Independent Protection of Different Layers in Spatially Scalable Video Coding

Ala Abu-Zahra, Zafar Shahid, Amjad Rattrout, William Puech

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 10 (2012) 240 – 246

Procedia Computer Science

The 3rd *International Conference on Ambient Systems, Networks and Technologies* (ANT 2012)

Independent protection of different layers in spatially scalable video coding

Ala Abu-Zahra[1], Zafar Shahid[2], Amjad Rattrout[3], William Puech[2]

azahra2@science.alquds.edu, zafar.shahid@lirmm.fr, amjad.rattrout@univ-lyon1.fr, william.puech@lirmm.fr

[1]*Al-Quds Univeristy, Abu Dis, Jerusalem*

[2]*LIRMM,UMR CNRS 5506, University of Montpellier II, France*

[3]*University Claude Bernard, Lyon, France*

Abstract

In this paper, a novel method for independent access of different layers of protected scalable video coding (SVC), based on framework of state of the art H.264/AVC, is presented. For a SVC bitstream, a user accessing the base layer needS only one key while users accessing the enhancement layer need keys for both the base layer and enhancement layer. Hence if a user is accessing the highest available layer, keys of all the lower layers are also needed to decode it properly. We have devised a scheme in which the user accessing any particular resolution use only one key whether it is the base layer or any of the enhancement layers, and all the keys have the same length. Advanced encryption standard (AES) algorithm in Cipher Feedback (CFB) mode has been used in this scheme to make the keys secure. The scheme has been tested both for offline playback and online streaming environment. Five different benchmark video sequences having three resolutions and containing different combinations of motion, texture and objects are used for experimental evaluation of the proposed algorithm.

Keywords: scalable video protection, AES, SE-CAVLC, SE-CABAC, selective encryption

1. Introduction

With the evolution of Internet to a heterogeneous network both in terms of processing power and network bandwidth, different users demand the different versions of the same content. This has given birth to the scalable era of multimedia where a single bitstream contains multiple versions of the multimedia content which can be different in terms of resolution, frame rate or quality. As different customers purchase rights for different versions of the same content, the concern about the protection and authentication of SVC bitstreams have surfaced. Encryption can be used to restrict access to only authenticated users for the respective version of the multimedia content. Since video data is huge in amount and multimedia applications have real time constraints, full encryption of multimedia content is avoided. The concept of selective encryption (SE) has been evolved in which only a small part of the whole bitstream is encrypted [1].

Encryption of H.264/AVC has been studied in [2, 3] and encryption of arithmetic coding has been discussed in [4, 5]. But the problem with these techniques is that they make the bitstream non-compliant to the standard.

1877-0509 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.procs.2012.06.033

H.264/AVC supports two types of entropy codings; namely context adaptive variable length coding (CAVLC) and context adaptive binary arithmetic coding (CABAC). Algorithms have been developed to perform SE of both of these modules which fulfills real-time constraints by keeping the bitrate unchanged, generating completely compliant bitstream and utilizing negligible processing power [6, 7]. In this work, we are extending these algorithms to SE of SVC in such a way that independent protection of every resolution is performed. In Section 2, overview of SVC and SE of H.264/AVC is presented. We explain the proposed algorithm in Section 3. Section 4 contains its experimental results and performance evaluation, followed by the concluding remarks in Section 5.

