after authentication, keys for individual layers or a single key [19] for all scalable layers can be supplied, provided, of course, that suitable mechanisms are in place to prevent access to unauthorized layers. (Further discussion of key management is outside the scope of this paper but is surveyed in [20].) If an end user is to judge the suitability of video for their use without accessing the full-quality version, one way to do this is to permit access to a distorted view. That objective is only achievable if the encrypted video bitstream is decoder format compliant, because otherwise the debased version of the video cannot be de-compressed.

As remarked earlier, the means of transparent encryption developed in this paper is transferable to HEVC by the method proposed in [7], which concerns how to convert single-layer H.264/AVC with CABAC-based SE into the HEVC version of CABAC. A further problem, that encryption must maintain the CABAC context in order to preserve decoder format compliance, is also resolved in [7]. Other H.264 SE methods may not be convertible either for theoretical or practical reasons. For example, it appears that the method of [16], which does not encrypt CABAC parameters but instead pseudo-randomly permutes the sub-blocks of H.264/AVC macroblocks (MBs), may encounter an implementation problem if transferred to HEVC. This is because of the great variety of sub-block configurations [21] that are employed in HEVC, raising the practical difficulty of devising a permutation scheme for all of the configurations that can be selected from in rate-distortion analysis.

Prior work by the authors of this paper includes the original analysis of a SE scheme in [22], elaborated in [23], and extended to include key management in [24], none of which work included the further adaptation to transparent encryption or the second scheme of this paper. In this paper we propose two schemes: the first is transparent encryption in which we only encrypt the syntax elements of the ELs but the base layer (BL) remains unencrypted as a low quality ‘taster’ of the high-quality video. Essentially, the first scheme employed is similar to that of [18] by the authors but extended to include transparency. In the second reduced encryption scheme, by way of comparison, we transparently encrypt only bi-predictive B-frames to evaluate the impact on transparency. That is we evaluate whether transparent encryption of only B-frames is able to provide a ‘taster’ of a video stream contents. This procedure leaves anchor and reference I- and P-frames, which can either by fully encrypted or encrypted by some other SE method. In other words, this paper is neutral on the treatment of frame types other than B-frames, though there is a discussion of various possibilities in Section 3.3. In both these methods, the decoder bitstream remains format compliant. Other work by us has confirmed the resistance to perceptual attacks [25] and examined to what extent the core SE technique can tolerate errors in a wireless channel [26]. The main part of this paper is the evaluation of the practical effectiveness of the two schemes. For that reason, a simplified block coding method of encryption was used in the interests of low complexity and speed of encryption. This eXclusive OR (XOR) scheme replaced the more normal encryption by a stream cipher or a block cipher acting in a stream mode. This met the intention of the research in this paper of reducing encryption delay and increasing the utility of the encryption by means of transparent encryption, at a cost of "reduced encryption" as it were. Overall in order of computation time, timings demonstrate that full SE is the most costly, the transparent encryption scheme decreases the computational overhead further, while the reduced encryption scheme based on encrypting only B-frames takes up the least time. It should be mentioned that the computational overhead of encryption compared to encoding the video without encryption remains small for all encryption schemes.

The rest of this paper is organized as follows. An overview of scalable encoding with H.264/SVC, as well as aspects of CABAC and related research on transparent encryption is presented in Section 2. The following Section 3 gives a concise description of the proposed transparent encryption scheme, before evaluating its impact in terms of video quality and computational overhead in Section 4. Finally, the conclusion in Section 5 discusses the findings from this research and the directions of future research.

2. Context

2.1. H.264/SVC essentials

H.264/SVC is composed, Fig. 1, of a BL, which is compatible with single-layer H.264/AVC, and one or more ELs, which provide video scalability in up to three dimensions (i.e., time, quality and resolution). For example, in Fig. 1 for a device to receive Common Intermediate Format (CIF) (352 × 288 pixels/frame) resolution it would need to receive the BS and EL 1, which would also result in an increase in frame rate. This is because, in SVC, all upper layers of SVC video are predicted from lower layers, Fig. 2, as well as through inter and intra coding within a layer, as appropriate. An important feature of Fig. 1 is that a receiving terminal can control what which layers it receives through feedback to the MANE. Therefore, in a transparent encryption scheme a user at a terminal can request just the base layer as a taster before purchasing a key and requesting one or more ELs.

Fig. 1 illustrates the situation where the temporal and spatial resolutions change but the quality or video distortion remains fixed, as determined by the Quantization Parameter (QP). In Fig. 2, the BL consists of key pictures at a lower frame rate, at a lower picture resolution, and with reduced quality, while ELs 1 and ELs in Fig. 2 have predictively coded frames with the same picture resolution but at different frame rates and SNR. EL 1 utilizes predictively-coded P-frames, whereas EL 2 also includes bi-predictively-coded B-frames.

Any encryption of SVC that accomplishes adaption transparency must act in such a way that an untrusted MANE or other MPEG-21 adaptation engine [27] can access a partial bitstream may be accomplished by not encrypting essential syntax elements of the H.264/SVC VCL contained in NAL units. These include the NAL unit headers and slice headers, elements of which in [28] are also available as an initialization vector (IV) for a stream cipher (or a block cipher in a chained mode such as Advanced Encryption Standard (AES) Cipher Feedback Mode (CFB) acting as a self-synchronizing stream cipher). (IVs are normally sent as plaintext because they do not contain information available to an attacker.) The SVC standard also forbids four specific markers appearing in byte-aligned positions. As it is possible [28] that these could arise as a result of encryption, alternative codewords, again available from the SVC specification to avoid emulating the markers, can be inserted. Additionally encryption should not result in the last byte of a VCL NAL unit being 0x00, which is accomplished by slight modification of the last byte of a VCL NAL [29]. Notice that conversely to our approach, it is also possible [30] to force an H.264/SVC decoder to reject NAL units that are encrypted by employing an unrecognized SVC NAL unit type.

2.2. Overview of SVC with CABAC

There are two entropy coding implements in H.264/SVC one is based on variable length coding (VLC) and the other is based on binary arithmetic coding (BAC) both of which are applied in a context adaptive way, known as Context Adaptive Variable Length Coding (CAVLC) [31] and the other as CABAC [13] (refer to Section 1). The main difference between the two forms of entropy coding is that additional syntax elements are coded with CABAC such as: the intra prediction modes; the MB type; reference picture indexes; and motion vectors. Run-length coding of transform coefficient residuals is exchanged for map coding, which defines Non-Zero (NZ) coefficient positions in 4 × 4 block of coefficients. CABAC [13], which obtains up to a 15% higher compression ratio than CAVLC, is also computed easily on standard to high-complexity decoder devices. Conversely, CAVLC adaptively codes only the residual transform coefficients [32] and will not be considered further herein.

CABAC coding consists of three steps (refer to Fig. 3). The first step is called the binarization and it is the primary step for the CABAC coder. It converts all non-binary syntax elements into bin

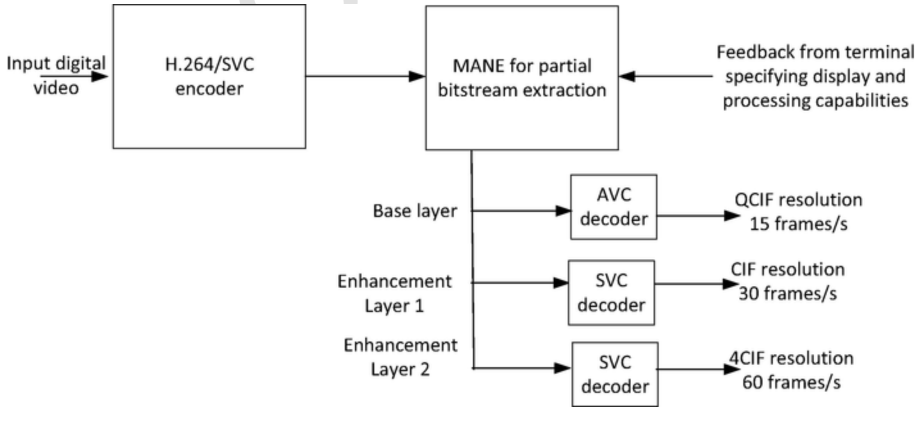

Input digital video
H.264/SVC encoder
MANE for partial bitstream extraction
Feedback from terminal specifying display and processing capabilities
Base layer
AVC decoder
QCIF resolution 15 frames/s
Enhancement Layer 1
SVC decoder
CIF resolution 30 frames/s
Enhancement Layer 2
SVC decoder
4CIF resolution 60 frames/s

Fig. 1. Overview of H.264/SVC with example layers.

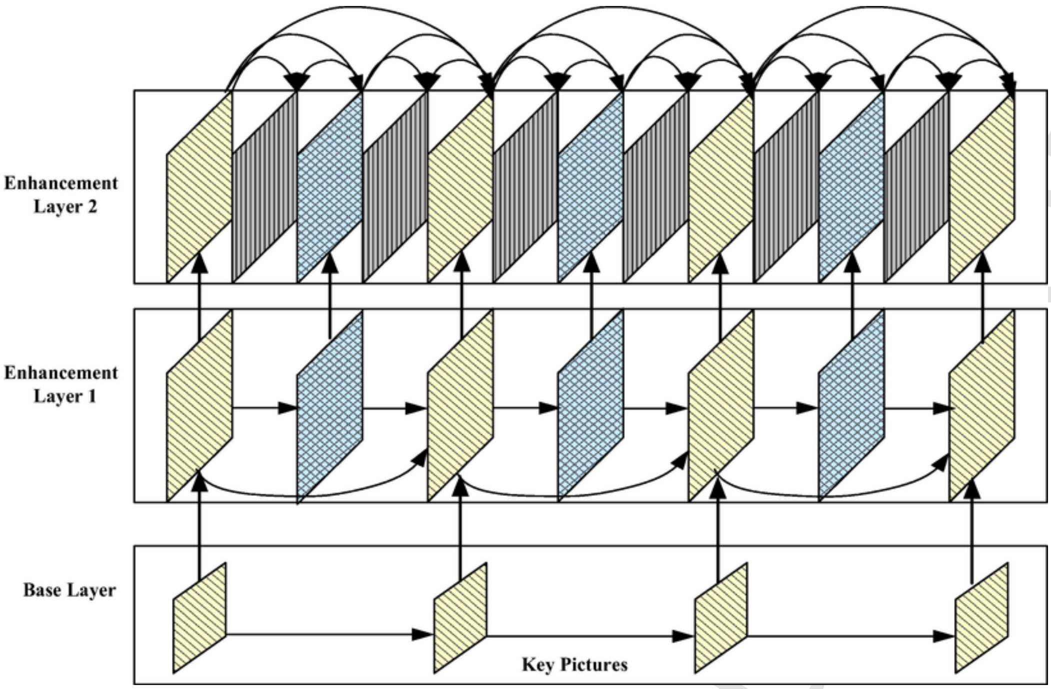


Fig. 2. Combined scalability (temporal, spatial, SNR).

