Selective Encryption of the MC EZBC Bitstream for DRM Scenarios

Heinz Hofbauer and Andreas Uhl
Department of Computer Sciences
University of Salzburg
Jakob-Haringer Str. 2, A-5020 Salzburg, Austria
{hhofbaue,uhl}@cosy.sbg.ac.at

ABSTRACT

Universal Multimedia Access (UMA) calls for solutions where content is created once and subsequently adapted to given requirements. With regard to UMA and scalability, which is required often due to a wide variety of end clients, the best suited codecs are wavelet based (like the MC-EZBC) due to their inherent high number of scaling options. However, we do not only want to adapt the content to given requirements but we want to do so in a secure way. Through DRM we can ensure that the actual content is safe and copyright is observed. However, traditional encryption removes the option of scalability in the encrypted domain which is opposed to what we want to achieve for UMA. The solution is selective encryption where only a part of the content is encrypted, enough to ensure safety but at the same time little enough to keep scalability intact. Towards this goal we discuss various methods of applying encryption to the bitstream produced by the MC-EZBC in order to keep scalability intact in the encrypted domain while also keeping security intact with regard to various DRM scenarios.

Categories and Subject Descriptors

K.4.4 [**Computer and Society**]: Electronic Commerce—*Security*; I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Security

Keywords

Security, in-network adaption, wavelet, selective encryption, scalability,DRM

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 changing low bandwidths for mobile wireless devices. In such an environment it is simply not possible to encode a video for every 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) [25].

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 the 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 actual 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. [10]. Wu et al. [26] 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 naturally highly scalable and rate adaption as well as spatial and temporal scaling is easily achieved. Furthermore, wavelet based codecs achieve a coding performance similar to H.264/SVC, c.f. Lima et al. [13]. 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. [5] 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.

Shannon [22] in his work on security and communication

MM&Sec'09, September 7–8, 2009, Princeton, New Jersey, USA.

made it clear that the highest security is reached through a secure cipher operating on a redundancy free plain text. Current video codecs exploit redundancy for compression and we can consider the bitstream to be a redundancy free plain text in the sense of Shannon. Thus for maximum security we just need to encrypt the whole bitstream with an state of the art cipher, i.e. AES. But we also loose the flexibility of the scalable bitstream. If we want to continue scaling in the network we have to provide the key to every node in the network where we want to perform scaling. However, the required key management is another likely security risk since it generates more attack points, i.e. key transmission and the receiving network node could be targeted to gain access to the key. However, if we relax our security standard, i.e. we do not want perfect security, then it is possible to combine security and scalability. This is exactly what we will assess in this paper.

Selective encryption is the encryption of only a part of the bitstream we wish to protect, usually with the goal of keeping some information contained in the file accessible. While this lowers the security of the encrypted bitstream it also yields benefits. The first thing we should realize is that often we do not need full security, take television broadcasting for example. It is not necessary to prevent people from recognizing what movie is airing on an encrypted channel, we just want to reduce the viewing experience without the corresponding key. This is also a good example why we want to keep information intact: we do not want the receiver thinking it receives noise (and properly encrypted signals should look like a random signal) but we want it to recognize a valid signal, e.g. a video stream, we just do not want the receiver to be able to reconstruct the contents. Other goals could be to retain scalability, to generate preview versions from an encryption stream and so on.

Regarding security Lookabaugh et al. [14] showed that selective encryption is sound and demonstrated its relation to Shannon's work. However, in practice a bitstream is not always redundancy free, as required by Shannon. For example, Said [21] showed that side information can compromise security. And of course even the best video codec does not exploit all redundancies in the bitstream. As such, it is expedient to include an attack in the examination of a selective encryption scheme to be able to gauge the actual security. For an overview about prior selective encryption methods see the papers by Massoudi et al. [19] and Liu et al. [16].

So as stated our main goal is to keep scalability intact while providing security to some extent. The possible security goals we want to achieve with selective encryption in different DRM scenarios are as follows:

Confidentiality Encryption means complete security, except for the information we want to give away. This is not easily achieved, since headers and other information which are necessary to recognize a bitstream can contain information which can lead to an identification of the content, see [6] for an example of such an attack.