2. Preliminaries

2.1. Scalable video coding

SVC is based on H.264/AVC and uses pyramid architecture. In SVC, the video bitstream contains a base layer and number of enhancement layers. Enhancement layers are added to the base layer to further improve the coded video. The improvement can be made by increasing the spatial resolution, video frame-rate or video quality, corresponding to spatial, temporal and quality/SNR scalability. Previous video standards such as MPEG-2 [8], MPEG-4 [9] and H.263+ [10] also contain the scalable profiles but they were not much appreciated because the quality and scalability came at the cost of coding efficiency. SVC based on H.264/AVC has achieved significant improvements both in terms of coding efficiency and scalability as compared to scalable extensions of prior video coding standards. Similar to the previous scalable video coding standards, SVC is also built upon a predictive and layered approach to scalable video coding.
In spatial scalability, the inter-layer prediction of the enhancement-layer is utilized to remove redundancy across video layers as shown in Fig. 1.a. The resolution of the enhancement layer is either equal or greater than the lower layer. Enhancement layer P images can be predicted either from lower layer or from the previous frame in the same layer. In temporal scalability, the frame rate of enhancement layer is better as compared to the lower layer. This is implemented using I, P and B frame types. In Fig. 1.b, I and P frames constitute the base layer. B frames are predicted from I and P frames and constitute the second layer. In quality/SNR scalability, the temporal and spatial resolution of the video remain the same and only the quality of the coded video is enhanced.

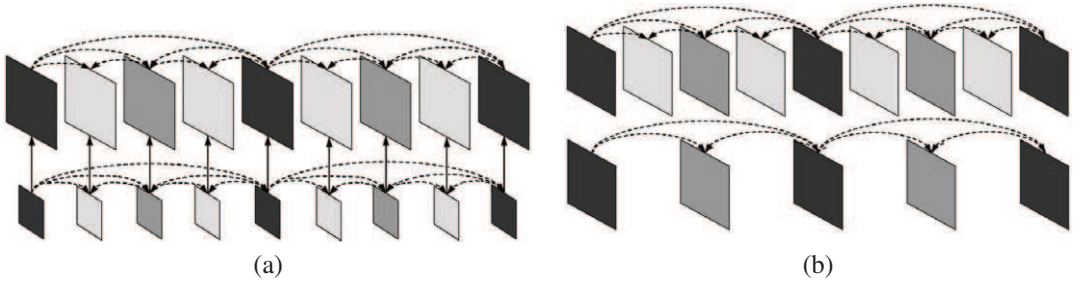

Fig. 1. Examples of scalability offered by SVC: (a) spatial scalability, (b) temporal scalability.

2.2. SE based on CAVLC and CABAC

Here, we are giving an overview of the SE based on CAVLC and CABAC which has already been presented in [6, 7]. Our SE techniques fulfill the real-time constraints by keeping the format compliance, the bitrate unchanged and utilizing negligible processing power. To keep the bitstream format compliant, we cannot encrypt MB header. Syntax elements which belong to MB header are usually used for prediction and their encryption makes the bitstream undecodable.
In SE of CAVLC, the encryptable *syntax elements* are the *signs* of T1's, *levels* and their *signs*. CAVLC use multiple tables to adapt the local statistics of DCT coefficients. To keep the bitrate unchanged, VLC

codewords, having same length, constitute the encryption space (ES). This ES equals to 2^n where n is the number of the VLC table being used.
In SE of CABAC, we encrypt the *bin strings* before the binary arithmetic coding (BAC) module for the sake of format compliance. The encryptable *syntax elements* are $suffix$ of NZs with $|NZ| > 14$ and *signs* of all the NZs. The encryption space is $log_2(n + 1)$ where n is the suffix part of absolute value of NZ.

3. The Proposed Algorithm

In scalable encryption, *interlayer prediction* option is very important and can be off, *on* or *adaptive*. If it is off and we use the same pseudorandom number generator (PRNG) to encode all the layers in scalable encoding, even a single layer from the bitstream cannot be decrypted and decoded no matter if it is a base layer or an enhancement layer. The reason lies in the fact that a scalable video decoder decodes only the requested layer and not the subsequent lower layers because this layer is not predicted from lower layers. So PRNG on encoder and decoder side will be out of synchronization and no layer can be decrypted and decoded. Thus if *interlayer prediction* is off, it is mandatory to use independent PRNG for each layers. In case *interlayer prediction* is *on* or *adaptive*, we can decrypt and decode only the highest resolution and not any lower layer. So even in this case, for playback of lower resolutions, it is mandatory to have separate PRNG for each layer.