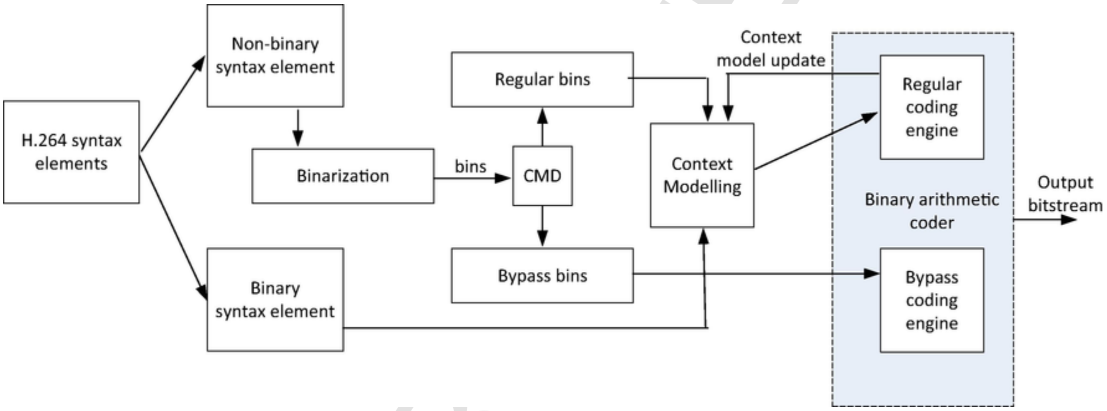


Fig. 3. Top-level view of CABAC coding, CMD = Coding Mode Decision.

strings. Each bin string has a bit position (a bin) which is transferred to either the regular coding mode decision or to the by-pass coding mode. The bins of regular coding mode are forwarded to the second processing step, context modeling (CM), and subsequent to that in the third step the regular BAC engine further codes the stream. The bins of the bypass coding mode do not enter into the CM stage and are straight away transferred to the bypass BAC engine for the purpose of coding. These bins are associated with the sign data of motion vector differences (MVDs) and the sign data of transform coefficient (TCs) levels or for the less important bins which are assumed to be distributed uniformly. In the proposed scheme, this module is chosen to encrypt the syntax elements of the video because by doing so any impact on CM is avoided. It is important to avoid that impact, as otherwise the coding statistics would be altered, which could increase the bitrate overhead.

2.3. Converting H.264 CABAC to HEVC CABAC methods

The encryption method employed by us, described in detail in prior publications such as [24] by us, in this paper relies on CABAC [33] rather than the alternative Context Adaptive Variable–Length Coding (CAVLC) entropy coding mode, which has reduced time complexity but also has around 12% greater bitrate overhead. CABAC can be relatively easily computed on medium- to high-performance decoder devices and is used for encoding a broader range of syntax elements than CAVLC. CABAC is designed to better exploit the features of Non Zero (NZ) coefficients in zigzag scanning and replaces run-length coding by significant maps coding which specifies the position of Non-Zero (NZ) TCs within a 4 × 4 block.

To understand the way the H.264 CABAC to HEVC conversion process works requires an explanation or digression, which might be passed over in a first reading. In both H.264 and HEVC the m-ary arithmetic coder of an H.263 encoder is replaced by a binary arithmetic coder with the intention of improving the computational performance of context adaptive coding. To achieve this, the quantized transform residuals as well as other non-binary syntax elements are binarized to form binstrings. H.264 CABAC offers four basic binarization codes, which can be combined to form binstrings. For example, the absolute level of non-zero quantized coefficients (NZs) is coded by a concatenation of truncated unary code and 0th order Exp-Golomb code (EG0). However, only the output of some of the basic codes is suitable for encryption, namely those codes that do not vary the length of a binstring, as after arithmetic coding these codes will not lead to an increase in bitrate. The codewords output after entropy coding of the binstrings must also be valid if format compliance at the decoder is to be maintained. Which of the codes is employed in binarization is dependent on the underlying probability distribution of the elements it is applied to. For example, the fixed length code (one of the four basic coders) is suitable for input with a near Uniform probability distribution. While restricting the selection of binstrings of those elements that preserve the bitrate and maintain format compliance, it is also advisable to maintain the average percentage of bits that can be selectively encrypted. However, HEVC CABAC adds another binarization code to the four supplied with H.264, namely the truncated Rice code which is more suitable for the distribution of HEVC residuals. Even then, one of the binstring's output as a truncated Rice code with static context-p is not suitable for encryption with the result that the number of different binstring types that can be encrypted (the encryption space) is no longer a power of two, i.e. is non-dyadic. The conversion method of [17] allows the encryption space to be converted back to being dyadic. Once the encryption space for the truncated Rice codes with context-p is converted into dyadic form the data can then be encrypted by AES in CFB mode. (CFB mode is normally employed to use the block encryption method of AES.)

2.4. Other approaches to transparent encryption

In [34] the BL at a lowered quality acts as the preview layer along with some of the ELs. However, simply encrypting the remaining ELs through the AES means that the stream is not decoder format compliant. As [34] also concedes, the implication is a new file format for the encrypted ELs. For single-layer HEVC, the authors of [35] flip the sign bits of luminance transform coefficients (TCs) up to a given percentage of such bits. One issue that is reported is that the number of non-zero TCs varies between intra-coded frames and inter-coded frames, implying that for high QP (low quality) as few as three bits per frame are altered in the preview version. This appears to make the method vulnerable to a replacement attack of the encrypted bits to recover a higher quality version. A further disadvantage of [35] is mentioned by the authors: at very low QPs, i.e. very high quality, the transform step is more often skipped resulting in a reduction in distortion because there are fewer TC signs to encrypt. Conversely, in our method, encryption of ELs increases the distortion at lower QPs because more detail is encrypted. Also for single-layer HEVC [36] "bundles in", i.e. adds to the method of [35], with the intention of improving the security and increasing distortion. The main difference with earlier methods is encryption of the transform skip bit, which is carried in Picture Parameter Set (PPS) NAL unit packets. However, there is a risk of increasing the bitrate by sending more PPS packets than are needed, especially as in some implementations an additional redundant PPS packet may be transmitted to increase roubustness.

Adaptation transparency is a form of transparency that does not offer perceptual encryption but does allow MANEs to process the bitstream. In [28], parts of the compressed bitstream not required for adaptation transparency are encrypted. However, this entails including an IV within the bitstream whenever encryption takes, place resulting in an overhead of around 8.5 bytes for every NAL so encrypted. In [37], a NAL encryption method for adaptation transparency was also presented. As in [33], an encrypted NAL is signaled by means of an unspecified NAL type, causing a decoder not in decryption mode to simply drop that NAL. The method selects which NALs can be encrypted in this way and also checks that reserved header bytes do not inadvertently appear in the stream after encryption. Unfortunately, an IV is still required, which for the selection of NALs used resulted in a bitrate overhead of up to 3.4%.

3. Proposed schemes

3.1. Encryption method

The CABAC encoder has a good number of parameters or bin strings that can be encrypted, for example (in no particular order): MB types; Coded Block Flag; TCs; MVDs; delta quantization parameters (dQPs); and the numerical signs of TCs and MVDs. The distinction of this research from others is the choice of parameters selected for encryption according to the requirements of maintaining decoder format compliance and the need to not to disturb the statistical characteristics of the final compressed video bitstream so that the bit rate remains unchanged. The former requirement implies that the encryption does not violate H.264/SVC standardization, as it is the bitstream that is standardized. The latter requirement results in no change to the streaming rate, which means that there is no increase in latency, which would impact upon real-time applications of video, especially interactive applications. Thus, we encounter three bin-strings that satisfy the purpose of SE, these are as follows:

- Signs of the MVDs;
- Signs of the NZ-TC levels; and
- Signs of the texture values;

The sign bits of MVDs have two interpretations depending on whether $0 < |MVD| < 9$ or $|MVD| \geq 9$. The sign of the NZ-TC levels and/or 'texture', i.e. quantized TCs, is present when the absolute value of the syntax element is greater than 14.

Because the data in the selected bins, namely the signs of various syntax elements, are uniformly distributed, encryption of the selected bins does not affect the compression ratio. The selection of signs is also decoder format compliant as the bits encrypted may flip their values but do not assume disallowed values and are not encrypted if they are not present. Their encryption also does not alter the arithmetic coder's context models in any way because the data selected for encryption bypasses context modeling. Moreover, the chosen bins impact upon the three scalabilities of SVC video, because in SVC every layer potentially requires changes to the NZ-TC level signs and MVD signs.

3.2. Scheme 1

In Scheme 1, as mentioned in Section 1, XOR encryption technique is exploited for encrypting the H.264/SVC stream rather than employ AES, possibly using CFB mode. The motivation is to improve the speed of computation and to reduce the implementation complexity. The technique simply XORs the sign bit of MVDs, TCs and texture binstrings with a changing secret value. Because we chose to encrypt the CABAC parameters, so encryption is applied to the binstrings and not to the output bitstream. To preserve transparency, the encryption technique is only applied to the SVC ELs. In order to generate a sequence of random values, a pseudo-random number generator [38] is initiated with a seed value. The seed value, in effect is input to the generation of the initial key for the selective encryption of the stream and it is this key which must be securely distributed to the receiver. The next paragraph contains details of the seed, key and random number length, as well as the method of random number generation. Fig. 4 summarizes the XOR technique to produce the encrypted binstrings which form part of the output SVC bitstream when appropriately combined with the other parts of the output from the CABAC coder. Table 1 contains a summary of Scheme 1 and Scheme 2 to aid the following discussions.

In the evaluation of Section 4, a seed of n = 128 bits was used in the pseudo-random number generator. This results in a sequence of length 2^{128} bits before the sequence repeats itself, which should be enough for most selective encryption purposes. The resulting key size, arising after input of the seed, by default in the Yarrow algorithm [38] is 160 bits. The initial seed is not applied directly but is combined with a pool of ‘entropy’ via the SHA-1 cryptographic hash function. (Entropy is formed from prior unpredictable inputs such as computer mouse movements.) It is the key size that defines the block size used in each XOR operation. Moreover, a threshold or generator gate, set at 10 by default, causes the Yarrow key to be reset. Because no new source of entropy is introduced into the key [38], there is no need to send the new key to the receiver. Yarrow uses each key as input to the Triple Data Encryption algorithm, which applies the Data Encryption Standard cipher algorithm to each block of input bits. The block of bits taken from the random number stream in Fig. 4 are XORed with an 160-bit block taken from the concatenation of successive groups of three sign bits, i.e. approximately 53 groups at a time. Recall from Section 3.1 that the input sign bits are Uniformly distributed and, hence, the concatenation of these bits is itself Uniformly distributed. In other words, the input before encryption is already randomized, though not encrypted. The XOR encryption cannot be broken by trivial mathematical means if the seed is not reused, which is the case in the Yarrow random number generator, but it is obviously not as secure as (say) AES encryption. As the intention in this paper is to optimize computational speed rather than security, the XOR method matches our needs. It should be noted that other choices of random number generator can also be substituted for the one used in the tests, such as the later Fortuna algorithm from the same researchers as in [38], the main difference from the Yarrow algorithm being the method of initial entropy generation.