Sufficient Encryption means we do not require full security, just enough security to prevent abuse of the data. This is of course heavily dependent on what we want to achieve. In this case we want to prevent people without a key to be able to view the video sequence. This does not mean that we do not want them to recognize what is in the video sequence, we just want to reduce the visual quality to a level which is regarded as unviewable by the general public. Another goal of sufficient encryption is the reduction of computational complexity, e.g. less time or memory required as compared to traditional encryption.

Transparent Encryption means we want people to see 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. Also, computational complexity for transparent encryption is secondary, the main goal is to provide a preview version.

Regarding the standard H.264/AVC/SVC there has also been done research regarding selective encryption. For both AVC and SVC Magli et al. [17, 18] created a transparent encryption scheme. All the other works presented are regarding sufficient encryption of AVC only. The only bitstream oriented encryption schemes, i.e. encryption after compression, are done by Shi et al. [23] and Iqbal et al. [9] and are not format compliant, i.e. a standard coder would not be able to decode the encrypted bitstream. The methods proposed by Li et al. [12], Bergeron et al. [2] and Lee and Nam [11] are to our knowledge format compliant but also compression integrated. Especially the compression integrated algorithms are troublesome to use since a change of keys would require a new encoding of the bitstream.

We want to apply selective encryption to the bitstream produced by the MC-EZBC [8, 3, 4, **?**] which is a t+2D scalable video codec. This choice was made mainly because the source code is available[1], which enables our experiments. Scalability in a video codec means that after one encoding step we get a bitstream which can be scaled to different bit rates, spatial and temporal (i.e. frame rate) resolutions, without reencoding the video sequence. The MC-EZBC uses motion compensated temporal filtering, with 5/3 CDF wavelets, followed by regular spatial filtering, with 9/7 CDF filtering, see fig. 3 for a GOP size of 8. This method, temporal first and spatial later, is referred to as t+2D coding scheme. For temporal filtering a full decomposition is used and thus the GOP size is discernible by the number of temporal decomposition levels t, i.e. GOP size = 2^t. Both temporal and spatial filtering are done in a regular pyramidal fashion. Statistical dependencies are exploited by using a bit plane encoder, the name giving embedded zero bit coder (EZBC), and motion vectors are encoded with differential pulse code modulation followed by an arithmetic coding scheme. Also note that I frames lead each GOP and furthermore can appear later in a GOP in case of a scene change (the dashed outline in fig. 3, lower part, shows possible occurrences of further I frames).

The outline of the paper is as follows. Section 2 gives an overview of the goals we want to achieve with the selective encryption, the method we use and a performance

[1] The source code for the ENH-MC-EZBC is available from `http://www.cipr.rpi.edu/research/mcezbc/`.

analysis. Experimental results for sufficient and transparent encryption are given in section 3. A summary, conclusion and outlook to future work is given in section 4.

2. SELECTIVE ENCRYPTION

Our goal with selective encryption is to achieve sufficient and transparent encryption while conserving the scalability in the encrypted domain. If we were to use regular encryption we would have to decrypt the bitstream prior to scaling and reencrypt it afterwards, which of course also requires that we have the key at the node which does the scaling. With our proposed method we can directly perform scaling on the encrypted bitstream, which not only saves time (since we can skip the de- and encryption steps), but also simplifies key management since we now only need the key at the endpoints of the channel. However, assuming that the unencrypted bitstream is our plaintext and the selectively encrypted bitstream is the ciphertext, then some portions of the ciphertext are copies of the plaintext. This means that perfect security, as specified by Shannon, can not be achieved, as this would require a full traditional encryption with a state of the art cipher.

A preview is naturally a lower quality version of the original sequence, but so is a downscaled version for a device which has a limited resolution. For example, the preview sequence of a HD video might be even better than the normal quality of the sequence if it is viewed on a mobile phone. This dichotomy cannot be readily resolved since really low level end devices border the region to sufficient encryption, e.g. a preview for a video sequence on a cell phone may not be viewable at all. And versions which could be considered preview sequences on a hand held device might be regarded as unviewable when watched on HD ready devices, e.g. when upscaling a sqCIF version of the sequence to a HD resolution the occurring pixelation will effectively degrade quality.

