EFFICIENT SELECTIVE ENCRYPTION WITH H.264/SVC CABAC BIN-STRINGS

Mamoona Naveed Asghar, Mohammed Ghanbari, Martin Fleury, and Martin Reed

University of Essex, Colchester, United Kingdom.

ABSTRACT

The distribution of copyrighted scalable video content to differing digital devices should be protected during rendering and transmission. The proposed scheme is applied to H.264 Scalable Video Coding (SVC) CABAC bin strings in a compression-friendly and decoder format compliant manner. It achieves this by careful selection of the entropy coder syntax elements for selective encryption (SE) with respect to SVC. Tests show that: decoding delay is small, replacement and key substitution attacks are fruitless; there is no increase in bitrate; and the stream remains format compliant. The proposed SE scheme is extremely suitable for video distribution to users who have subscribed to differing video qualities on medium- to high-computationally capable digital devices.

Index Terms— AES-CFB, CABAC, H.264/SVC, selective encryption

1. INTRODUCTION

This paper investigates some important issues in the protection of scalable video distribution and proposes an efficient encryption system for the H.264/Scalable Video Coding (SVC) codec [1]. H.264/SVC permits the transmission and decoding of partial bit-streams to provide video services at various temporal, spatial, and quality resolutions, as well as preserving a reconstruction quality that is comparable to the rate of the partial bit-streams.

As encryption will alter the data characteristics, it should be applied where it has a minimal side effect. That is it should be applied to entropy coding, where all the natural redundancies have already been exploited for maximum compression efficiency. However, entropy coding still needs to be handled with great care, since tampering with the statistical dependency of the symbols, also harms the compression efficiency. We propose a selective encryption (SE) scheme on H.264 scalable layers through the Advanced Encryption Standard in a cipher feedback mode (AES-CFB). SE is applied to the Context Adaptive Binary Arithmetic Coding (CABAC) bin strings [2] in a compression-friendly and format-compliant manner. The reason for choosing CABAC over its Huffman counterpart, Context Adaptive Variable Length Coding (CAVLC), is because of the greater range of parameters for encryption that CABAC provides over CAVLC. The H.264 Main Profile and the various High Profiles, which deal with higher resolution pictures (4CIF and above), support CABAC. Thus, it appears that the multi-scale video distribution of the future will support CABAC. Currently, the outlook is for full VGA resolution on standard streaming mobile applications (e.g. Apple's FaceTime), full 720p high definition (HD) on mobile devices and full 1080p HD for desktop streaming, which will require the reduction in bitrate supported by CABAC.

In the recent past, some work has been performed on the SE of H.264/SVC layers. In [3], after compression H.264/AVC and SVC Network Abstraction Layer (NAL) units were individually encrypted, in order to have no side-effect on format compliancy. By setting the NAL unit type of encrypted NALs to be outside the defined range, the decoder is forced to reject those NALs, unless decryption is enabled. The low-quality SVC base layer is not encrypted but users must subscribe to higher enhancement layers if decryption is enabled. In [4], a low-quality free preview application was developed by performing transparent encryption on the H.264/SVC layers, resulting in non-format compliant enhancement layers. The algorithm encrypts the scalable enhancement layers, while leaving the base layer in plain format. On the other hand, there is a conviction that if the base layer is protected then no one can get the data from the enhancement layers and the whole SVC bit-stream is secured. Although it is a useful technique to reduce the computational cost, research shows [5] that if objects are encrypted in this way their content can be easily guessed without decryption. Consequently, Algin et al. [6] proposed the idea of SE on SVC with three security levels. The idea of [6] involves the encryption of signs of coefficients, signs of motion vectors and the alteration of DC values. Sign encryption has no effect on the bitrate and compression efficiency (as the signs are equally likely) but DC value alterations must change the video statistics and affect the compression efficiency, which consequently increases the bitrate. That is to say the bitrate overhead increases, while the video quality remains the same. Another promising technique applied in [7] to non-scalable H.264/AVC also uses SE on the codewords/bin-strings of the entropy coder. This has the advantage of preserving the bitrate, being standard compliant and involving negligible overhead.