3.3. Scheme 2

As mentioned in Section 1, selective encryption of only I- and P-frames is possible [24] in order bring performance benefits such as reduced bitrate overhead and rapid computation. In [39] there is an analysis of a system for H.264/AVC I-frame only SE which also brings low computation, low bitrate overhead and decoder format compliance after encryption, though no analysis of the video distortion or image structural distortion arising from partial encryption was made in [39]. As another example employing CABAC-based encryption in [40] only I- and P-frames are selectively encrypted. The underlying motivation behind such schemes is that even if the bitstream is captured by packet ‘sniffer’ software or stored by some means at the receiver device successful decoding of the P- or B-frame compression data depends on being able to decode the corresponding I-frame within a Group of Pictures (GOP) [41]. Therefore, in [39] P- and B-frame data are sent in the clear. However, if it is possible to only encrypt B-frames, as in our experimental scheme, then the computational and bitrate overhead is much reduced. This is because, as these frames are computationally efficient by reason of bi-prediction, their data are much reduced in comparison to I- and P-frames, in fact approximately only one the size of I-frames in earlier single-layer codecs [41].

In scheme 2, we present the effect of employing the same SE method as in scheme 1, but only encrypting B-frames. As mentioned in Section 1, the encryption treatment of I- and P-frames is left open, though clearly if their compressed data is sent in the clear the possibility of recreating the selectively encrypted B-frames exists, as B-frames are predicted from I- and P-frames. As an example of that treatment, full encryption could be applied to I- and P-frames while still making savings from not fully encrypting the whole of the stream. Or, if the possibility of de-streaming by means of a stream recorder or stream ‘ripper’ software could be discounted then the I- and P-frame parts of the compressed bitstream could be sent in the clear. Unfortunately, there are many de-streaming software programs available such as ‘Download Studio’ or ‘Orbit Downloader’. If the receiver streaming platform is not controlled by the user for example if it is a set-top box then it may be possible to relax encryption of I- and P-frames. The possibility of stream-casting by capture from a display screen (the so-called analog hole) and re-compressing the stream is unattractive if there is partial encryption of the B-frames.

4. Evaluation

4.1. Scheme 1

In this Section, Scheme 1 is evaluated in which the BL is not encrypted and ELs are selectively encrypted, thus resulting in transparent selective encryption (TSE). The video configuration is summarized in Table 2. We used the reference implementation of H.264/SVC, which is Joint Scalable Video Model (JSVM) 9.18 in SVC mode. Common Intermediate Format (CIF) (352 × 288 pixels/frame) was employed in the interests of speeding up testing. Three hundred

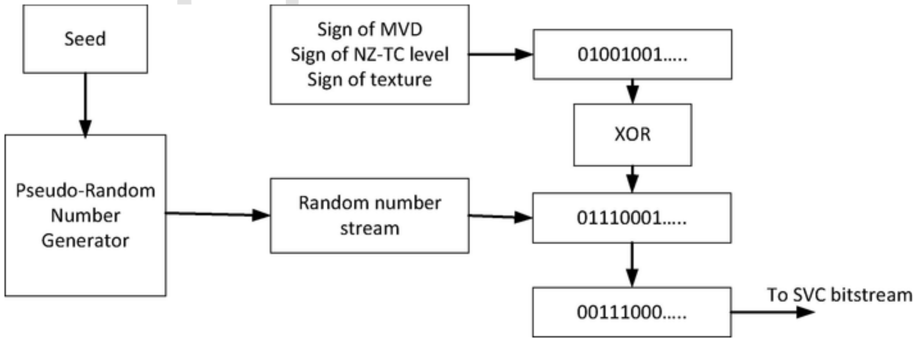


Fig. 4. Encryption of selected syntax elements using XOR method.

Table 1
Summary of schemes 1 and 2. As Scheme 2 only differs in one way from scheme 1, its difference is only recorded.

Scheme 1	Value
Codec	H.264/SVC
Encryption stage	CABAC entropy coding
Encrypted parameters	MVD signs, NZ-TC level signs, signs of TCs
Frame types applied to	I, P, B
Part of SVC applied to	ELs
Encryption method	XOR
Pseudo-Random Number Generator (PRNG)	Yarrow [38]
Seed size of PRNG	128 bits
Block size of PRNG	160 bits
PRNG sequence length	2^{128} bits
Scheme 2	**Value**
Frame types applied to	B

Table 2
Summary of video configuration used in both scheme 1 and 2.

Setting	Parameter
Codec implementation	JSVM 9.18
Format	CIF
Frame rate	30
Group-of-Pictures	IBBP….
Refresh period	15
Chroma sampling	4:2:0
Number of SNR layers	4
Encoding method	VBR
QPs tested	8, 24, 48

frames of the well-known Foreman sequence were chosen for encryption. The sequence was configured as CIF @ 30 Hz, with standard 4:2:0 sampling and a variable bit-rate (VBR). The frame format was IBBP… that is a periodic intra-coded frame every 15 frames, with intermediate bi-predicted B-frames and one–way predicted P-frames.

A BL (layer 0) and the three ELs (layers 1–3) were employed. Fig. 5(a), (c) and (e) show a sample frame from the sequence without encryption for the VBR video, while Fig. 5(b), (d) and (f) demonstrate the visual impact of SE upon the same frame when reconstructing all four layers including a transparent BL (labeled as Scheme-1 TSE in Fig. 5). Objective video distortion is reported in decibels (dB) for Peak Signal to Noise Ratio (PSNR) [42] for the YUV signals (compared to the uncompressed video) and structural distortion through the Structural SIMilarity (SSIM) index [43], which is intended to better capture Quality of Experience (QoE) than PSNR, with a real-valued score ranging from 0 to 1. Notice that the effect of combining the transparent base-layer with encrypted ELs, in Fig. 5 and later illustrations, is a tendency to leave some planar areas of a frame relatively untouched. This is because SNR ELs concentrate on higher spatial frequency detail. It is this detail in the ELs that is separately selectively encrypted. Despite the presence of base layer material in the reconstructed frames, it seems unlikely that a viewer would pay to view the distorted four-layer versions of the frames. For example, the expression of the Foreman cannot be seen in Fig. 5(b), even though this is the main part of the semantic content in the frame shown. Equally, a home viewer is unlikely to want to use up access network bandwidth, for which there may be a fee, to basically watch the background in Fig. 5(b).

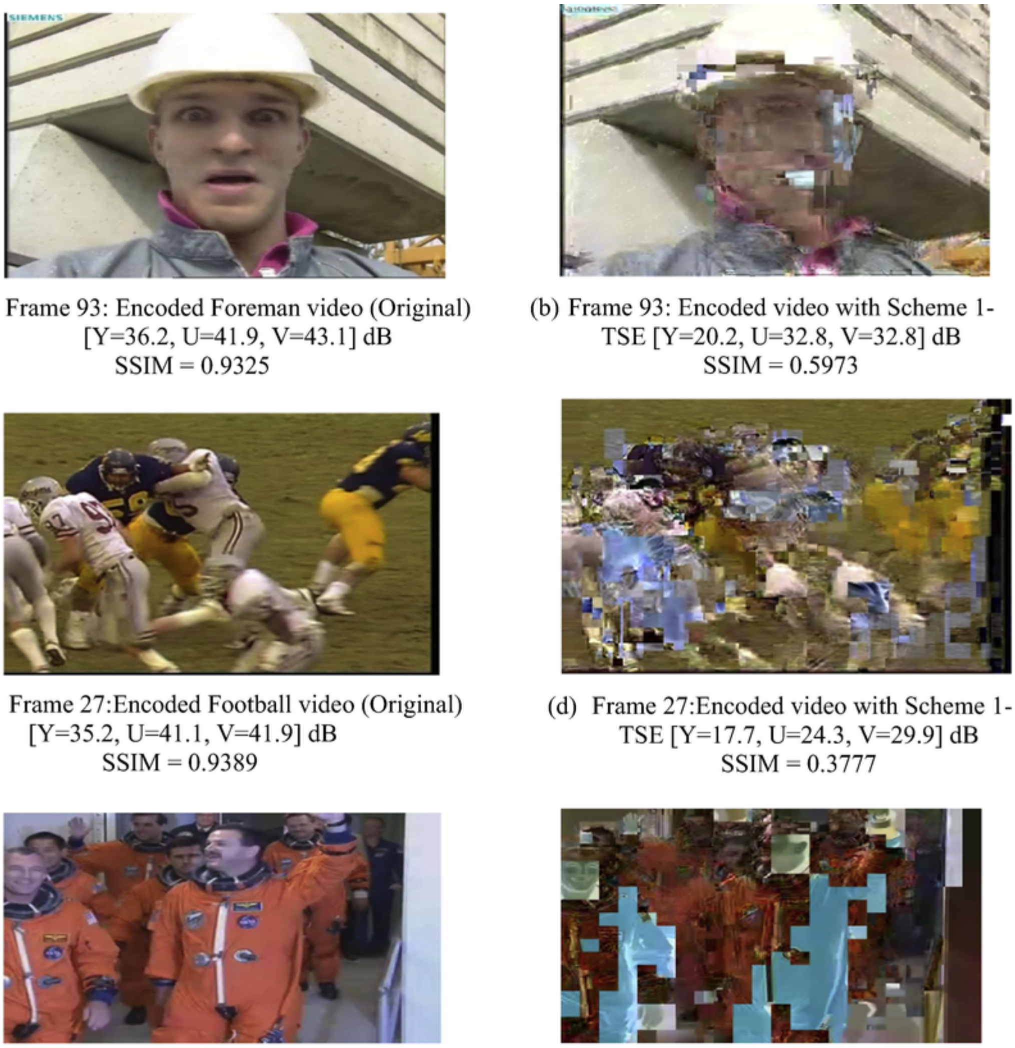
The encoding timings for seven reference video sequences with and without Scheme 1 TSE, were assessed. From Fig. 6, across seven reference video sequences VBR encoded, the additional encoding delay from applying SE was found to be on average 21.6 ms over each sequence, which is a small though noticeable delay. (The delay was evaluated by simply subtracting the encoding times without and with SE.) The selection of seven sequences is intended to show any content dependency as reflected in the encoding complexity.

As a point of comparison, the PSNR and SSIM of the seven sequences were evaluated by luminance (Y) and the two color components (U and V). Though luminance is generally thought to be most important in visual recognition, the color components can enable recognition. In Tables 3 and 4, 'plain' refers to the unencrypted version of the video sequence, while 'SE' refers to the Scheme 1 of Fig. 6. From Table 3, PSNR luminance is seriously degraded after TSE in all videos except perhaps 'City' and 'Crew', which have 'poor' quality close to unwatchable. The impact of TSE is less severe upon the chrominance components but as most information results from the luminance signal, this is not such a weakness if the luminance signal is distorted. In fact, SSIM is helpful in showing the overall poor visual appearance that results from applying TSE, for example when comparing 'Crew' and 'City' with 'Foreman'.