Let us have a scalable bitstream containing three resolutions to be protected. On the encoder side, let us use three separate keys K_1, K_2 and K_3 to encode 1^{st}, 2^{nd}, and 3^{rd} layers respectively shown in Fig. 2. On the decoder side, if we want to decode any of the higher resolutions, we need the keys for that enhancement layer and all the lower layers sothat we can decode each layer properly. To avoid the need of multiple keys for higher resolutions, let us have three composite keys K_{c1}, K_{c2}, K_{c3}, each for 1^{st}, 2^{nd} and 3^{rd} resolution respectively. Then we need to build a function which can extract the layer key K_1 from all the three composite keys K_{c1}, K_{c2} and K_{c3}; key K_2 from keys K_{c2} and K_{c3} and K_3 from K_{c3} as shown in Fig. 3.

A first approach to solve this problem consists of concatenating the keys together. For example, for decryption of 2^{nd} resolution, we can concatenate $K_1\&K_2$. But it makes the size of keys different for every resolution which is not a desirable feature. To have the keys of equal length for each resolution, we concatenate zeros to keys of lower layers and then encrypt them using AES algorithm in CFB mode. In that case, AES will replace the zeros with non-zero values and as a result we get composite keys for every resolution as shown in Fig. 4.

For a scalable bitstream containing three layers, with the key length of 64 bits for each layer. To generate the KC3, we will concatenate all the three keys, thus making the length of composite key K_{c3} equals to 192. For generation of K_{c2}, we will put zeros in place of K_3 and will encrypt it with AES to get K_{c2} of 192 bits. Similarly for K_{c1}, we will replace both K_2 and K_3 with zeros and will encrypt it using AES.On the decoder side, when K_{c1} will be decoded by AES. It will output K_1 only. Similarly K_{c2} and K_{c3} will output respectively two and three keys for decoding of 2^{nd} and 3^{rd} resolutions.

For the encryption process, let the layer keys K_1, K_2 and K_3 are of 64 bits each and K_{c1}, K_{c2}, K_{c3} are of 192 bits. Fore encryption, we have used AES in CFB mode. In CFB mode, AES is a stream cipher.

In this mode, the previous encrypted block Y_{i-1} is used as the input of the AES algorithm in order to create keystream Z_i. Then, the current plaintext X_i is XORed with Z_i in order to generate the encrypted text Y_i.

For the 1^{st} iteration, Y_0 is substituted by an initialization vector (IV). IV is created using the secret key K_m as the seed of the PRNG. K_m is divided into 8 bits sequences. The PRNG produces a random number for each byte component of the key. Then, we use IV as Y_0. Z_i and Y_i are created as:

$$\begin{cases} Z_i &= E_k(Y_{i-1}), for\ i \geq 1 \\ Y_i &= X_i \oplus Z_i \end{cases}, \tag{1}$$

As illustrated in Fig. 5, with the CFB mode of the AES algorithm, the generation of the keystream Z_i depends on the previous encrypted block Y_{i-1}. Consequently, if two plaintexts are identical $X_i = X_j$ in the CFB mode, then always the two corresponding encrypted blocks are different, $Y_i \neq Y_j$. So using the stream cipher, we encrypt keys in a series. In this case, every encrypted key is used for the encryption of the subsequent key.

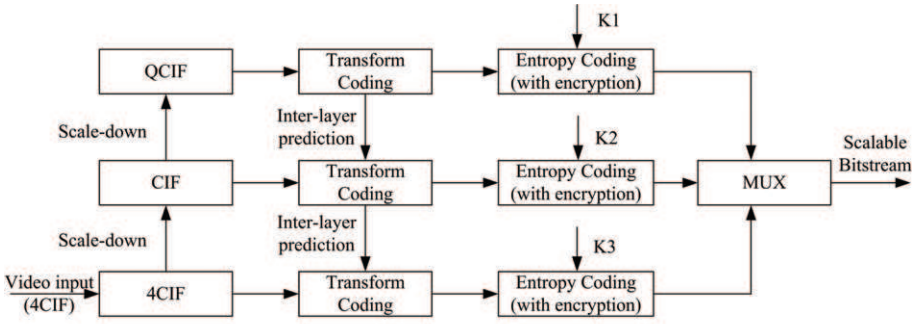

Fig. 2. Architecture for independent encryption of each layer in scalable video coding.