The remainder of this paper is organized as follows. Section 2, details the choice of syntax components for SVC. Section 3 outlines the validation tests that have been performed, while Section 4 makes some concluding remarks.

2. SELECTIVE ENCRYPTION FOR SVC

After briefly outlining CABAC encoding, this Section describes the principles that led to the selection of the syntax elements for SE. The elements are selected firstly in general terms for CABAC within H.264 and then in terms of those elements most suitable for the SVC extension to H.264.

2.1. CABAC operation

CABAC encoding is based on three steps: 1) Binarization; 2) Context Modeling; and 3) Binary Arithmetic Coding (BAC). In binarization, any input non-binary syntax elements, such as the quantized transform coefficients, macroblock type specifiers, and motion vector components are converted into unique binary codewords known as bin strings for a given syntax element. There are five basic code trees: the *unary code*; the *truncated unary code (TU)*; the *kth order Exponential-Golomb code (EGk)*; the *fixed length (FL) code*; and the *concatenation of the first and third schemes (UEGk).* Which code (or combination of codes) is applied depends on the characteristics of the syntax element. The bit position in each bin string is known as a bin. Each bin is then passed to one of the two coding decision modes: regular coding mode and by-pass coding mode. The bins in regular coding mode are passed to the next step, context modeling/probability distribution and then encoded by the regular BAC engine. The bins from the bypass coding mode skip the context modeling step and directly enter the bypass BAC engine for the encoding process.

2.2. Selection of SE components

The CABAC coder has multiple parameters (bin strings) which can be encrypted, for example: transform coefficients (TC); motion vector differences (MVD); delta quantization parameters (dQP); and the arithmetical signs of TC and MVD. To make the SE more effective one needs to sensibly choose the parameters for the encryption. There are two concerns in parameter selection:

1) *Compression friendliness* specified that the SE must not disturb the compression efficiency of the encoder by increasing the bitrate of video file. This consequently increases the encrypted data size to transfer on a given bandwidth. It can be controlled by keeping the size of an encrypted bin string (codeword) the same as the size of the input bin string, and also by keeping the context model unchanged for the given syntax element.
2) *Format compliance* means the SE must not change the overall video statistics, which change would otherwise make the SVC decoder complain about decoding the selectively encrypted bitstream.

To fulfill the above constraints one can make some recommendations for SE. Some can be made on the basis of experimental results, while others are in respect to the nature of the syntax elements. The SE should *not* apply to:

- the intra-coded syntax elements having a relationship with neighboring MBs syntax elements such as Intra DC and AC, as this would elevate the bitrate and cause variation in the values of syntax elements. Consequently, the bitstream would not be decodable at some stage.
- the inter-coded syntax elements such as MVDs as, by changing their magnitudes, the bitrate would be increased.
- the delta QP syntax element, as this would also cause bitrate fluctuations, either increasing or decreasing the overall bitrate according to new encrypted dQP values.
- the macroblock (MB) header information (encoded first in the CABAC encoding), because this is used for the prediction of future MBs.
- the Coded-Block-Flag (CBF) to make the bitstream format compliant. Every 4×4 block within a MB is encoded if CBP and MBmode are set for it. The encoded 4×4 block has a CBF syntax element showing the non-zero coefficients existing in the current block.
- the Unary and truncated unary (TU) bin strings; they have different codeword lengths and would cause a change of bitrate.
- The FL bins, because they have the mandatory header information.

The chosen bin strings must hold the above two conditions (1 & 2). SE can be applied to the bin strings that are uniformly distributed, as doing this does not change the compression efficiency of the codec and as they are encoded by the bypass BAC engine for fast encoding and with the assumption of uniform probability. We found three bin-strings to fulfill our purpose of selective encryption, these being:

i) UEG3 suffix;
ii) UEG0 suffix; and
iii) Signs of the transform coefficient levels.

The UEG3 suffix consists of the MVD sign bits if two conditions hold i.e. $|MVD| \geq 9$ and $0 < |MVD| < 9$. The sign bits of the TC levels and the suffix of UEG0 can be encrypted only when abs_level > 14. The selected bins are fully compression friendly and format compliant, and have no issues in terms of altering context models.