The video distortion and structural distortion resulting from the transparent scheme were also assessed at various other configurations of the QP. Recall that the QP setting controls the coarseness of quantization [41], with the range of H.264's QP being 0–51 [44], with higher values representing lower video distortion. Again four layers were encoded by H.264/SVC but the base layer was left unencrypted. By way of visual comparison, Fig. 7 shows the visual quality after reconstructing all four layers of the reference 'Foreman', 'Football' and 'Crew' sequences after employing transparent SE (TSE). An interesting feature of this comparison is that the very high quality version of QP = 8 appears to suffer more visual distortion as a result of applying SE to the ELs, which would otherwise contribute more to the quality of the visual appearance. Though this form of SE has comparatively little impact upon the chrominance, the degradation of the luminance is significant and is reflected in the SSIM score. Because there is greater detail in higher quality video, it is expected that the encrypted ELs will introduce more distortion into a recombined video frame. In fact, the lower quality, higher QP, example frames are largely included to show in a comprehensive manner the distortion across the quality range. Thus, as QP = 48 is near the bottom of the available quality range for H.264/SVC, it is highly unlikely that anyone would pay to view QP = 48 video, as it is of too low a quality.

Fig. 8 compares the encoding times (i.e. video encoding with encryption) from applying Scheme 1 to the trial sequences. When the QP value is reduced and, thus, the video quality is improved, the amount of data to encode and encrypt increases. Poor-quality video at QP = 48 takes less time to encode and encrypt compared to encoding at QP = 24 or very high quality at QP = 8. Taking QP = 24 at near broadcast quality the extra latency introduced by including Scheme 1 encryption is no more than 40 ms.

Tables 5 and 6 make PSNR and SSIM comparisons for the set of QP configurations, after applying Scheme 1 to the four-layer SVC video sequences. At interesting feature of these results is that both for PSNR and SSIM the video or structural distortion respectively is not consistent with the QP. This is especially the case for QP = 8 when comparing the luminance (Y) PSNR values with those at the other two QPs or the overall SSIM values. The reason for this is the same as mentioned for Fig. 7 and elsewhere, namely that as more detail is held in the ELs of lower QP (higher quality) video, encrypting of the ELs results in greater distortion, thus improving the relative quality of encrypted higher QP (lower quality) video. Thus if improved video or structural distortion is set by configured by changing the QP this does not result in a better SE experience for the viewer. This is a beneficial feature, as otherwise Scheme 1 would be quality dependent



(a) Frame 93: Encoded Foreman video (Original) [Y=36.2, U=41.9, V=43.1] dB SSIM = 0.9325

(b) Frame 93: Encoded video with Scheme 1-TSE [Y=20.2, U=32.8, V=32.8] dB SSIM = 0.5973

(c) Frame 27:Encoded Football video (Original) [Y=35.2, U=41.1, V=41.9] dB SSIM = 0.9389

(d) Frame 27:Encoded video with Scheme 1-TSE [Y=17.7, U=24.3, V=29.9] dB SSIM = 0.3777

(e) Frame 157: Encoded Crew video (Original) [Y=36.2, U=41.2, V=39.7] dB SSIM = 0.9034

(f) Frame 157: Encoded video with Scheme 1-TSE [Y=19.3, U=29.3, V=26.3] dB SSIM= 0.3274

Fig. 5. Impact on video distortion and structural distortion of Scheme 1 SE at QP = 28.

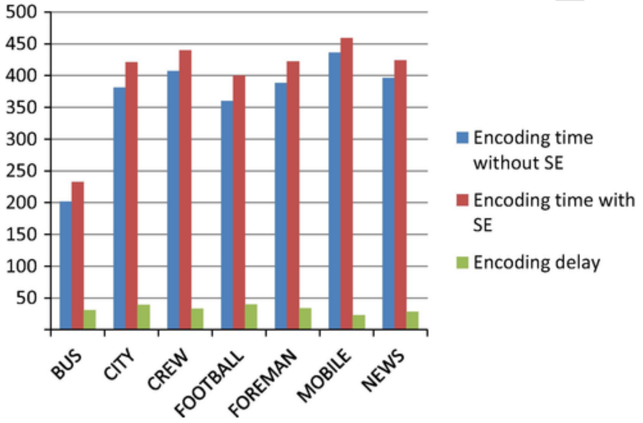


Fig. 6. Impact on encoding delay (ms) of Scheme 1-TSE at QP = 24.

Table 3
Objective video distortion (PSNR) (dB) comparing ‘plain’ (unencrypted) and Scheme 1-TSE. Four layers, one transparent BL and three encrypted ELs, were encoded and then decoded to form the resulting comparison sequence with the raw YUV input sequence.

Video:	Plain PSNR(Y)	SE PSNR(Y)	Plain PSNR(U)	SE PSNR(U)	Plain PSNR(V)	SE PSNR(V)
Bus	33.8127	7.5672	41.4443	27.1731	42.6340	28.8414
City	35.6983	21.7449	43.6323	37.4172	44.6468	39.0928
Crew	36.0810	19.2596	41.2232	29.2787	39.7137	26.3325
Football	35.0203	10.9995	40.9714	16.0459	41.8172	22.9100
Foreman	36.0996	9.8684	41.9013	25.2492	42.9718	24.4196
Mobile	33.1513	8.5743	37.4355	15.0468	36.3129	13.0113
News	38.7243	8.5743	42.6260	15.0468	42.6783	13.0113

and reducing distortion would increase the risk of content extraction in some way. As in Table 3, in Table 5 the chrominance distortion is small but again, as Table 6 confirms, the overall impact of Scheme 1 is large, implying that the chrominance impact is relatively small.

Table 4
Structural distortion (SSIM) comparing ‘plain’ (unencrypted) and Scheme 1-TSE. Four layers, one transparent BL and three encrypted ELs were encoded and then decoded in order to form the comparison sequence with the raw YUV input sequence.

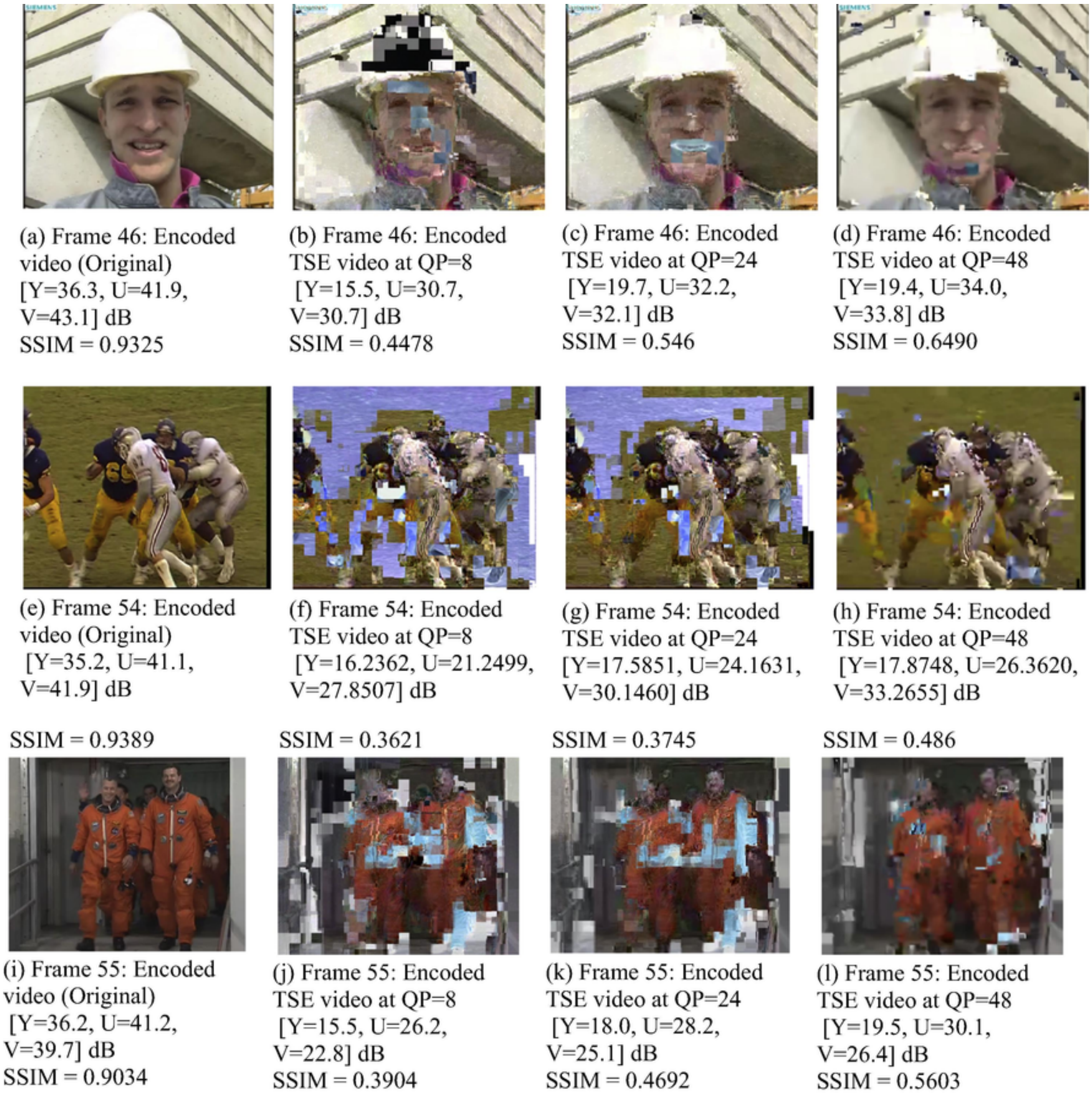
Video:	Plain SSIM	SE SSIM
Bus	0.9505	0.2121
City	0.9399	0.3569
Crew	0.9034	0.3274
Football	0.9389	0.3777
Foreman	0.9325	0.5973
Mobile	0.9573	0.3698
News	0.9630	0.4800

4.2. Scheme 2

In Scheme 2 reduced selective encryption (RSE) is applied in which computation time is reduced as much as possible, by transparently encrypting B-frames only according to the GOP configuration of Scheme 1. In other respects, the video configuration is the same as described for Scheme 1. Fig. 9 shows Scheme 2 comparisons.

Fig. 10 shows that as a result computation time is reduced compared to Scheme 1, being on average reduced to 17.6 ms compared to 21.6 ms for Scheme 1. Tables 7 and 8 make the equivalent comparison of video and structural distortion for Scheme 2, as for Scheme 1. Because in the test run there was no encryption of I- and P-frames, the distortions are numerically much less than for Scheme 1 in respect to PSNR luminance. The SSIM shows that structural distortion is dependent on content. For example, the structural distortion is numerically much less for ‘Bus’ and ‘News’ than it is for the other reference video sequences. Therefore, any benefits of employing this reduced encryption scheme need to be carefully considered in view of the sometimes limited reduction in video quality.