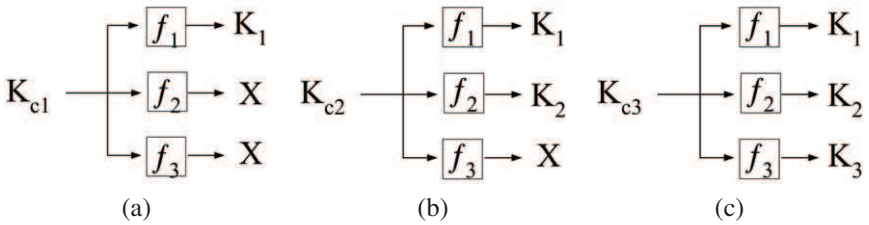

Fig. 3. Extraction of layer keys K_1, K_2 and $K3$ from the composite keys: (a) K_{c1}, (b) K_{c2}, (c) K_{c3}.

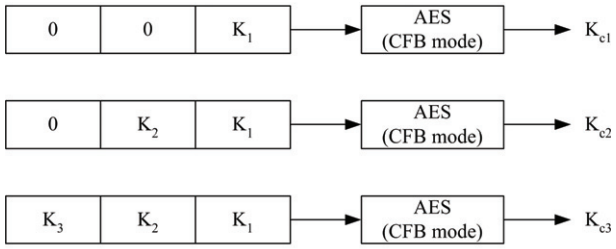

Fig. 4. Creation of composite keys K_{c1}, K_{c2} and K_{c3} from layer keys K_1, K_2 and K_3.

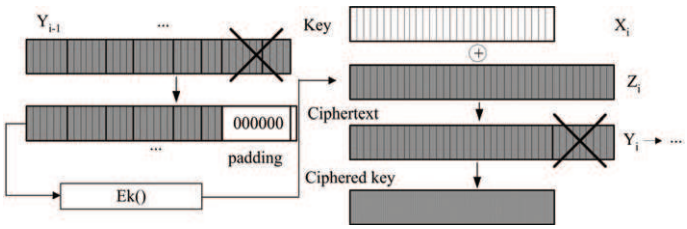

Fig. 5. AES encryption of layer keys in CFB mode.

4. Experimental Results

Let us have a scalable bitstream containing three independently protected layers having resolutions of QCIF, CIF and 4CIF with composite keys K_{c1}, K_{c2} and K_{c3}. For simulation, we have tested both off-line and on-line scenarios as explained in the following:

4.1. Off-line simulation

In first scenario, we have a scalable protected bitstream on the computer and want to playback it off-line. When the specific composite key is given, the decoder decrypts the composite key and hence, gets the layer keys. The user just gets the part of the bitstream to which he is authorized. In this application, bitstream extractor functionality is embedded in the *read_bitstream* module of decoder, thus it only provides the packets of those layers to the core decoding process which are to be decoded. Layer keys are used to initialize the PRNGs of respective layers and proper decryption of authorized resolution is performed.

4.2. On-line scenario scenario.

In client server application, the scalable bitstream is on the server, and the client sends the request to the server for playback of a certain video alongwith composite key. The *bitstream extractor* is the server application in this scenario and it extracts the layer keys. From the key information, it will send the packets of respective layers. On the client side, the decoder extract the layer keys and initialize the PRNGs with the keys. In this case, the client has no knowledge about the details of the SVC bitstream i-e how many layer it contains. The user just gets the part of the bitstream to which he is authorized, thus saving the bandwidth. Fig. 6 shows the decoding of a *city* frame when using K_{c1}, K_{c2} and K_{c3} keys. One can note that there is a

(a) (b) (c)

Fig. 6. PSNR (dB) of {Y, U, V} of 1^{st} frame of 4CIF resolution decoded with key: (a) K_{c1} {22.2, 40.0, 42.5}, (b) K_{c2} {24.2, 41.8, 44.7}, (c) K_{c3} {44.2, 47.6, 49.0}.

considerable degradation in the quality of the image when the enhancement layers are not decoded with the valid keys. This degradation will force the user to see the scaled-up version of the lower layer video which is authorized.