2.1 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. 3. 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. For both motion field and image data the frames are kept separate, i.e. no interleaving of frames, and 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). 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. In our context the encryption of image data is called *selective encryption*, i.e. we do not encrypt headers, motion fields or chunk size. From the remaining data, which constitutes about 99% of the bitstream we can choose what to actually encrypt. If we choose to encrypt all we will denote that *full selective encryption*, if we choose to encrypt a subset we denote that as *partial selective encryption*. The size of the data in chunks is not aligned in any way and scaling happens in the image data chunks. As such we need an encryption scheme which can encrypt arbitrary block length and which does not reorder bytes, e.g. no ciphertext stealing. Given this information the choice of AES in OFB mode seems reasonable, since OFB mode has the desired properties of keeping the bitstream in order while AES is a well known state of the art cipher. Note that every cipher which does not rearrange bytes and can be cut of is useable here, e.g. basically every stream cipher. Since the visual data is easily accessible in the bitstream it seems to be a good choice to separate encryption and encoding, resulting in the work flow shown in fig. 2. The program `pull` was provided with the MC-EZBC source and does bitstream adaption, `inspect` is our tool to view the layout of the bitstream, encrypt and attack it.

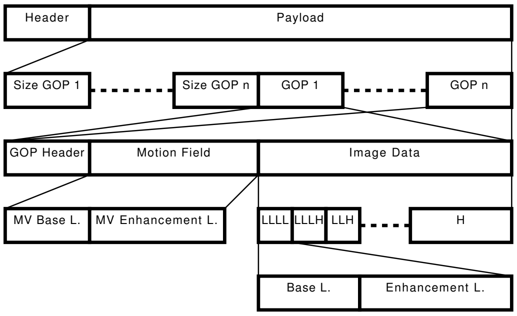


Figure 1: The layout of the MC-EZBC bitstream

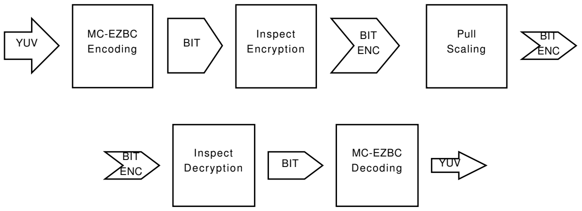


Figure 2: Workflow of the encoding, scaling, decoding process with encryption

2.2 Scaling Performance Analysis

The computational performance of selective encryption vs. traditional encryption is discussed controversially in literature. Basically, parsing and locating of what to encrypt generates an overhead and often a full traditional encryption is faster, especially with fast ciphers like AES. One can of course claim that the added advantage of keeping the ability to scale in the encrypted domain is worth the tradeoff of 'slow' encryption but it is still interesting to see how well we do.

2.2.1 Runtime Overview

Table 1 shows an overview of a full run through the work flow outlined above, and shown in fig. 2. The sequence en-

Table 1: Performance of the various steps in the work flow for the Flower sequence with a total of 128 frames and GOP size 128.

encoding	15m 47s 33ms	97.67%
encryption	148ms	0.02%
scaling	96ms	0.01%
decryption	50ms	0.01%
decoding	22s 344ms	2.30%
total	16m 9s 671ms	100.00%

coded was the well known flower sequence with a total of 128 frames and a temporal resolution of 7, resulting in a GOP size of 128. The highest quality version of the sequence is encrypted (all image data but no headers or motion vectors), then the sequence is downscaled to 128kbps (in the encrypted domain) and subsequently decrypted. What we see is that compared to encoding, and even decoding, the encryption and decryption process is extremely fast, and scaling is likewise. However, in terms of performance we should rather look at the absolute values, since if a bitstream is given (e.g. in retrieval scenarios like video on demand) encoding is not considered. For the highest quality version of the sequence we can encrypt, or decrypt, with a speed of roughly 1.15ms/frame and for the 128kbps version we have about 0.4ms/frame for full selective encryption. This translates to a throughput of about 870 frames per second for the full quality stream and 2500 frames per second for the downscaled version.

