First published in the Proceedings of the 18th European Signal Processing Conference (EUSIPCO'10) in 2010, published by EURASIP.

SELECTIVE ENCRYPTION OF THE MC-EZBC BITSTREAM AND RESIDUAL INFORMATION

Heinz Hofbauer and Andreas Uhl

Department of Computer Sciences
University of Salzburg
{hhofbaue, uhl}@cosy.sbg.ac.at

ABSTRACT

When selective encryption is used for security in DRM schemes some information of the original bitstream is intentionally left in plain text. This can have various reasons, e.g. generating preview versions for try and buy scenarios. In the case of the MC-EZBC there is also the goal of retaining the scaling capability in the encrypted domain. However, since parts of the bitstream remain in plaintext this information is available to a potential attacker at all times. In this paper we will assess which attacks can be done with this residual information. Consequently we will extend a prior version of selective encryption for the MC-EZBC to include motion vectors.

1. INTRODUCTION

The use of digital video in todays world is ubiquitous. Videos are viewed on a wide range of clients, ranging from hand held devices with QVGA resolution (320x240) over PAL (768x576) or NTSC (720x480) to HD 1080p (1920x1080) or higher. Furthermore, streaming servers should be able to broadcast over the internet with regard to a wide range of bandwidths, from fixed high bandwidth lines like ADSL2 to various low bandwidths for mobile wireless devices. In such an environment it is simply not possible to encode a video for each application scenario. So content providers either have only a fixed number of options available or they use scaling video technology to adapt the video for bandwidth and resolution requirements of the client. The concept of creating the content once and adapting it to the current requirements is preferable and is better known as Universal Multimedia Access (UMA) [10].

One of the enabling technologies of UMA is the use of scalable video coding. This averts the need for transcoding on the server side and enables the server to scale the video. However, even scaling takes up computation time and reduces the number of connections the server can accept. Furthermore, variable bandwidth conditions, which happen frequently on mobile devices, further taxes the server with the need to adapt the video stream. The solution to this is usually in-network adaption, shifting the need to scale to a node in the network where a change in bandwidth is occurring. The core adaption with these restrictions takes place on the server and adaption due to varying channel capability is done in-network. For design options and comparisons of in network adaption of the H.264/SVC codec see Kuschnig et al. [8]. Wu et al. [11] give an overview of other aspects of streaming video ranging from server requirements to protocols, to QoS etc.

For video streaming in the UMA environment, i.e. a high number of possible bandwidths and target resolutions, wavelet based codecs should be considered. Wavelet based codes are intrinsically highly scalable and rate adaption as well as spatial and temporal scaling can easily be done. Furthermore, wavelet based codecs achieve a coding performance similar to H.264/SVC, c.f. Lima et al. [9]. For an overview about wavelet based video codecs and a performance analysis as well as techniques used in those codecs see the overview paper by Adami et al. [1]. Under similar considerations Eeckhaut et al. [4] developed a complete server to client video delivery chain for scalable wavelet-based video. The main concern of research regarding UMA is usually performance with respect to scaling and in-network adaption. However, digital rights management and security is also a prime concern.

These considerations on network streaming and the inherent scaling capability of wavelet based codecs lead to the development of a selective encryption approach [6] for the MC-EZBC(motion compensated embedded zeroblock coder) [7, 3] video codec. In this approach information was left in plain text in order to be format compliant, meaning that even the encrypted bitstream is decodeable by a standard decoder. Additionally, this approach allows scalability in the encrypted domain.

In section 1.1 an overview will be given about security, selective encryption and objectives of an attack. In order to facilitate the understanding of the encryption method and attacks a short overview of the MC-EZBC bitstream will be given in section 1.2.

In section 2 we will investigate the information which was intentionally left in plain text, namely motion fields and header information in order to mount attacks on the video sequence. While we will specifically look at the MC-EZBC video codec similar attacks are possible on other video and image codecs, e.g. [5] for a header information attack on JPEG2000.

In section 3 the selective encryption method will be extended to include motion vectors and section 4 will give a summary over the attacks and the extended encryption approach.