2.3. Bin selection for SVC layers

The selected bins must be compliant with all three scalabilities in SVC-coded video, so the bin selection is also performed by keeping in mind the specific scalability type as well. In SVC, every temporal layer requires changes to the MVDs, the dQP, and the coefficients. The spatial layers require changes to the coefficients only and the SNR layers require changes to the dQP and the coefficients. Such SVC layer behavior shows that the UEG0 suffix and the sign of the coefficient levels are the most suitable parameters for the SVC layers encryption, because the coefficients change with every scalability option. UEG3 suffix encryption is more suitable for temporal scalability but is meaningful for all

three scalabilities, as spatial and SNR scalability is usually combined with temporal scalability.

2.4. *SE on Bins by AES-CFB*

The SE is implemented on an individual SVC coded layer by using AES-CFB, a stream cipher; hence, it does not alter the number of output bits and also maintains the self-synchronization between transmitter and receiver.

The CFB uses an initialization vector (IV) (fixed in our implementation) and an encryption key (variable) for each data block. The single encryption key is used to encrypt all data blocks of each layer, i.e. one encryption key will encrypt all three chosen bin strings on individual layers. Therefore, the client will receive only one encryption key to decrypt and watch the subscribed data. Suppose that the three chosen bin strings are taken as $P1B1$, $P2B2$, $P3B3$ and their encryption represent as $C1B1$, $C2B2$, $C3B3$ then the general cipher process can be represented as:

$$\sum (C1B1, C_2B_2, C_3B_3) = \{\sum (P_1B_1, P_2B_2, P_3B_3)\} \; XOR \; \{Encrypt\,(C_1B_1\text{-}1, C_2B_2\text{-}1, C_1B_1\text{-}1)\} \quad (1)$$

The encryption process (Fig. 1) is performed in a unique way that makes the bitrate consistent. The sign bits encryption is not tricky, as the bins sizes are constant, although handling of UEG0 suffixes bins is different, as the suffixes have variable length codewords. If the sizes of suffix codewords are changed after encryption, the bitrate will definitely increase. Thus, to make the codewords compression friendly, we first count the suffix bins present in each UEG0 bin string and then do the encryption only on the number of existing suffix bins, rather than the whole suffix allocated size. This scheme makes the encrypted codewords of the same size as the original one.

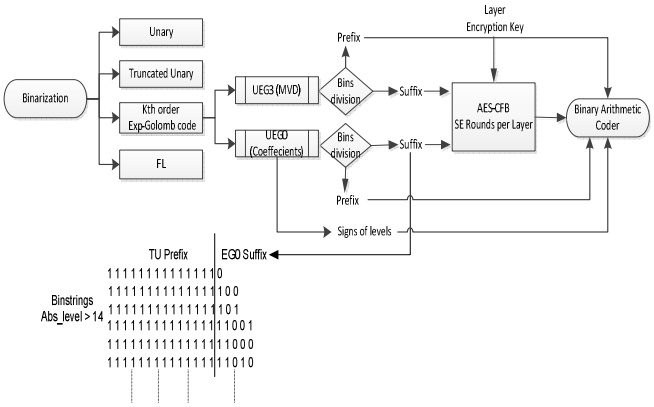

Fig. 1. Block diagram of SE over CABAC bins

The decryption process is the reverse on the CABAC decoding side. The client uses the same supplied encryption key (as used for ciphering) and the encrypted values of bin strings are converted into the original bin values and then passed to the inverse binarization, quantization and transfrom processes to get the finally de-ciphered and fully decoded bit-stream. The general de-cipher process can be represented as:

$$\sum (P1B1, P2B2, P3B3) = \{Encrypt\,(C1B1\text{-}1, C2B2\text{-}1, C1B1\text{-}1)\} \; XOR \; \{\sum (C1B1, C2B2, C3B3)\} \quad (2)$$

3. EVALUATION

Experiments were performed over Common Intermediate Format (CIF) resolution video sequences, selected for the variety of their imagery. They were encoded into four layers (one base with three enhancement layers) representing three temporal, two spatial and two SNR scalable levels in an H.264 Main/High profile (for base-layer encoding) and Baseline profile for enhancement layers. 4:2:0 chroma sub-sampling was set. The SVC reference software (Joint Scalable Video Model) JSVM 9.19.10 version encoder was employed. The Intra and Inter frames were selectively encrypted in order of their occurrence in a bit-stream with Group of Pictures (GOP) size 8 and Intra period 16.