Fig. 11 shows the visual quality after reconstructing all four layers of the reference ‘Foreman’, ‘Football’ and ‘Crew’ video sequences after employing Scheme 2. Comparing Fig. 9 with Fig. 11, for QP = 24 but different frames, it can be seen that introducing encrypted ELs, leads to differing distortion for differing frames. As far as a viewer is concerned it is the combined impact on their QoE that



(a) Frame 46: Encoded video (Original) [Y=36.3, U=41.9, V=43.1] dB SSIM = 0.9325

(b) Frame 46: Encoded TSE video at QP=8 [Y=15.5, U=30.7, V=30.7] dB SSIM = 0.4478

(c) Frame 46: Encoded TSE video at QP=24 [Y=19.7, U=32.2, V=32.1] dB SSIM = 0.546

(d) Frame 46: Encoded TSE video at QP=48 [Y=19.4, U=34.0, V=33.8] dB SSIM = 0.6490

(e) Frame 54: Encoded video (Original) [Y=35.2, U=41.1, V=41.9] dB SSIM = 0.9389

(f) Frame 54: Encoded TSE video at QP=8 [Y=16.2362, U=21.2499, V=27.8507] dB SSIM = 0.3621

(g) Frame 54: Encoded TSE video at QP=24 [Y=17.5851, U=24.1631, V=30.1460] dB SSIM = 0.3745

(h) Frame 54: Encoded TSE video at QP=48 [Y=17.8748, U=26.3620, V=33.2655] dB SSIM = 0.486

(i) Frame 55: Encoded video (Original) [Y=36.2, U=41.2, V=39.7] dB SSIM = 0.9034

(j) Frame 55: Encoded TSE video at QP=8 [Y=15.5, U=26.2, V=22.8] dB SSIM = 0.3904

(k) Frame 55: Encoded TSE video at QP=24 [Y=18.0, U=28.2, V=25.1] dB SSIM = 0.4692

(l) Frame 55: Encoded TSE video at QP=48 [Y=19.5, U=30.1, V=26.4] dB SSIM = 0.5603

Fig. 7. Impact on video distortion and structural distortion of Scheme 1 i.e. SE with transparency.

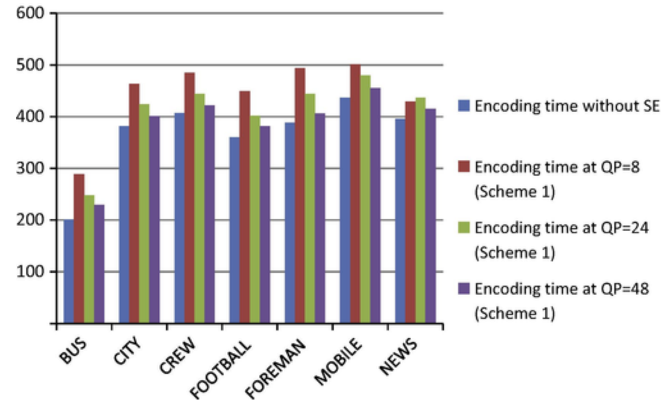


Fig. 8. Impact on encoding delay (ms) of Scheme 1 by QP compared to encoding without SE at QP = 24.

Table 5
Objective video distortion (PSNR) (dB) for Scheme 1, SE with transparency at various QP. Four layers, one BL and three ELs, were encoded and then decoded to form the comparison sequence with the raw YUV input sequence.

Video:	PSNR at QP = 8	PSNR at QP = 24	PSNR at QP = 48

Bus	Y = 14.7065,
	U = 32.1326,
	V = 33.5629
		Y = 15.4475,
		U = 32.8441,
		V = 34.3368
			Y = 14.1103,
			U = 34.1708,
			V = 36.3819

City	Y = 21.0067,
	U = 36.0487,
	V = 37.6778
		Y = 21.6377,
		U = 37.0483,
		V = 38.7481
			Y = 21.9428,
			U = 41.0066,
			V = 42.2481

Crew	Y = 15.4863,
	U = 26.2483,
	V = 22.8910
		Y = 18.0550,
		U = 28.1828,
		V = 25.0462
			Y = 19.5161,
			U = 30.0286,
			V = 26.3879

Football	Y = 16.2362,
	U = 21.2499,
	V = 27.8507
		Y = 17.5851,
		U = 24.1631,
		V = 30.1460
			Y = 17.8748,
			U = 26.3620,
			V = 33.2655

Foreman	Y = 15.4781,
	U = 30.6830,
	V = 30.6885
		Y = 19.7152,
		U = 32.1339,
		V = 32.1232
			Y = 19.3802,
			U = 34.0413,
			V = 33.8755

Mobile	Y = 16.0873,
	U = 24.0464,
	V = 22.4842
		Y = 16.4235,
		U = 24.6036,
		V = 23.2733
			Y = 13.9330,
			U = 23.2713,
			V = 21.7817

News	Y = 14.5198,
	U = 22.5164,
	V = 26.2998
		Y = 15.4924,
		U = 22.7608,
		V = 26.5983
			Y = 10.6712,
			U = 21.3146,
			V = 24.8221

Table 6
Structural distortion (SSIM) comparing ‘plain’ (unencrypted) and SE Scheme 1 at various QP. Four layers, one transparent BL and three encrypted ELs were encoded and then decoded in order to form the comparison sequence with the raw YUV input sequence.

Video:	Plain SSIM	SSIM at QP = 8	SSIM at QP = 24	SSIM at QP = 48
Bus	0.9505	0.1796	0.2039	0.4635
City	0.9399	0.3293	0.3490	0.4109
Crew	0.9034	0.3904	0.4692	0.5603
Football	0.9389	0.3621	0.3745	0.4860
Foreman	0.9325	0.4478	0.5460	0.6490
Mobile	0.9573	0.3206	0.3547	0.4017
News	0.9630	0.4303	0.4782	0.5456

will be affected, not whether some more details are visible in one particular frame rather than another.

For consistency with Scheme 1, Fig. 12 illustrates delays from encrypting at various QP configurations. Again, encoding time with Scheme 2 SE at QP = 8 results in the most delay, whereas at the low quality setting of QP = 48, the impact of SE is small.

4.3. Comparison of both schemes

From the evaluation of both schemes it is concluded that the scheme 2-RSE is more efficient compared to scheme 1-TSE in terms of encoding times. Fig. 13 compares the results of both schemes and shows that the impact of RSE on the distortion of a video sequence is ‘sufficient’ to make it useful when streaming videos in real-time. That is to say, when comparing the same frames, encrypted either by Scheme 1-TSE or Scheme 2-RSE, the distortion appears similar to the viewer.

Fig. 14 shows that for Scheme 2 encoding time is reduced to a minimum. This makes Scheme 2 especially appropriate to real-time video streaming. For example, it could be applied to video conferencing, when encoding delay has a critical impact.

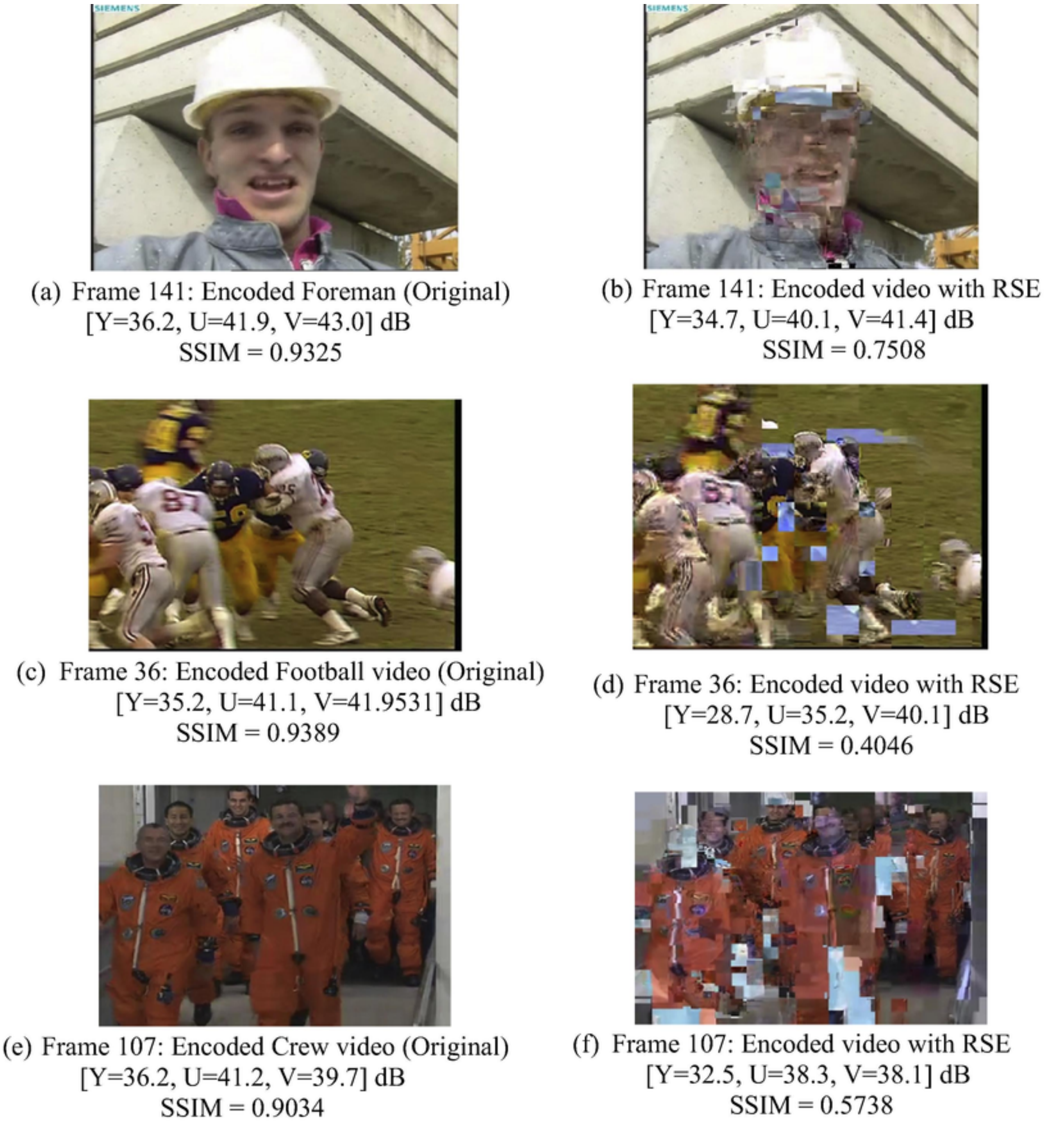
4.4. Security analysis

In the two schemes presented, no existing standard method of encryption has been employed. Therefore, it might be claimed that the security is low, and, in particular, sign manipulation may be vulnerable to a guessing attack. This is because an attacker has only two values available for the signs of non-zero TCs and MVDs. That is to say, the sign can be positive or negative and no other value. Hence, it might be claimed that it is easy to guess the values of the changed signs and, thus, make the video watchable. In addressing this issue, consider Table 2 of [25]. In [25] it is demonstrated that there are literally millions of MVDs and TCs in some of the video sequences listed in Table 9 of the current paper, which video sequences also appear in Table 2 of [25]. The probability of guessing the signs [25] can be found from the standard formula for a combination:

$${}^{a}C_{b} = \binom{a}{b} = \frac{a(a-1)\ldots(a-b+1)}{b!\,(a-b)!}$$

where for example, a denotes the number of non-zero MVDs and b denotes the number of guesses. Because the number of signs that would need to be guessed successfully is so large, the probability of guessing all the signs is very low. In [25] it is demonstrated that if the *Bus* sequence is taken as an example, the number of ways of guessing MVDs signs with complete accuracy is 2^119904. In [25] it is also shown that even if 80% of the MVD signs were guessed successfully the *Bus* video would still not be in a watchable condition. Therefore, the proposed sign changing method is sufficiently secure without applying a standard encryption method such as AES.