1.1 Overview Over Selective Encryption

Selective encryption refers to encrypting, carefully selected, parts of a plaintext. Two common reasons for this approach are reduction in resources, usually time saved when only a part of a plaintext is encrypted, and maintaining properties of the plaintext in the encrypted domain. The discussed selective encryption approach for the MC-EZBC is of the second kind where the objective is to retain the ability to scale the encrypted bitstream.

Furthermore selective encryption can be utilized to pro-

tect only parts of the bitstream for digital rights management (DRM) scenarios, e.g. a freely decodeable preview version with embedded but encrypted high quality version.
The possible security goals we want to achieve with selective encryption in different DRM scenarios are as follows:

Confidentiality Encryption means MP security (message privacy). The formal notion is that if a system is MP-secure an attacker can not efficiently compute any property of the plaintext from the ciphertext [2].

Sufficient Encryption means we do not require full security, just enough security to prevent abuse of the data. Regarding video this could for example refer to destroying visual quality to a degree which prevents a pleasant viewing experience.

Transparent Encryption means we want people to be able to view a preview version of the video but in a lower quality while prevent them from seeing a full version. This is basically a pay per view scheme where a lower quality preview version is available from the outset to attract the viewers interest. The distinction is that for sufficient encryption we do not have a minimum quality requirement, and often encryption schemes which can do sufficient encryption cannot ensure a certain quality and are thus unable to provide transparent encryption.

Regarding attacks the focus will be to breach message privacy under the assumption that the visual data is fully encrypted. We will look at header and motion field information and determine what information can be produced regarding the content of the video sequence.

1.2 The MC-EZBC Bitstream

A schematic overview of the MC-EZBC bitstream is given in fig. 1 and an illustration of the decomposition of a GOP is given in fig. 2. The main layout is a header followed by GOP sizes (this is the size of the image data in a GOP) followed by a sequential ordering of GOPs. Each GOP is lead by a header, giving scene change information, i.e. which frames are I frames, followed by the motion field and image data. Motion field and image data are kept separate. For image data the frames are ordered lowest to highest temporal resolution (which is equal to lowest to highest temporal frequency bands). Likewise for each frame the image data is stored from lowest to highest resolution (which is equal to lowest to highest spatial frequency bands). Motion vector fields are stored lowest to highest temporal resolution and in order of frame for each temporal band, in case a given frame is stored as an I-frame the motion vector field for this frame is omitted. Each base layer and each enhancement layer is stored as chunk of data (not shown in the figure), meaning a leading header giving the length of the data block followed by the data block itself.

For a parsing of the bitstream the layout into chunks is beneficial since we do not have to search for marker sequences but can directly skip large parts of the file. Also when headers, including chunk headers, and GOP size information is kept intact the whole bitstream can subsequently be parsed correctly, which is important to be able to scale after the encryption.

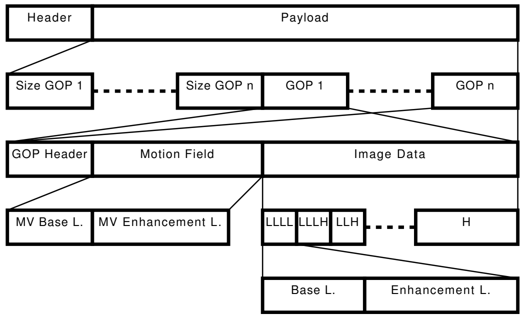


Figure 1: The layout of the MC-EZBC bitstream

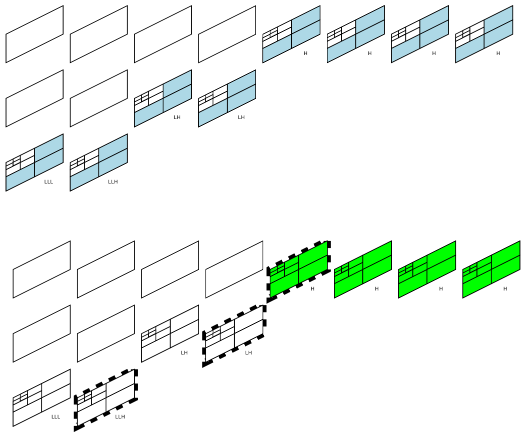


Figure 2: Overview of the decomposition of a GOP with GOP size 8 with marked high temporal layer (lower part), high spatial layer (upper part) and possible I frames as dashed outline on the lower part.