3.1. Effect of SE on PSNR

Table 1 compares the average PSNR of 90 frames (Intra with Inter frames) with and without SE to show the suitability of our SE scheme for both Intra and Inter frames. The average PSNR value of the luma component is the lowest of the range for all sequences. We performed experiments at different QP values of 8, 16, 24, 32, 40 and 48 for Intra and Inter frames. Fig. 2 demonstrates that our SE scheme is independent of QP value, and the average PSNR is still in the lowest range across all QP values.

Table 1. Comparison of Average PSNR of 90 frames (I+P+B) at QP=24

Sequences (CIF)	Plain PSNR Y (dB)	SE PSNR Y (dB)	Plain PSNR U (dB)	SE PSNR U (dB)	Plain PSNR V (dB)	SE PSNR V (dB)
CITY	38.6	10.3	45.4	30.0	46.8	31.7
FOOTBALL	38.5	10.6	43.1	20.6	44.0	19.4
FOREMAN	39.6	8.1	44.8	24.5	47.2	26.2
HARBOUR	37.6	7.4	44.7	21.7	45.8	34.7
ICE	42.3	11.5	48.5	29.7	48.9	25.3
MOBILE	37.6	7.3	41.3	18.8	41.1	15.0
NEWS	42.2	11.3	45.1	19.9	46.3	24.3

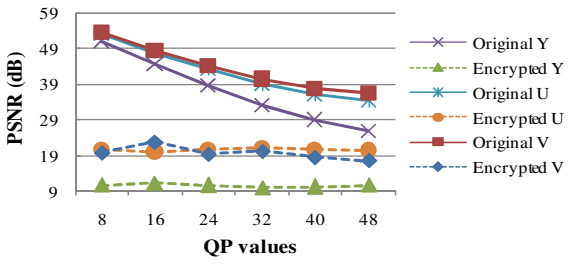

Fig. 2. PSNR variance of the *Football* sequence at a different QP values

3.3. Effect of SE on computation

The computational overhead was calculated on the basis of the additional processing time required for encoding and decoding with SE. The experiments were run on a machine with an Intel Core 2 Duo (3.33GHz) processor with 4 GB of RAM. Table 2 shows the encoding and decoding timings of the *News* sequence for an increasing number of frames with and without SE. The processing delays are negligible in

terms of milliseconds, verifying the efficiency of the proposed scheme on Intra and Inter frames for four-layer SVC, on both encoder and decoder side.

Table 2. The computational overhead measurement (milliseconds) for the *News* sequence at different number of encoded frames (I+P+B) and QP 24

No. of frames	Encoding time without SE	Encoding time with SE	Encoding Delay	Decoding time with SE	Decoding time without SE	Decoding Delay
10	3010.8	3026.4	15.6	571.8	561.6	10.2
30	8751.6	8784.0	32.4	1051.9	1029.6	22.3
50	14726.4	14780.8	54.4	1436.7	1391.2	45.5
70	20389.2	20474.0	84.8	2294.9	2228.0	66.9
90	26642.4	26754.8	112.4	4737.1	2324.4	88.3

3.4. Security analysis

The robustness of proposed SE scheme against various attacks was also evaluated by the following tests.

i) Replacement attacks: We performed experiments on different sequences by replacing encrypted bits of the data with constant bits and determining the PSNR as a result of such a guessing attack. As an illustration for *News*, we replaced the encrypted data with 0's for the MVD signs, 1's for the signs of run levels and added a constant integer value five for the UEG0 and UEG3 suffixes. The resulting picture is distorted, as is apparent from Fig. 3.