2.2.2 *Traditional vs. Full Selective Encryption*

While overall the performance is quite good the question remains how the full selective encryption process compares to full traditional encryption when scaling is applied. Taking the same high quality flower bitstream as above we perform full traditional encryption and full selective encryption, where the latter amounts to 99.41% of this bitstream. The encrypted bitstream is then downscaled. For traditional encryption we need to decrypt the bitstream prior to scaling and reencrypt it after scaling was performed. For full selective encryption we can directly scale the encrypted bitstream.

Full traditional encryption takes 114ms and full selective encryption takes 148ms, resulting in a speedup of 0.77. So if we do not scale the full traditional encryption is faster. Full selective encryption encrypts nearly the same amount of data as traditional encryption and also has a parsing overhead.

When we perform scaling however full selective encryption is faster since we can skip the decryption and encryption steps before and after scaling. Scaling takes 96ms for both encryption methods. With traditional encryption we have to decrypt before (114ms) and encrypt after (39ms) scaling. Thus, we get a total of 249ms for traditional encryption and 96ms for full selective encryption resulting in a speedup of 2.59.

The performance of partial selective encryption will be discussed in section 3.3.

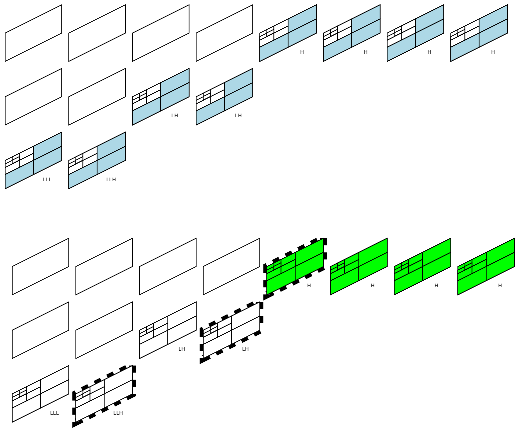


Figure 3: Overview of the decomposition of a GOP with GOP size 8 with marked high temporal layer (lower part), high spatial layer (upper part) and possible I frames as dashed outline on the lower part.

3. EXPERIMENTAL RESULTS

Since the various parts of the bitstream are basically wavelet decomposed signals we have a clear idea what to encrypt using partial selective encryption. For sufficient encryption we will target the low frequency bands, both temporally and spatially, as well as I frames. For transparent encryption we will encrypt the high bands, reducing the detail level of the sequence. To illustrate this, fig. 3 shows an overview of the decompositions, the marked frames in the lower part show the highest temporal band versus the highest spatial band in the upper part. The lower part of the figure also depicts possible I frames after the first frame in the GOP. Of course feeding a random signal into the arithmetic decoder will produce visual garbage in any case so it is expedient to consider an attack on the encrypted video sequence. This provides us not only with more insight into how well the sufficient encryption does but also gives us a method to remove the encrypted part of the sequence for the generation of the preview video for transparent encryption.

There are a number of possible attacks in literature. For an overview of selective encryption and attacks see Engel et al. [7] for JPEG2000, and Lookabaugh et al. [15] for MPEG-2. Specifically there are attacks which copy structurally similar symbols from one part of the bitstream to another or inject a forged version into the bitstream. This aims at removing the distortion introduced by decoding the encrypted bitstream or making decoding possible at all. These attacks also try to improve the resulting quality of the attack by forging the injected part of the bitstream in a way to minimize the decoding error. In literature such an attack is known as *error concealment attack* or *replacement attack*, a detailed description of such an attack can be found in Podesser et al. [20].

We will consider the error concealment attack of nulling out the encrypted part of the sequence. This basically exploits the fact that the arithmetic coder then maps the attacked part of the sequence to the most common output. While this also messes up the length of the bitstream segment with regard to the decoder we can still use it since the length is explicitly given. This allows the decoder to properly reset after the attacked part of the sequence and continue the proper decoding. Also note that although in the still images presented here structural information may not, or only hardly be visible, the structure can often be seen better when the actual video sequence is seen in motion. So even if the attacked images sometimes give the impression that we have achieved confidential security, this is not the case. Also note that we will use the encryption only on selected parts of the image like the low temporal bands to get a better idea how this influences the video sequence, while in an actual application scenario one would probably mix these encryption schemes, e.g. encrypt low temporal and spatial bands at the same time.