2. RESIDUAL INFORMATION

The original approach to selective encryption of the MC-EZBC [6] leaves the header and motion information unencrypted. The motion information is left unencrypted in order to be able to decode the bitstream with the original MC-EZBC implementation. The header information is used for scaling and has to be changed when scaling is performed, so an encryption is not possible. In the following this residual information from the selective encryption approach will be used to gain information about the encrypted video sequence. The akiyo, bus, coastguard, container, flower, foreman, mobile, news, silent, tempete and waterfall sequences are used to perform these tests. In the following subsections we refer to full selective encryption which means the format compliant encryption of image data, cf. fig. 1 and [6], leaving header information and motion fields in plain text.

2.1 Header Information

Assuming an attacker intercepts a video stream which is encrypted using full selective encryption. Assuming further that we do have a catalog of available videos from the source of the stream. If this information is present can we identify the video sequence which was intercepted? If this is possible message privacy would be breached since an attacker is able to identify the video sequence.

Since the header information is available a video stream with the same scaling parameters (bitrate and resolution) can be requested. The size of the motion field and visual data is a part of the plain text headers in the encrypted stream. Using this information we can identify whether the requested stream matches the intercepted stream. Also note that this can be done even if the new stream is sent encrypted and we have no possibility of decrypting it. For each stream requested a similarity score S will be calculated in comparison to the intercepted stream as follows,

$$S = \sum_{i \in MV} (o_i - c_i)^2$$

where MV is the set of indices of motion vector chunk lengths, o_i and c_i are the length of the ith motion vector chunk of the original and comparison sequence respectively.

In the following experiment the sequences were split into subsequences, each 8 frames in size, in order to simulate a larger catalog of video sequences as well as to show that even for this low number of frames the similarity score identifies the source sequence with precision.

In fig. 3 a plot is shown where the waterfall32 subsequence, starting at frame 32, is compared to other subsequences, including waterfall. The dashed line shows subsequences not connected to the waterfall sequence, the solid line show subsequences from the waterfall sequence and the mark at the abscissa shows the waterfall32 subsequence.

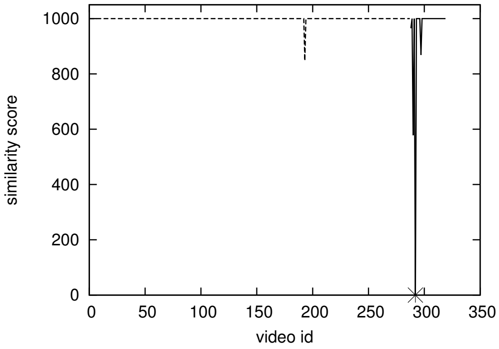


Figure 3: Similarity calculation for various sequences compared to waterfall32.

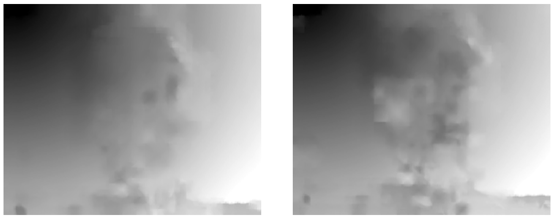
The plot is artificially capped at 1000 in order to show the more interesting lower range of similarities. From the illustration it can be seen that other subsequences originating from the same sequence can also give a similarity response, overall this is because the type of motion throughout subsequences are quite similar. The only other subsequence with a decent similarity response is news8, sequence id 193, which is a near static image with a downward motion from dancers in the center similar to the motion in the waterfall sequence. This similarity of motion over 8 frames is the reason for the response, the longer the subsequences the lower the similarity response outside a video sequence will be.

2.2 Motion Vector Information

Assuming an attacker intercepts a full selective encrypted sequence it is possible to inject image data into the bitstream in order to gain information about the content, which again breaches message privacy.

A visual object can be injected into the video sequence by encoding a still image sequence of the injected object and merging the two sequences using the motion information from the original sequence and the visual data from the still image sequence. The main header can be kept since it is the same for both sequences resulting from using the same parameters for encoding the still image sequence. Motion header and image header information is taken from the respective sequence. This leaves only the GOP length information to be adjusted which is a trivial task. Regarding which object to inject there are two possible courses, one is to analyze the motion field in order to gain information about the sequence. The other is to identify the sequence by using the header information as described in the previous section and utilize side channel information.