In our analysis, 100 frames of benchmark video sequences were encoded as *intra* in a scalable bitstream containing QCIF, CIF and 4CIF resolutions. Then the 4CIF resolution was decoded with composite keys K_{c1}, K_{c2} and K_{c3} as shown in Table 1 for all the benchmark video sequences at quantization parameter (QP) value of 18. Same analysis was performed for *city* in Table 2 over whole range of QP values. One can note that PSNR with the respective composite key is a lot more better than with composite keys of other layers.

In another experiment, we have analyzed whether decoding the lower layer with respective composite key and its scaled-up to 4CIF resolution is better than decoding the 4CIF resolution with composite key of lower layers. In Table 3, the comparison has been performed for K_{c1} for all the sequences and the same experiment was performed for *city* for different QP values in Table 4. The result shows that scaled-up 4CIF resolution has better quality than decoded with K_{c1}. The difference between scaled-up and decrypted versions also reduces as the QP value goes high. The results are also confirmed by same experiment for K_{c2} in Table 5 and Table 6.

Table 1. PSNR for decoded 4CIF with different compsite keys for benchmark video sequences at QP value of 18 for *intra* frames.

Seq.	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	K_{c1}	K_{c2}	K_{c3}	K_{c1}	K_{c2}	K_{c3}	K_{c1}	K_{c2}	K_{c3}
city	22.7	24.9	44.2	40.2	42.1	47.7	42.6	44.7	48.9
crew	28.4	31.3	44.9	39.2	41.4	46.5	37.8	41.4	47.7
harbour	21.1	25.1	44.3	39.8	42.2	47.4	41.6	44.2	48.7
ice	28.4	31.1	46.1	42.5	45.4	51.0	38.5	42.4	51.6
soccer	25.1	27.5	44.8	40.0	42.1	47.6	42.1	44.4	49.1

Table 2. PSNR for decoded 4CIF with different compsite keys for *city* at different QP values for *intra* frames.

QP	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	K_{c1}	K_{c2}	K_{c3}	K_{c1}	K_{c2}	K_{c3}	K_{c1}	K_{c2}	K_{c3}
12	22.5	24.8	49.4	41.0	42.6	51.1	43.9	46.1	51.9
18	22.7	24.9	44.2	40.2	42.1	47.7	42.6	44.7	48.9
24	23.2	25.2	39.4	39.1	40.7	44.4	41.4	43.2	46.1
30	23.5	25.5	35.0	37.7	39.3	42.2	40.1	41.9	44.3
36	22.6	24.7	30.9	36.5	37.9	40.1	38.3	40.1	42.2
42	21.1	22.9	27.1	35.4	37.1	39.0	37.3	38.7	40.4

Table 3. Comparison of scaled QCIF and decrypted 4CIF with K_{c1} of different resolutions for benchmark video sequences at QP value of 18 for *intra* frames.

Seq.	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
	scaled	encrypted	scaled	encrypted	scaled	encrypted
city	24.7	22.7	40.9	40.2	43.8	42.6
crew	30.1	28.4	40.0	39.2	37.9	37.8
harbour	23.1	21.1	40.6	39.8	42.5	41.6
ice	30.7	28.4	43.1	42.5	39.7	38.5
soccer	27.3	25.1	40.3	40.0	43.0	42.1

Table 4. Comparison of scaled QCIF and decrypted 4CIF with K_{c1} for *city* at different QP values for *intra* frames.