The sequences used in this section will be Container and Waterfall, both with a length of 256 frames and a GOP size of 256 (leading to 8 temporal levels) with CIF resolution. No scaling was performed and a full quality sequence was used as base for the experiments.

3.1 Sufficient Encryption

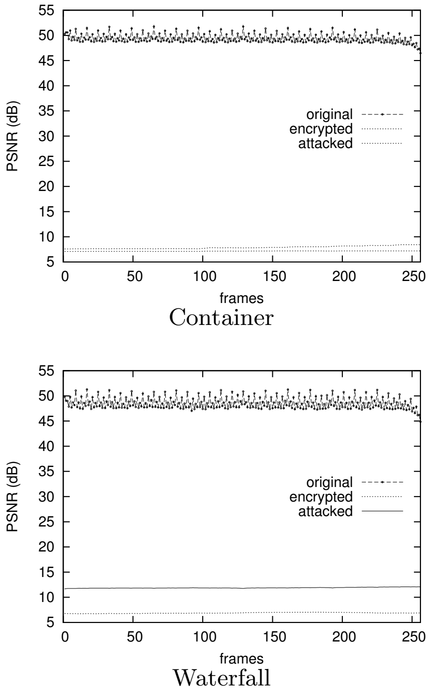
For sufficient encryption we to target the parts of the bitstream which codes the visually most significant data. The codec exploits redundancy and inter frame dependencies and concentrates the high information content of the video in the lower frequency bands, both temporal and spatial. The low frequency frames effect all frames in their GOP through the wavelet synthesis and are thus prime targets for sufficient encryption. Likewise, the I-frame introduce information into the current GOP and effect frames in a pyramidal fashion (stemming from temporal decomposition). This makes I frames also good candidates for sufficient encryption. In the following we will look at the influence of I frames and low frequency frames for sufficient encryption. Each possibility will be evaluated on its own to better gauge the effect it has on the resulting video quality.

3.1.1 I Frame Encryption

To encrypt I frames is a good way to conceal a high amount of visual information. Figure 4 shows the PSNR per frame plot for the Container and Waterfall sequences for the baseline, encrypted and attacked version of the stream. Here the encrypted version is a decoding of the stream without prior attack or decryption, the attacked version has the encrypted parts of the bitstream nulled prior to decoding it. Depending on the sequence the attack can only obtain a limited amount of information: for Container which is a slow pan most information is stored into the motion field so naturally the refinement information has less energy. The Waterfall sequence on the other hand is a zoom which cannot be compensated as well by the motion estimation and this is clearly visible in the attacked version where we basically have a comparison of the refinement information with the original sequence. For a comparison of image quality between Container and Waterfall see fig. 5. In any case as can be seen from the PSNR plot the visual quality can be considered to be sufficiently degraded for our purpose, and even our attack hardly improves the visual quality.

3.1.2 Low Frequency Band Encryption

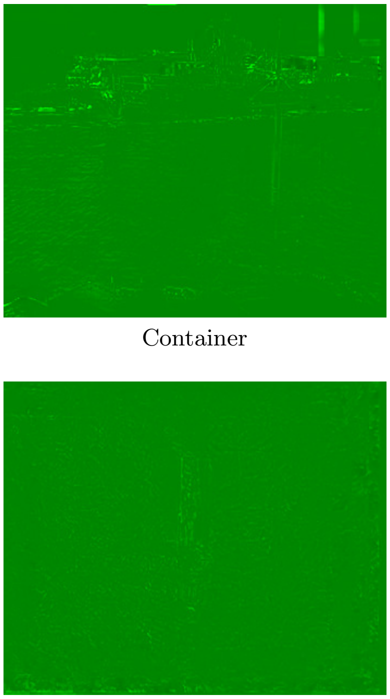
The next part of the bitstream which contains a high amount of information are the low frequency bands, temporal as well as spatial. Both are good candidates for encryption.



Container

Waterfall

Figure 4: PSNR per frame plot for the Container and Waterfall sequences for encrypted and attacked I frames.



Container

Waterfall

Figure 5: Frame 128 of the Container and Waterfall sequence with encrypted and attacked I frames.