By analyzing the motion field it is relatively easy to determine in which parts of the image actual motion is happening as opposed to general movement like panning or zooming. A simple way of doing this is injecting a gradient image and watch the resulting sequence. In the example of the foreman sequence it is easily discernible that the sequence is of the head and shoulders type, see fig. 4.



Frame 50 Frame 100

Figure 4: Frames 50 and 100 of the foreman sequence with injected gradient image.

The related attack is given in fig. 5 where a head is inserted into the foreman sequence. For encoding a GOP length of 128 was used and just the first GOP of the sequence will be used here. A head which roughly fits the proportions of the moving object in the center of the image was inserted, the inserted head has not the exact right size nor the right proportions. Note that the background in the inserted image was left blank since there is nearly no background motion in this GOP to work with.

While only two frames of the sequence are compared in fig. 5 it can be seen that the inserted head goes through the same motion as the original foreman head. In the actual video sequence even the movements of the mouth are perceivable. In any case the quality is a dramatic improvement over a direct decoding of the encrypted sequence, frame 15 and 62 are shown in fig. 6.

Under the assumption that the video sequence can be identified through the header information a search can be done for still images from the actual sequence. Given that such a still image can be found, either a preview version or a screenshot of the video sequence, a much better approximation can be done. In the example given in fig. 7 we used frame 20 of the foreman sequence to inject. The steps for injecting the image data are the same as for the more general case, but the result is much better. This is mostly due to the pictures being more similar and thus the artefacts introduced by motion compensation are less visible.

This second attack using motion vectors also makes the identification of the video sequence through header informa-

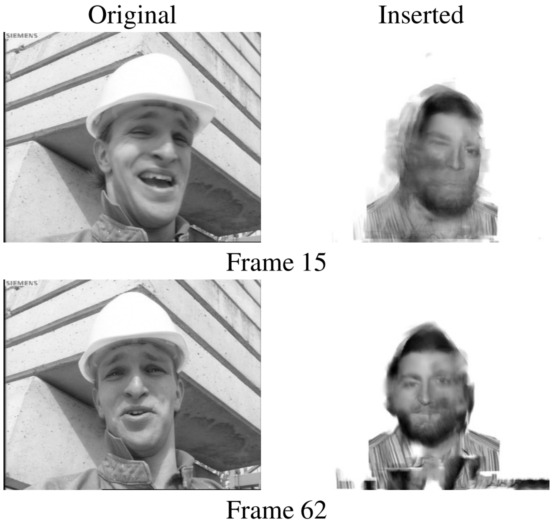


Figure 5: Foreman frames 15 and 62 compared with an injected image of a head.

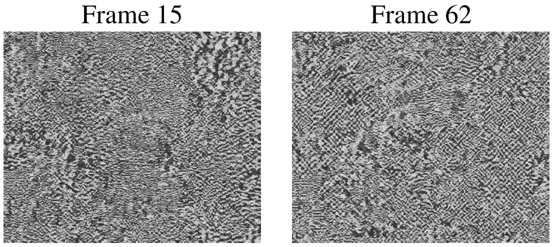


Figure 6: Frame 15 and 62 of a direct decoding of the encrypted foreman sequence.

tion much more dangerous. Not only do we gain knowledge about the video sequence but we can mount a more effective attack on the sequence.

3. SELECTIVE ENCRYPTION WITH MOTION VECTORS

The current MC-EZBC video codec supports scalable motion vectors [12]. Motion vectors are available for each temporal resolution and thus are structured in order of temporal resolution first and frame order in the given resolution second. In terms of the bitstream the motion data is, like the image data, given in chunks, i.e. a leading header gives the length information of the following block of data. The amount of motion data in relation to the whole bitstream, depending on the bitrate of the sequence, ranges from 0.5% (full bitrate) to nearly 40% (128kbps) under full temporal resolution.

The primary goal of adding encryption to motion vectors is still to keep the scalability intact in the encrypted domain. However unlike with corrupt image data the decoder is far less resistant to errors in the motion vectors. This results in format compliance only on a bitstream level, i.e. the bitstream can still be parsed and scaled, but the standard decoder will most likely be unable to deal with the random input of the encrypted motion vectors.