QP	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
	scaled	encrypted	scaled	encrypted	scaled	encrypted
12	24.8	22.5	41.6	41.0	44.9	43.9
18	24.7	22.7	40.9	40.2	43.8	42.6
24	24.6	23.2	39.7	39.1	42.4	41.4
30	24.2	23.5	38.8	37.7	41.2	40.1
36	23.5	22.6	37.9	36.5	39.8	38.3
42	22.4	21.1	37.1	35.4	38.7	37.3

Table 5. Comparison of scaled CIF and decrypted 4CIF with K_{c2} of different resolutions for benchmark video sequences at QP value of 18 for *intra* frames.

Seq.	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
	scaled	encrypted	scaled	encrypted	scaled	encrypted
city	27.0	24.9	42.5	42.1	45.6	44.7
crew	33.2	31.3	42.6	41.4	42.1	41.4
harbour	27.2	25.1	43.1	42.2	45.3	44.2
ice	33.5	31.1	46.4	45.4	44.0	42.4
soccer	29.7	27.5	42.9	42.1	45.5	44.4

Table 6. Comparison of scaled CIF and decrypted 4CIF with K_{c2} for *city* at different QP values for *intra* frames.

QP	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
	scaled	encrypted	scaled	encrypted	scaled	encrypted
12	27.0	24.8	43.3	42.6	47.0	46.1
18	27.0	24.9	42.5	42.1	45.6	44.7
24	26.8	25.2	41.1	40.7	43.9	43.2
30	26.3	25.5	40.0	39.3	42.7	41.9
36	25.4	24.7	38.9	37.9	41.0	40.1
42	23.8	22.9	38.1	37.1	39.6	38.7

5. Conclusion

A novel framework for independent access of different resolutions in SVC based on CAVLC and CABAC has been presented. The proposed scheme is quite efficient for local playback and in streaming environment on heterogeneous network in terms of bandwidth and processing power. The proposed scheme works fine whether we use *interlayer prediction* or not. Owing to no escalation in bit rate, our encryption scheme is suitable for heterogeneous multimedia streaming scenarios in real-time environment. The experiments have shown that we can achieve the protection of each resolution independently while providing the same composite keys of same length for different resolution. It has been shown that scaled_up version has better PSNR than decoded version with composite key of lower layers. The proposed scheme can be successfully extended for temporal and quality scalability.

References

[1] A. Uhl, A. Pommer, Image and Video Encryption: From Digital Rights Management to Secured Personal Communication, Springer, 2005.
[2] S. Lian, Z. Liu, Z. Ren, Z. Wang, Selective Video Encryption Based on Advanced Video Coding, Lecture notes in Computer Science, Springer-verlag (3768) (2005) 281–290.
[3] C.-P. Wu, C.-C. Kuo, Design of Integrated Multimedia Compression and Encryption Systems, IEEE Transactions on Multimedia 7 (2005) 828–839.
[4] M. Grangetto, E. Magli, G. Olmo, Multimedia Selective Encryption by Means of Randomized Arithmetic Coding, IEEE Transactions on Multimedia 8 (5) (2006) 905–917. doi:10.1109/TMM.2006.879919.
[5] W. Jiangtao, K. Hyungjin, J. Villasenor, Binary arithmetic coding with key-based interval splitting, IEEE Signal Processing Letters 13 (2) (2006) 69–72. doi:10.1109/LSP.2005.861589.
[6] Z. Shahid, M. Chaumont, W. Puech, Fast Protection of H.264/AVC by Selective Encryption, in: SinFra 2009, Singaporean-French IPAL Symposium, Fusionopolis, Singapore, 18-20 Feb. 2009.
[7] Z. Shahid, M. Chaumont, W. Puech, Fast Protection of H.264/AVC by Selective Encryption of CABAC for I & P frames, in: Proc. 17th European Signal Processing Conference (EUSIPCO'09), Glascow, Scotland, 2009, pp. 2201–2205.
[8] I. .-. I. technology Generic coding of moving pictures, . associated audio information: Video, 2nd Ed.
[9] . ISO/IEC 14496-2:2004 Information technology Coding of audio-visual objects: Visual, 3rd Ed.
[10] V. C. f. L. B. C. ITU Telecommunication Standardization Sector of ITU, in: ITU-T Recommendation H.263 Version 2, 1998.