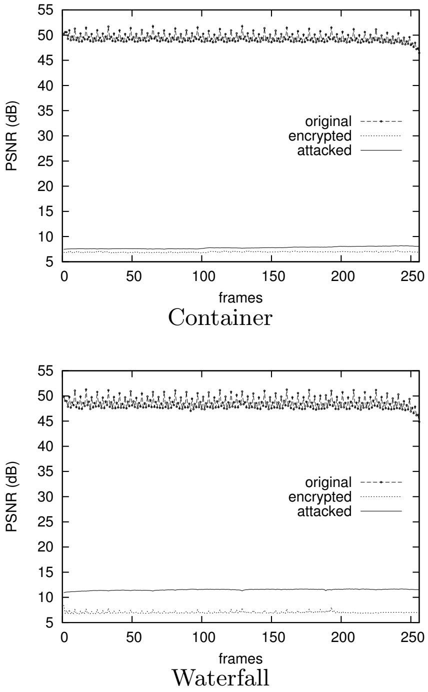
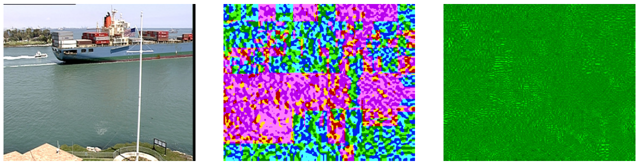


Figure 6: PSNR per frame plot for the Container and Waterfall sequence for encrypted and attacked low spatial frequencies.

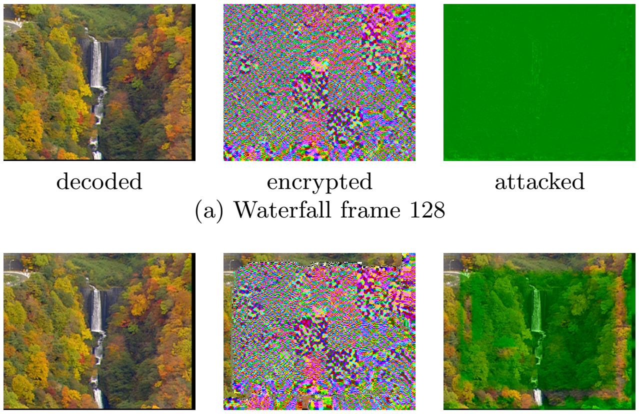
The PSNR over frame plots for the encryption of low frequency spatial bands for both sequences, again original, attacked and encrypted versions, are given in fig. 6. The PSNR plot looks quite similar to the I frame case, as we actually did encrypt parts of the I frames as well. The advantage of encrypting the low frequency bands is of course that we also encrypt large parts of the temporal refinement information. To get a rough idea of how much information is left, fig. 7 shows frame 128 for the Container sequence in encrypted, decoded and attacked version. The encrypted version is a garbled output which stems from the fact that we actually input a random signal into the arithmetic decoder. The attacked image in this case looks rather inconspicuous but still gives of quite a bit of information when it is viewed as a motion sequence. This is also the main distinction between encrypted I frames and encrypted low spatial frames. The I frame version shows a much clearer attacked image where edges can be directly identified while the low spatial frequency version really needs motion to properly recognize structure. This can be easily seen when comparing the attacked Container sequence in fig. 7 (low spatial bands) and fig. 5 (I frames).

Encrypting the low temporal frequencies we expect something similar to the I frame version since GOPs in the MC-EZBC bitstream start with I frames, this coincides with the lowest temporal frequency. The PSNR plot for Container and Waterfall can be seen in fig. 9 and frame 128 of the decoded, attacked and encrypted version of the Waterfall sequence can be seen in fig. 8(a). What we can clearly see, and which was to be expected, is that for the Waterfall sequence, which contains a scene change, the PSNR rises after