The encryption method discussed in Section 3.1 and used in this research is a simple but secure encryption method. The first likely form of attack is to guess is the signs of the MVDs and/or the TCs and from the above sample calculation it is demonstrated that the schemes are secure. The second likely form of attack is to guess the random number that is XORed with those signs. Each random number is one in a sequence of numbers calculated from the seed used to initiate the random sequence. However, the seed and its length are changed for each video sequence. Without the seed, which is distributed over another secure back-channel, it becomes very difficult to guess the random numbers that result. Hence our encryption method will be sufficiently secure for the two proposed methods. Fig. 15 shows a sample frame resulting from a test in which the seed was guessed and the resulting random number sequence was applied to decrypt the sequence. From Fig. 15, it is apparent that after guessing the seed, which acts as a key to the random number sequence, the video quality might actually deteriorate rather than improve. Overall,



(a) Frame 141: Encoded Foreman (Original)
[Y=36.2, U=41.9, V=43.0] dB
SSIM = 0.9325

(b) Frame 141: Encoded video with RSE
[Y=34.7, U=40.1, V=41.4] dB
SSIM = 0.7508

(c) Frame 36: Encoded Football video (Original)
[Y=35.2, U=41.1, V=41.9531] dB
SSIM = 0.9389

(d) Frame 36: Encoded video with RSE
[Y=28.7, U=35.2, V=40.1] dB
SSIM = 0.4046

(e) Frame 107: Encoded Crew video (Original)
[Y=36.2, U=41.2, V=39.7] dB
SSIM = 0.9034

(f) Frame 107: Encoded video with RSE
[Y=32.5, U=38.3, V=38.1] dB
SSIM = 0.5738

Fig. 9. Impact on video distortion and structural distortion of Scheme 2-RSE at QP = 24.

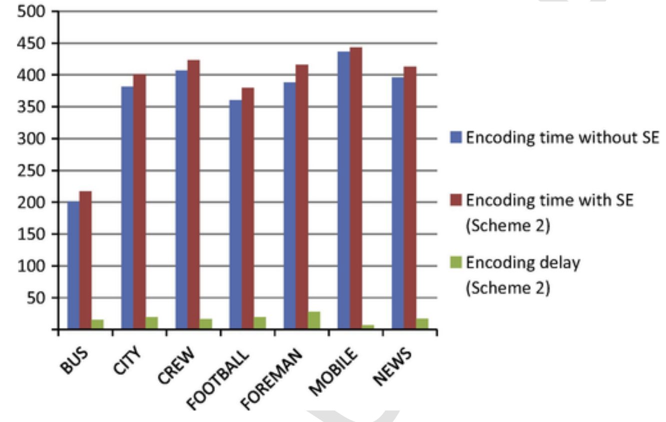


Fig. 10. Impact on delay of Scheme 2 'B-frame SE' i.e. SE with transparency only operating on B-frames.

Table 7
Objective video distortion (PSNR) (dB) comparing 'plain' (unencrypted) and Scheme 2 SE. Four layers, one transparent BL and three encrypted ELs, were encoded and then decoded to form the resulting comparison sequence with the raw YUV input sequence.

Video:	Plain PSNR(Y)	SE PSNR(Y)	Plain PSNR(U)	SE PSNR(U)	Plain PSNR(V)	SE PSNR(V)
Foreman	36.2649	34.7012	41.9294	40.3657	43.0567	41.4930
City	35.6985	34.4405	43.6319	41.4848	44.6468	42.9246
Football	35.1999	28.7438	41.0895	35.2479	41.9531	40.0931
Mobile	33.3150	30.3358	37.4350	34.6155	36.3101	33.8428
News	38.7365	35.0797	42.6317	40.3666	42.6866	41.0666
Bus	34.0392	29.5499	41.5835	39.5326	42.8261	40.1849
Crew	36.1534	32.4864	41.2410	38.3539	39.7444	38.0666

Table 8
Structural distortion (SSIM) comparing ‘plain’ (unencrypted) and RSE Scheme 2. Four layers, one transparent BL and three encrypted ELs were encoded and then decoded in order to form the comparison sequence with the raw YUV input sequence.

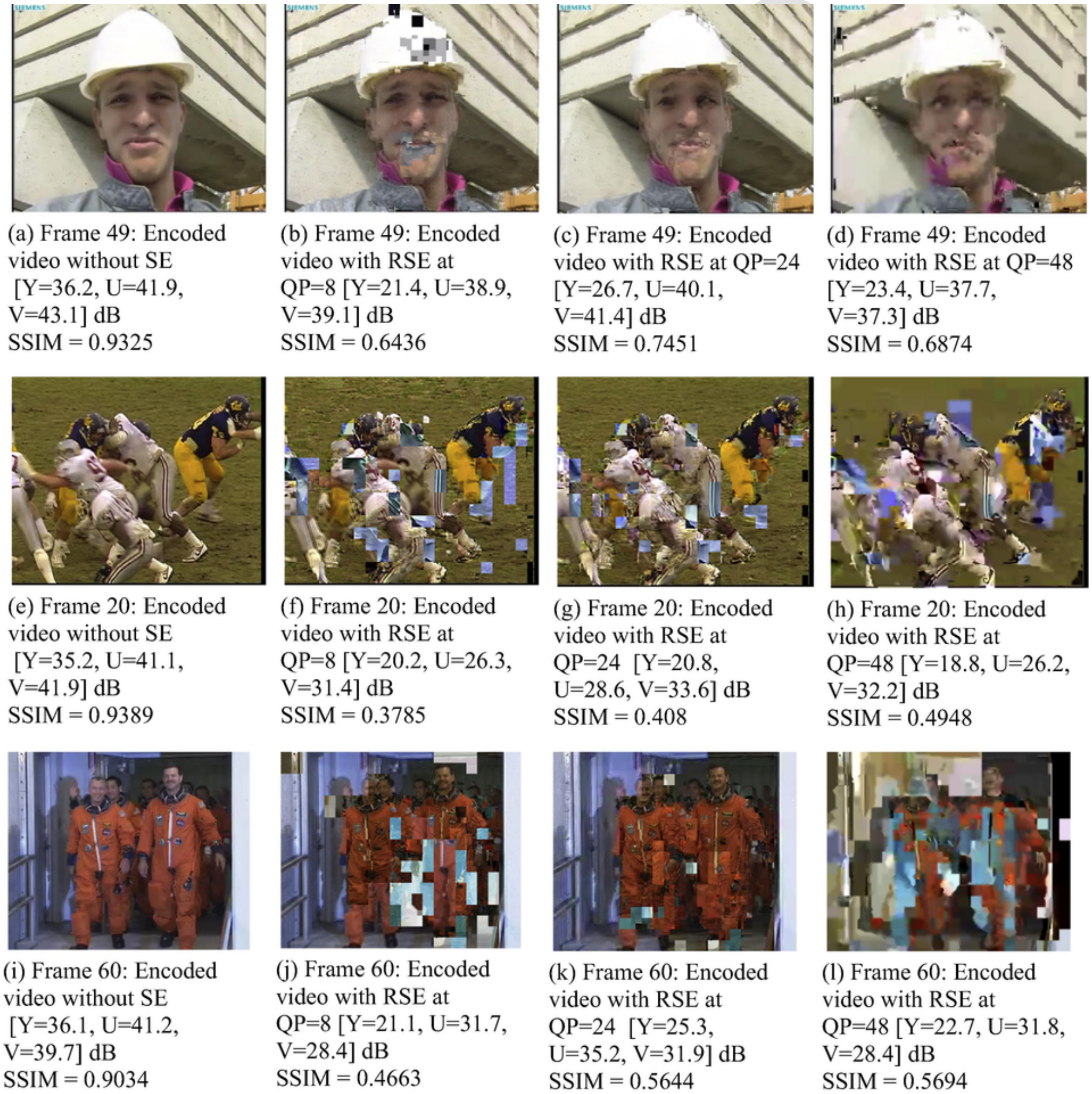
Video:	Plain SSIM	SE SSIM
Bus	0.9505	0.8712
City	0.9399	0.3748
Crew	0.9034	0.5738
Football	0.9389	0.4046
Foreman	0.9325	0.7508
Mobile	0.9573	0.3830
News	0.963	0.8921

the simplified encryption method is sufficiently secure against the most likely forms of attack.

4.5. Comparative analysis

As a comparison of the two proposed transparent SE schemes with prior work, a selection of CABAC-based SE methods have been used. The parameters chosen as a means of comparison are as follows:

P_1**: Selected encryption items:** This specifies the items upon which encryption is based.
P_2**: Compression efficiency:** This describes the compression overhead.
P_3**: Format compliancy:** The encrypted bit streams are compatible to the SVC requirements and also consistent with the standard SVC decoder, if this parameter is fulfilled.
P_4**: Friendly bandwidth utilization:** Implies that there is no bitrate overhead if this parameter is fulfilled.
P_5**: Computational complexity:** This parameter specifies the computational time required to encrypt an SVC video. If the encoding time is low it means that the computational complexity is also low and vice versa if the encoding time is high.
P_6**: Results with different QP values:** This parameter is used to check video statistics i.e. a higher QP gives lower quality.
P_7**: Encryption domain:** Whether SE is applied.
P_8**: Level of security:** This parameter describes the proposed schemes from different authors and shows to what extent that they are secure.



(a) Frame 49: Encoded video without SE [Y=36.2, U=41.9, V=43.1] dB SSIM = 0.9325

(b) Frame 49: Encoded video with RSE at QP=8 [Y=21.4, U=38.9, V=39.1] dB SSIM = 0.6436

(c) Frame 49: Encoded video with RSE at QP=24 [Y=26.7, U=40.1, V=41.4] dB SSIM = 0.7451

(d) Frame 49: Encoded video with RSE at QP=48 [Y=23.4, U=37.7, V=37.3] dB SSIM = 0.6874

(e) Frame 20: Encoded video without SE [Y=35.2, U=41.1, V=41.9] dB SSIM = 0.9389

(f) Frame 20: Encoded video with RSE at QP=8 [Y=20.2, U=26.3, V=31.4] dB SSIM = 0.3785

(g) Frame 20: Encoded video with RSE at QP=24 [Y=20.8, U=28.6, V=33.6] dB SSIM = 0.408

(h) Frame 20: Encoded video with RSE at QP=48 [Y=18.8, U=26.2, V=32.2] dB SSIM = 0.4948

(i) Frame 60: Encoded video without SE [Y=36.1, U=41.2, V=39.7] dB SSIM = 0.9034

(j) Frame 60: Encoded video with RSE at QP=8 [Y=21.1, U=31.7, V=28.4] dB SSIM = 0.4663

(k) Frame 60: Encoded video with RSE at QP=24 [Y=25.3, U=35.2, V=31.9] dB SSIM = 0.5644

(l) Frame 60: Encoded video with RSE at QP=48 [Y=22.7, U=31.8, V=28.4] dB SSIM = 0.5694

Fig. 11. Impact on video distortion and structural distortion of Scheme 2 i.e. RSE with transparency.