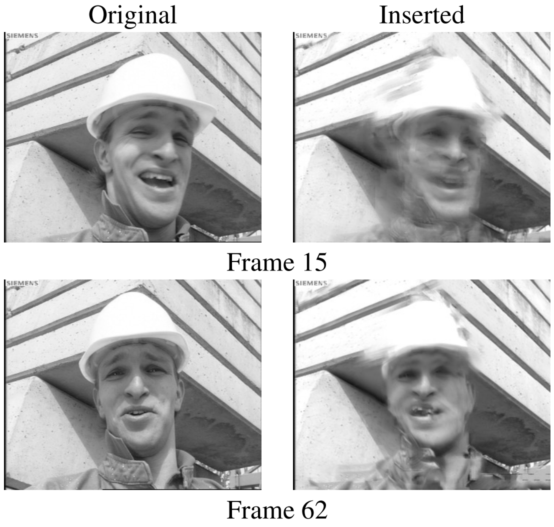


Figure 7: Foreman frames 15 and 62 compared with an injected image of foreman frame 20.

The encryption of the data in motion field chunks is not block aligned so a stream cipher has to be used. Furthermore, scaling away higher temporal resolution can disrupt ciphers in feedback mode, like AES in OFB, when the feedback is used over all chunks. Consequently it is best like with the original version of the encryption algorithm to use feedback only in a given chunk. The motion data encryption alone can not be used for sufficient or transparent encryption.

In order to assess the encryption of motion vectors only two attacks are used. One is the injection of a zero motion field into the bitstream similar to what is described in section 2.2. The other is to fix up the decoder to prevent it from crashing during motion field decoding. In the case where motion data is required beyond the bound of a chunk we introduce a one bit spike to prevent the decoder from locking up in a loop waiting for a symbol. Furthermore, the referencing to image data outside the boundaries of a given frame is prevented. The fix of the decoder will in the following be referred to as ”mvfix” attack.

For sufficient encryption, depending on the video sequence, the quality can be too high. Figure 8 shows the PSNR of the tempete sequence, high global motion, and silent sequence, a head and shoulder sequence with low global motion, for injection and mvfix attacks. In this attacks all motion fields were encrypted. For sequences with distinct global motion the mvfix attack does better because the residuals are distributed throughout the image while for a zero motion field the residual information is accumulated which leads to severe color bleeding. Figure 9 illustrates the color bleeding effect frames 62 and 250 of the tempete sequence. This effect becomes less distinct when the GOP size decreases. Additionally, the mvfix attack introduces more jitter resulting in a lower viewing quality.

Regarding transparent encryption the problem is how to control a target quality. The way to use motion vector encryption for transparent encryption would be to force the receiver to downscale on the temporal resolution, i.e. reducing the frame rate. Since the downsampling is done with wavelets the difference from the original frames are somewhat hard to measure since video quality indices (VQI)

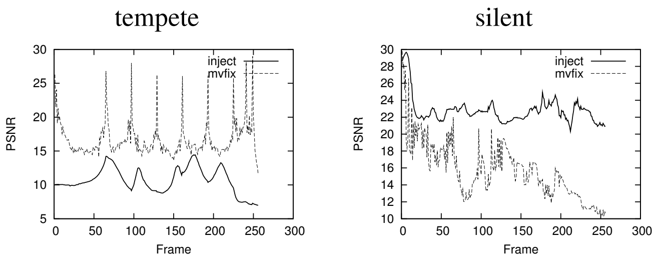
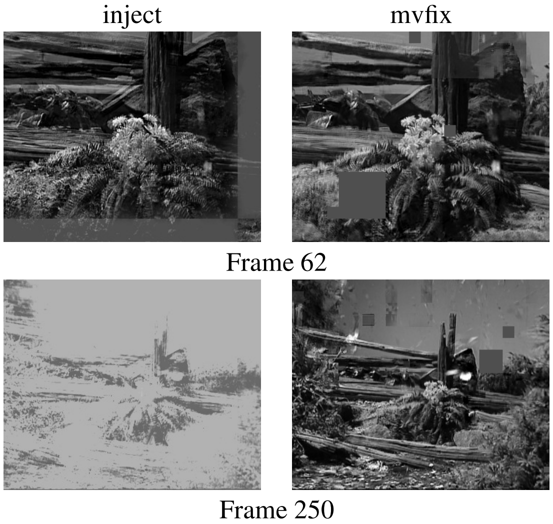