decoded encrypted attacked

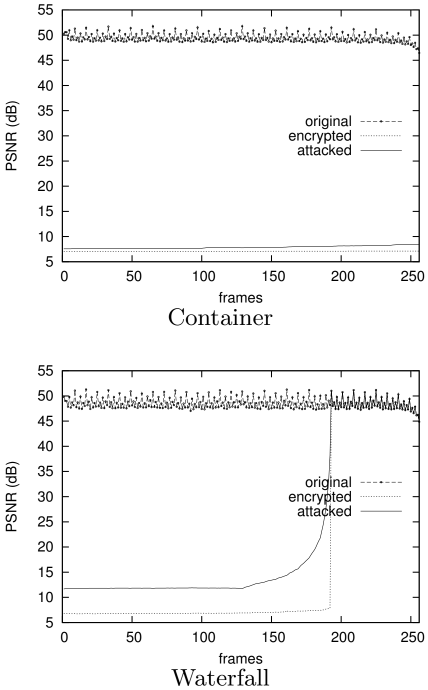
Figure 7: Comparison of encrypted, decrypted and attacked image to the original of frame 128 from the Container sequence (low spatial frequencies).



decoded encrypted attacked
(a) Waterfall frame 128

decoded encrypted attacked
(b) Waterfall frame 175

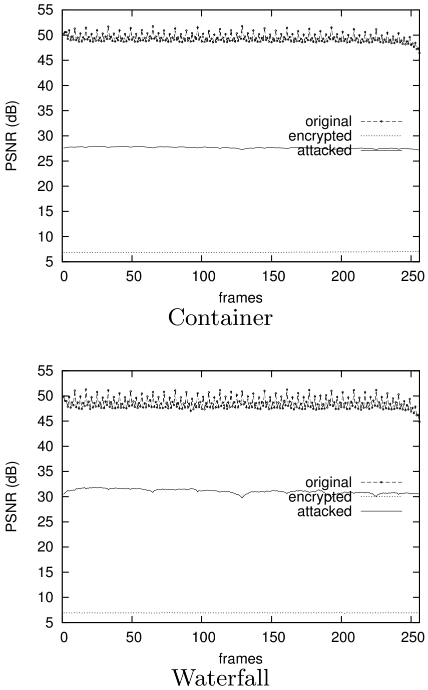
Figure 8: Comparison of encrypted, decrypted and attacked image to the original of frame 128 and 175 from the Waterfall sequence (low temporal frequencies).



Container

Waterfall

Figure 9: PSNR per frame plot for the Container and Waterfall sequence for encrypted and attacked low temporal frequencies.



Container

Waterfall

Figure 10: PSNR per frame plot for the Container and Waterfall sequence for encrypted and attacked high spatial frequencies.

the I frame in the later part of the sequence. Figure 8(b) shows frame 175 of the above versions for the Waterfall sequence, the influence of the I frame is clearly visible. The difference to the encrypted low spatial frequencies is rather obvious. Low temporal frequencies are full leading frames of the GOP, while low spatial frequencies are the low frequency information of all frames. Thus, low spatial frames include all I frames while low temporal frequencies only include leading I frames. Apart from the fact that we cannot ignore I frames when encrypting low temporal frames we can clearly see that encrypting low temporal frames also sufficiently destroys the visual quality. However, since we have to encrypt all I frames in addition to the low temporal frequencies it is usually sufficient to either encrypt I frames or low spatial frequencies (with or without full I frame encryption). Encryption of low spatial bands give a substantial gain vs. encryption of I frames only because they further destroy the visual quality of the difference frames. All versions however are sufficient to destroy the visual quality, while none gives confidential encryption.

3.2 Transparent Encryption

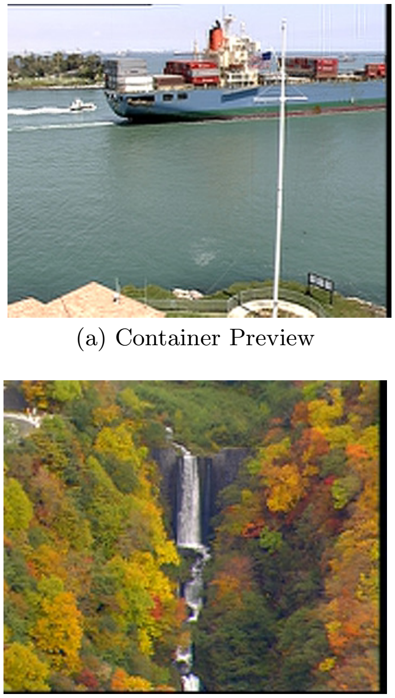
For transparent encryption the refinement information, residing in high frequency temporal and spatial bands, can be encrypted. The optimal solution would be to be able to completely choose a target PSNR for the preview image, this is not possible however since we only have a limited amount of steps, i.e. the decomposition depth of the sequence. However, adaption in this rough scale is possible and should be done.