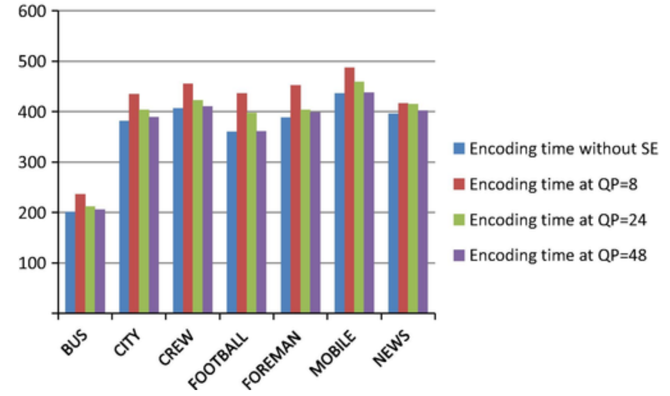


Fig. 12. Impact on encoding delay (ms) of Scheme 2 by QP compared to encoding without RSE at QP = 24.

$P_{9:}$ **Encryption applied to frames:** SVC has three main types of frames I, P & B. This parameter specifies whether encryption is applied to all frame types or specific frame types**.**

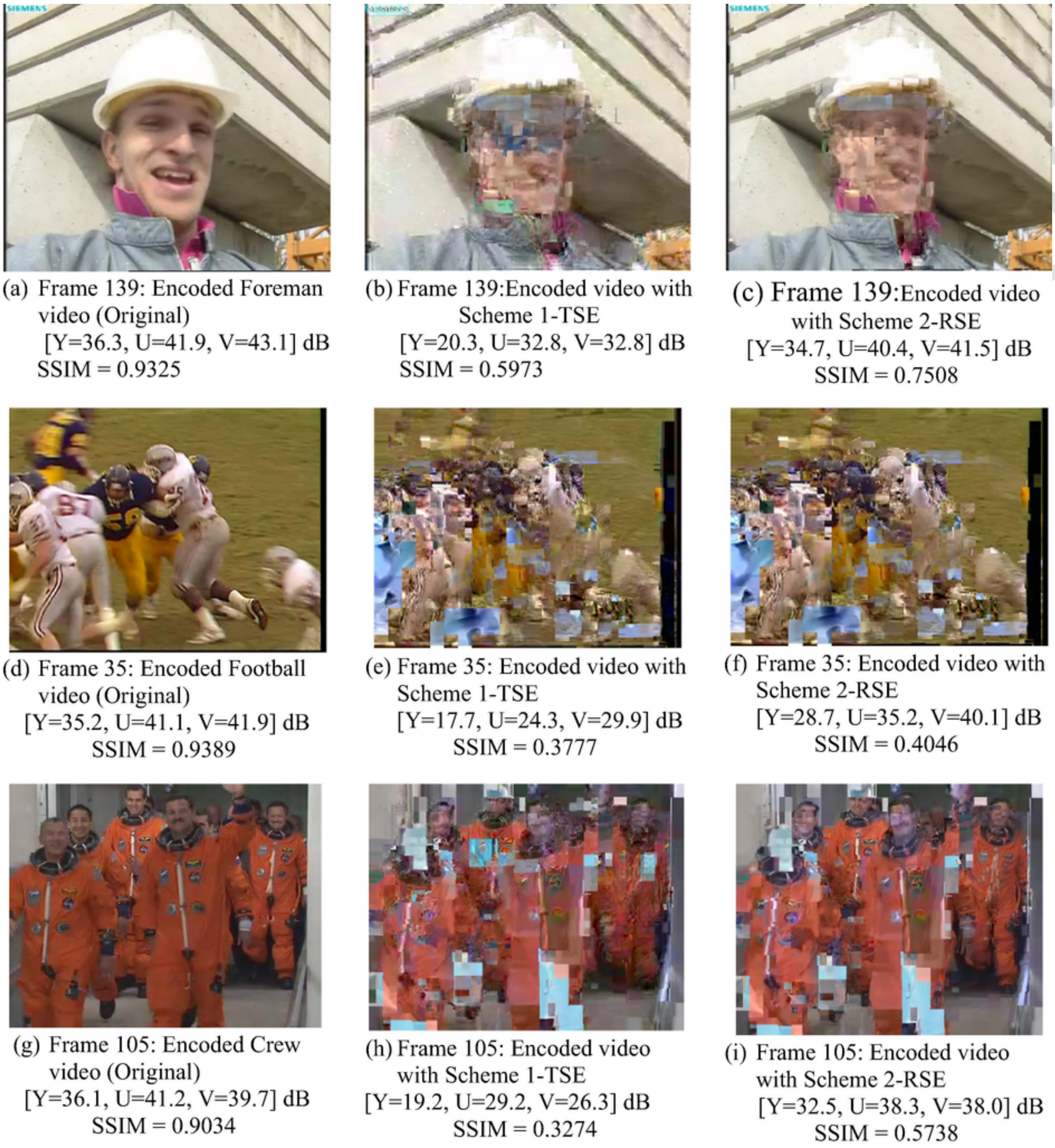
P_{10}**: Efficiency (in terms of time)** This parameter specifies how quickly the video will be encoded and decoded.

In Table 10, all the comparisons are based on encryption at the CABAC entropy coder stage of encoding. The encryption method proposed by Align and Tanali [45] relies on the alteration of the DC values, which effectively alters the video statistics before further compression is applied. Hence it causes some bit-rate overhead and consequently is less efficient. The proposal of [46] also incurs a bit-rate overhead. The computational cost of the schemes described in both [24,40] is high because they used AES-based encryption of the selected parameters. Thus, the comparison of Table 10 shows that Scheme 1 is efficient with low computational cost and no bandwidth overhead. The proposed Scheme 2 in comparison with Scheme 1 and others' work is highly efficient with minimal computational cost.

Overall, the two proposed schemes incorporate a simple XOR encryption algorithm for applying SE on SVC bin-strings. Comparisons show that the selected parameters of H.264/SVC that were used in the two schemes are most effective in their computational performance. On the other hand, recent methods of encryption have some drawbacks in terms computational complexity, bitrate overhead, and efficiency and may be implemented by choosing weak encryption parameters.

5. Conclusion

This paper examines two transparent encryption schemes applied to scalable video. Scalable video delivery emphasizes flexibility over optimal compression and in that sense is similar to transparent encryption that emphasizes commercial utility over complete content confidentiality. Underlying the form of transparent encryption employed herein is selective encryption, which also introduces a compromise in terms of video distortion. Furthermore, the paper pushes this flexibility one step further and asks whether reduced encryption, as it is called herein, can be used in a further compromise between protection and transparency by emphasizing reduction in encryption delay. In respect to the latter, it is also possible that I- and P-frames could be fully encrypted or completely selectively encrypted, while transparently encrypting B-frames in the manner illustrated. Therefore, the evaluations in this paper show the trade-offs and compromises possible and their resulting impacts. Reducing delay will be significant for real-time delivery of video streams, such as for sports video streaming, when a provider will not want their event to appear slightly later on a screen than a rival's in a neighboring building, especially when (say) a goal is celebrated. On the other hand, a provider will want their encryption-free video stream to appear superior to an encrypted version but may only require the one to appear distorted and not completely hidden, for example so that the position of a sport's ball is unclear but the sport's field is recognizable if still distorted. Further work will continue to investigate these encryption compromises in order to determine in a robust manner what reduction in delay and distortion is definitely achievable by what techniques and for what types of video content. As the HEVC codec is specialized to high-resolution video and as increasingly video is being watched on lightweight and/or mobile devices, encrypted commercial video should be adapted for this type of target device.



(a) Frame 139: Encoded Foreman video (Original)
[Y=36.3, U=41.9, V=43.1] dB
SSIM = 0.9325

(b) Frame 139:Encoded video with Scheme 1-TSE
[Y=20.3, U=32.8, V=32.8] dB
SSIM = 0.5973

(c) Frame 139:Encoded video with Scheme 2-RSE
[Y=34.7, U=40.4, V=41.5] dB
SSIM = 0.7508

(d) Frame 35: Encoded Football video (Original)
[Y=35.2, U=41.1, V=41.9] dB
SSIM = 0.9389

(e) Frame 35: Encoded video with Scheme 1-TSE
[Y=17.7, U=24.3, V=29.9] dB
SSIM = 0.3777

(f) Frame 35: Encoded video with Scheme 2-RSE
[Y=28.7, U=35.2, V=40.1] dB
SSIM = 0.4046

(g) Frame 105: Encoded Crew video (Original)
[Y=36.1, U=41.2, V=39.7] dB
SSIM = 0.9034

(h) Frame 105: Encoded video with Scheme 1-TSE
[Y=19.2, U=29.2, V=26.3] dB
SSIM = 0.3274

(i) Frame 105: Encoded video with Scheme 2-RSE
[Y=32.5, U=38.3, V=38.0] dB
SSIM = 0.5738

Fig. 13. Comparison of video frames encrypted with Scheme 1-TSE and alternatively with Scheme 2-RSE.

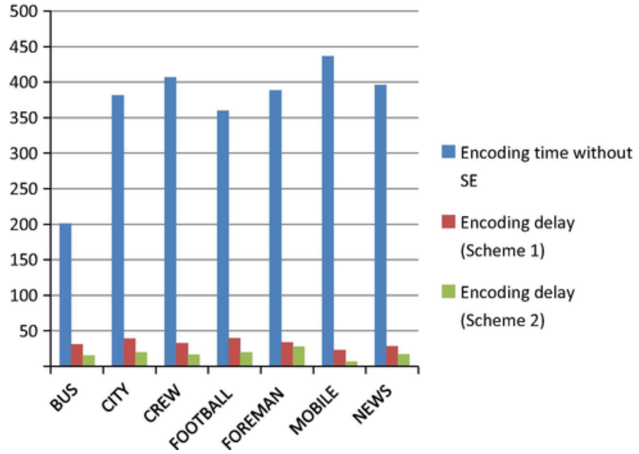
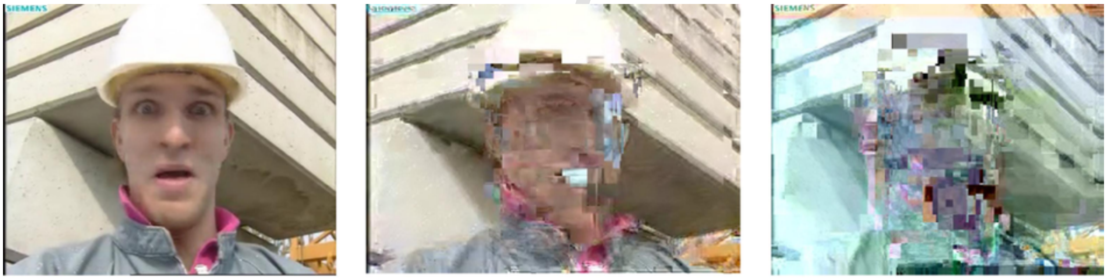


Fig. 14. Impact on encoding delay (ms) comparing Scheme 1 with Scheme 2 in relation to total encoding time (ms) without SE at QP = 24.