(a) (b) (c)

Fig. 3. Impact of replacement attack on the *News* sequence encoded with 90 frames (I+P+B) and QP 24. (a) Frame # 41 [Y=42.29, U=45.15, V=46.33] dB. (b) Proposed SE [Y=11.35, U=19.92, V=24.32] dB. (c) Replacement attack [Y=3.94, U=17.58, V=19.01] dB.

ii) Video perception test with key substitution: Assume one is able to guess the encryption key to a close approximation, i.e. there is a difference of only one or two bits as compared to the exact key value. This test showed that a hacker's attempt to guess the video characters will fail, unless the hacker is able to guess the exact key. Let us assume that the hacker has guessed the exact key (highly unlikely for a 128-bit AES key); even then, the hacker will succeed for a very short time, as every time the same sequence is played, it will be encrypted with a different encryption key. We tried to decrypt with a key with only one or two bits changed. The results, with Fig. 4 as an illustration, confirmed the robustness.

(a) (b) (c)

Fig. 4. Impact of keys on video perception of the *News* sequence encoded with 90 frames (I+P+B) and QP 24. (a) Frame # 41[Y=42.29, U=45.15, V=46.33] dB, (b) *ek* change by 1 bit [Y=11.34, U=19.87, V=24.78] dB, and (c) *ek* change by 2 bits [Y=11.37, U=19.79, V=24.70] dB.

3.5. Brief comparison with other schemes

We have already noted in Section 1 that the most recent scalable work at the time of writing [4] provides the transparent encryption of scalable layers by protecting the enhancement layers in non-format compliant manner. The work in [6] alters the video statistics prior to compression by changing the DC values and the intra prediction modes encryption in [8] [9], results in a potential bitrate overhead.

4. CONCLUSIONS AND FUTURE WORK

The SE is applied on sensibly chosen bin strings by keeping in mind the need for: the confidentiality of video sequences; compression efficiency; bitrate preservation; format compliancy; and scalability features (temporal, SNR and spatial) of H.264/SVC. Prior SE schemes have met some of these objectives but not all of them. The proposed SE can be extended to region-of-interest (ROI) processing for bitrate reduction in video surveillance [10] without any modification. Future work will document the extensive performance analysis of the proposed SE on higher resolution pictures, associated key management system and the error robustness issues [11] during the transmission of scalable layers. Another area worthy of investigation is the semantic security of encrypted video images against indirect 'image estimation' attacks.

5. REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103-1120, 2007.

[2] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circ. Syst. Video Technol*., vol. 17, no. 9, pp. 620-636, 2003.

[3] T. Stűtz, and A. Uhl, "Format-compliant encryption of H. 264/AVC and SVC," in *Proc. IEEE Int'l Symp.on Multimedia*, pp. 446 – 451, 2009.

[4] E. Magli, M. Grangetto, and G. Olmo. "Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video," *J. of Signal Processing*, vol. 91, no. 5, May 2011.

[5] B.B. Zhu, M.D. Swanson, and S. Li, "Encryption and authentication for scalable multimedia: Current state of the art and challenges," in *Proc. SPIE Internet Mult. Management Syst.*, vol. 5601, pp. 157–170, Oct. 2004.

[6] G.B. Algin, and E.T. Tunali, "Scalable video encryption of H.264/AVC codec," *J. of Visual Commun. and Image Representation*, vol. 22, no. 4, pp. 353-364, 2011.

[7] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," *IEEE Trans. Circuits Syst. Video Technol*., vol. 21, no. 5, pp. 565-576, May 2011.

[8] S. Park, and S. Shin. "An efficient encryption and key management scheme for layered access control of H.264/Scalable Video Coding," *IEICE Trans. Inf. and Syst.*, vol. 92, no. 5, pp. 851-858, 2009.

[9] C. Li, C. Yuan, and Y. Zhong, "Layered encryption for scalable video coding," in Proc. Int'l. Congress on Image and Signal Processing, pp. 1–4, Oct. 2009.

[10]Y. Kim, S.H. Jin, T.M. Bae, and Y.M. Ro: "A selective video encryption for the region of interest in scalable video coding," IEEE Region 10 Conference, pp. 1–4, 2007.

[11] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J.-J. Quisquater, "Overview on selective encryption of image and video: Challenges and perspectives," *EURASIP Journal on Information Security*, vol. 2008, Jan. 2008.