Figure 8: PSNR plot showing the comparison of the injection and mvfix attacks on the tempete and silent sequence with GOP size 256


inject

mvfix

Frame 62

Frame 250

Figure 9: Comparison of injection and mvfix attacks based on frames 62 and 250 of the tempete sequence with GOP size 256.

like PSNR would rate the blurring effects introduced by the downsampling as severe degradation even if the content is still viewable. Furthermore, the impact of zero motion injection or mvfix attacks are hard to evaluate purely on the basis of a VQI.

4. CONCLUSION

It was shown that confidentiality can not be reached with selective encryption for the MC-EZBC, header data alone can be used to identify a video sequence. Motion fields if left unencrypted have been shown to compromise content, i.e. an approximation of the content can be created using only motion vectors.

An enhancement of a selective encryption scheme to include motion vectors has been introduced and discussed in detail. The encryption of motion vectors alone has been shown to be insufficient for transparent or sufficient encryption schemes. However, the encryption of motion vectors can prevent reconstruction attacks as presented in this paper and should be used in conjunction with the selective encryption of image data.

Furthermore, since header data has to be left in plain text in order to allow scalability in the encrypted domain the identification attack is always possible. This shows that full cryptographic security can only be achieved with traditional methods, e.g. AES encryption over the whole bitstream.

REFERENCES

[1] N. Adami, A. Signoroni, and R. Leonardi. State-of-the-art and trends in scalable video compression with wavelet-based approaches. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1238–1255, September 2007.

[2] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In *Proceedings of Selected Areas in Cryptography, SAC '09*, volume 5867, pages 295–312, Calgary, Canada, August 2009. Springer-Verlag.

[3] P. Chen, K. Hanke, T. Rusert, and J. W. Woods. Improvements to the MC-EZBC scalable video coder. In *Proceedings of the IEEE Int. Conf. Image Processing ICIP*, volume 2, pages 81–84, Barcelona, Spain, 2003.

[4] H. Eeckhaut, H. Devos, P. Lambert, D. De Schrijver, W. Van Lancker, V. Nollet, P. Avasare, T. Clerckx, F. Verdicchio, M. Christiaens, P. Schelkens, R. Van de Walle, and D. Stroobandt. Scalable, wavelet-based video: From server to hardware-accelerated client. *Multimedia, IEEE Transactions on*, 9(7):1508–1519, November 2007.

[5] Dominik Engel, Thomas Stütz, and Andreas Uhl. Format-compliant JPEG2000 encryption in JPSEC: Security, applicability and the impact of compression parameters. *EURASIP Journal on Information Security*, 2007(Article ID 94565):20 pages, 2007.

[6] Heinz Hofbauer and Andreas Uhl. Selective encryption of the MC EZBC bitstream for DRM scenarios. In *Proceedings of the 11th ACM Workshop on Multimedia and Security*, pages 161–170, Princeton, New Jersey, USA, September 2009. ACM.

[7] Shih-Ta Hsiang and J. W. Woods. Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank. *Signal Processing: Image Communication*, 16(8):705–724, May 2001.

[8] R. Kuschnig, I. Kofler, M. Ransburg, and H. Hellwagner. Design options and comparison of in-network H.264/SVC adaptation. *Journal of Visual Communication and Image Representation*, pages 529–542, September 2008.

[9] L. Lima, F. Manerba, N. Adami, A. Signoroni, and R. Leonardi. Wavelet-based encoding for HD applications. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1351–1354, July 2007.

[10] A. Vetro, C. Christopoulos, and T. Ebrahimi. From the guest editors - Universal multimedia access. *IEEE Signal Processing Magazine*, 20(2):16 – 16, 2003.

[11] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha. Streaming video over the internet: approaches and directions. In *Circuits and Systems for Video Technology, IEEE Transactions on*, volume 11, pages 282–300, Mar 2001.

[12] Yongjun Wu, Konstantin Hanke, Thomas Rusert, and John W. Woods. Enhanced MC-EZBC scalable video coder. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(10):1432–1436, October 2008.