Table 9
Number of signs of MVDs and of TCs in test sequences.

Test Videos	No. of frames	No. of MVD signs	No. of TC signs = (Suffixes + Signs of NZ-TC)
Bus	150	120,526	2,325,724 = 24,343 + 2,301,381
City	300	110,899	1,938,693 = 9740 + 1,928,953
Crew	300	195,575	3,294,709 = 5415 + 3,289,294
Foreman	300	132,151	2,196,765 = 8898 + 2,187,867
Football	260	194,587	4,259,181 = 24,688 + 4,234,493
Mobile	300	196,352	6,160,677 = 102,930 + 6,057,747
News	300	54,459	1,102,597 = 19,675 + 1,082,922



Foreman video without encryption, frame (93) [Y=36.2649, U=41.9294, V=43.0567] dB SSIM = 0.9325

Foreman with Scheme 1-TSE frame (93) [Y=20.2793, U=32.8021, V=32.8775] dB SSIM = 0.5973

Decrypted *Foreman* video after guessing the seed for the random number sequence, frame (93) [Y=15.0246, U=25.5100, V=22.5412] dB SSIM = 0.4052

Fig. 15. Example of the effect of guessing the seed for Scheme 1-TSE.

Table 10
Comparison of proposed schemes with other CABAC-based SE methods (ROI = Region of Interest).

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Align & Tunali [45]	Alteration of DC, TC and MVD signs	NO	YES	YES	Low	NO	NULL	Low	NULL	Low
Park & Shin [46]	IPM, residual signs and MVD signs	NO	YES	YES	Low	No	ROI	High	NULL	High
Asghar & Ghanbari [24]	UEG3 suffix, UEG0 suffix, and signs of TC levels	Yes	Yes	No	High	Yes	Binstrings	High	I, P & B frames	High
Shahid et al. [40]	I & P frames encryption	No	Yes	No	High	Yes	Bitstream	High	I & P frames	High
Proposed Scheme 1	**Signs of: TC levels, MVDs, and TCs**	**Yes**	**Yes**	**No**	**Low**	**Yes**	**Binstrings**	**High**	**I, P & B frames**	**High**
Proposed Scheme 2	**Signs of: TC levels, MVDs, and TCs, for B frames only**	**Yes**	**Yes**	**No**	**Very Low**	**Yes**	**Binstrings**	**High**	**B frames**	**Very High**

References

[1] J.-R. Ohm, Advances in scalable video coding, Proc. IEEE 93 (1) (2005) 42–56.

[2] T. Wiegand, J.G. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H. 264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.

[3] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007) 1103–1120.

[4] G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668.

[5] P. Helle, H. Lakshman, H. Siekmann, J. Stegemann, et al., A scalable video coding extension of HEVC, In: Data Compression Conf., 2013, pp. 201–210.

[6] D. Hong, J. Wonkap, J. Boyce, A. Abbas, Scalability support in HEVC, In: IEEE Int'l Symp. on Circuits Syst., 2012, pp. 890–893.

[7] Z. Shahid, W. Puech, Visual protection of HEVC video by selective encryption of CABAC binstrings, IEEE Trans. Multimedia 16 (1) (2014) 24–36.

[8] R. Civanlar, A. Eleftheriadis, O. Shapiro, System and method for a conference server architecture for low delay and distributed conferencing applications, U.S. patent 7,593,032, 2009.

[9] J. Lotspiech, Digital rights management for consumer devices, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, Boca Raton, FL, 2004, pp. 691–714.

[10] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, J.J. Quisquater, Overview on selective encryption of image and video: challenges and perspective, EURASIP J. Inf. Security 2008 (2008) 1–18, article no. 5.

[11] S. Li, G. Chen, A. Cheung, B. Bhargava, K.-T. Lo, On the design of perceptual MPEG-video encryption algorithms, IEEE Trans. Circuits Syst. Video Technol. 17 (2) (2007) 1–10.

[12] N. Thomas, D. Bull, D. Redmill, A novel H.264 SVC encryption scheme for secure bit-rate transcoding, in: Picture Coding Symposium, 2009, pp. 1–4.

[13] B. Furht, D. Socek, A.M. Eskicioglu, Fundamentals of multimedia encryption techniques, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, Boca Raton, FL, 2002, pp. 95–132.

[14] D. Kundur, K. Karthik, Video fingerprinting and encryption principles for digital rights management, Proc. IEEE 92 (6) (2004) 918–932.

[15] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing routines for stateless receivers, Adv. Cryptol. (2001) 41–75.

[16] R.H. Deng, X. Ding, Y. Wu, Z. Wei, Efficient block-based transparent encryption of H.264/SVC bitstreams, Multimedia Syst. 20 (2) (2014) 165–178.

[17] Z. Shahid, M. Chaumont, W. Puech, Selective and scalable encryption of enhancement layers for dyadic scalable H.264/AVC by scrambling of scan patterns, In: IEEE Int'l Conf. on Image Proc., 2009, pp. 1273–1276.

[18] M. Podesser, H.-P. Schmidt, A. Uhl, Selective bitplane encryption of secure transmission of image data in mobile environments, in: 6th Nordic Signal Processing Symposium, 2002.

[19] M.N. Asghar, M. Ghanbari, Cryptographic keys management for H.264 scalable coded video security, in: 8th Int'l ISC Conf. on Info. Security and Cryptology, 2011, pp. 83–86.

[20] M.N. Asghar, M. Fleury, M. Ghanbari, Key management protocols for secure wireless multimedia services: a review, Recent Pat. Telecommun. 1 (1) (2012) 41–53.

[21] I.-L. Kim, J. Min, T. Lee, W.-J. Han, J. Park, Block partitioning structure in the HEVC standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1697–1706.

[22] M.N. Asghar, M. Ghanbari, M.J. Reed, Sufficient encryption with codewords and bin-strings of H.264/SVC, In: IEEE 11th Int'l Conf. on Trust, Security and Privacy in Computing and Communs, 2012, pp. 443–450.

[23] M.N. Asghar, M. Ghanbari, M. Fleury, M.J. Reed, Efficient selective encryption with H.264/SVC CABAC bin-strings, In: IEEE Int'l Conf. on Image Processing, 2012, pp. 2645–2648.

[24] M.N. Asghar, M. Ghanbari, An efficient security system for CABAC bin-strings of H.264/SVC, IEEE Trans. Circuits Syst. Video Technol. 23 (3) (2013) 425–437.

[25] M.N. Asghar, M. Ghanbari, M. Fleury, M.J. Reed, Confidentiality of a selectively encrypted H.264 coded video bit-stream, J. Visual Commun. Image Representation 25 (2) (2014) 487–498.

[26] M.N. Asghar, M. Ghanbari, M. Fleury, M.J. Reed, Analysis of channel error upon selectively encrypted H.264 video, in: 4th IEEE CEEC Int'l Conf., 2012, pp. 139–144.

[27] F. López, J.M. Martinez, V. Valdes, Multimedia content adaptation within the CAIN framework via constraints satisfaction and optimization, in: 4th Int'l Workshop on Adaptive Multimedia Retrieval, 2006, pp. 149–163.

[28] H. Kadikara Arachchi, X. Perramon, X. Dogan, A.M. Kondoz, Adaptation aware encryption of scalable H.264/AVC video for content security, Signal Process.: Image Commun. 24 (6) (2009) 468–483.

[29] Z. Wei, Y. Wu, X. Ding, R.H. Deng, A scalable and format-compliant encryption scheme for H.264/SVC bitstreams, Signal Process.: Image Commun. 27 (9) (2012) 1011–1024.

[30] T. Stütz, A. Uhl, Format-compliant encryption of H.264/AVC and SVC, in: Tenth IEEE International Symp. on Multimedia, 2008, pp. 446–451.

[31] T.C. Chen, Y.W. Huang, C.Y. Tsai, B.Y. Hsieh, L.G. Chen, Architecture design of context-based adaptive variable-length coding for H. 264/AVC, IEEE Trans. Circuits Syst. II Express Briefs 53 (9) (2006) 832–836.

[32] G. Sullivan, P. Topiwala, A. Luthra, The H.264/AVC Advanced Video Coding standard: overview and introduction to the fidelity range extensions, in: SPIE Conf. on Applications of Digital Image Processing, vol. XXVII, 2004, pp. 454–474.

[33] D. Marpe, H. Schwarz, T. Wiegand, Context-based adaptive binary arithmetic coding in the H.264 video compression standard, IEEE Trans. Circuits Syst. Video Technol. 13 (4) (2003) 620–636.

[34] E. Magli, M. Grangetto, G. Oglio, Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video, Signal Process. 91 (5) (2011) 1103–1114.

[35] H. Hofbauer, A. Uhl, A. Unterweger, Transparent encryption for HEVC using bit-stream selective coefficient sign extraction, in: IEEE Int'l Conf. on Acoustics, Speech and Signal Proc, 2014.

[36] Y. Tew, K. Minemura, K. Wong, HEVC selective encryption using transform skip signal and sign bin, in: Asia-Pacific Signal and Info. Process. Assoc. Ann. Summit and Conf., 2015, pp. 963–970.

[37] H. Hellwagner, T. Stütz, R. Kuschnig, A. Uhl, Efficient in network adaptation of encrypted H.264/SVC content", Signal Process.: Image Commun. 24 (9) (2009) 740–758.

[38] J. Kelsey, B. Schneier, N. Ferguson, Yarrow-160: Notes on the design and analysis of the yarrow cryptographic pseudorandom number generator, in: Sixth Ann. Workshop on Selected Areas in Cryptography, 1999, pp. 13–33.

[39] M. Abombara, O. Zakaria, O.O. Khalifa, A.A. Zaiden, B.B. Zaiden, Enhancing selective encryption for H.264/AVC using Advanced Encryption Standard, Int. J. Comput. Theory Eng. 2 (2) (2010) 282–289.

[40] Z. Shahid, M. Chaumont, W. Puech, Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I & P frames, IEEE Trans. Circuits Syst. Video Technol. 21 (5) (2010) 565–576.

[41] M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, IEE, Stevenage, UK, 2003.

[42] Q. Huynh-Thu, M. Ghanbari, The accuracy of PSNR in predicting video quality for different scenes and frame rates, Telecommun. Syst. 49 (1) (2012) 35–48.

[43] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[44] I. Richardson, The H.264 Advanced Video Compression Standard, second ed., J. Wiley & Sons, Chichester, UK, 2010.

[45] G.B. Algin, E.T. Tunali, Scalable video encryption of H.264 SVC codec, J. Visual Commun. Image Representation 22 (4) (2011) 353–364.

[46] S.W. Park, S.U. Shin, An efficient encryption and key management scheme for layered access control of H.264/scalable video coding, IEICE Trans. Inf. Syst. 92 (5) (2009) 851–858.