3.2.1 High Spatial Frequency Bands

Figure 10 gives the PSNR plot for both Container and Waterfall sequences, with attacked highest spatial frequency band. The drop in PSNR is clearly visible, the impact on the visual quality is not quite as obvious however, as illustrated in fig. 11(a), for the Container sequence. What we can see is that our attack (in this case rather preview image generation) is working well. However, the reduction in visual quality is not really as high as expected. To remedy this we will have to encrypt an additional layer of the decomposition. Figure 12 gives an overview what changes in this case for the Waterfall sequence. Now the degradation in visual quality is clearly visible, even though the PSNR dropped only an additional 5 dB. This also gives an impression of the scale on which we can adjust the visual quality with this method.

3.2.2 High Temporal Frequency Bands

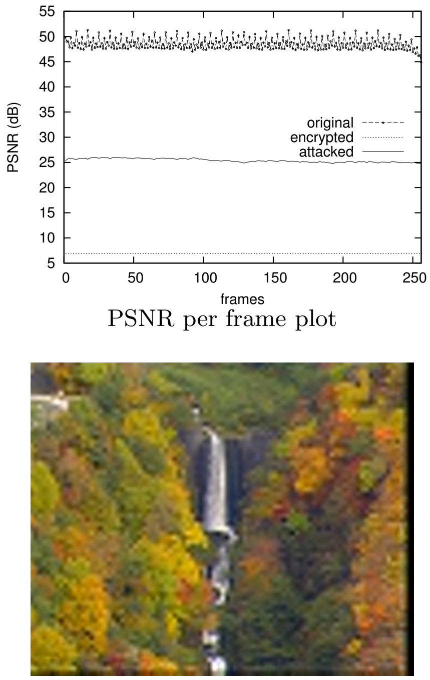
For high temporal bands the matter is a bit different. While spatial bands directly affect image quality, temporal bands do so to a lesser degree. They influence visual quality of course through blurring effects stemming from temporal filtering, but the main effect is a reduction in temporal resolution, i.e. frames per second. This can only partially be shown in a PSNR plot and still images, but nonetheless fig. 13 shows the PSNR plots for Container and Waterfall where the highest two (of eight total) temporal bands are encrypted. The visual impact can be seen in fig. 11(b), for the Waterfall sequence, the main effect being blurring which can be best seen at the waterfall itself. To show a stronger version of the blurring effect we also did a version where the



(a) Container Preview

(b) Waterfall Preview

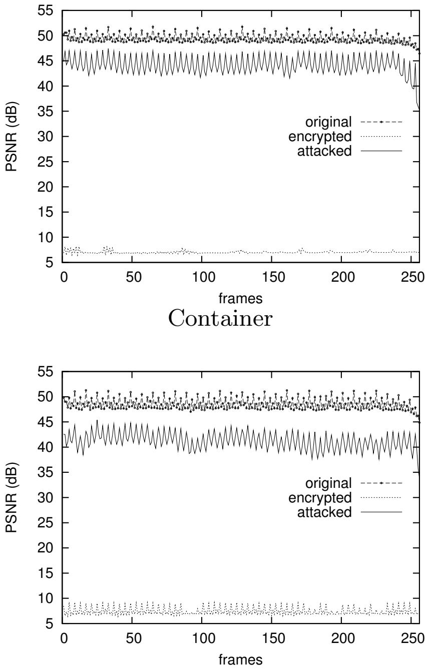
Figure 11: Preview image of the Container, high spatial frequencies, and Waterfall sequence, high temporal frequencies, frame 128.



PSNR per frame plot

frame 128 of the preview sequences

Figure 12: PSNR per frame plot for the Waterfall sequence and frame 128 of the preview (two highest spatial frequency bands encrypted) sequence.



Container

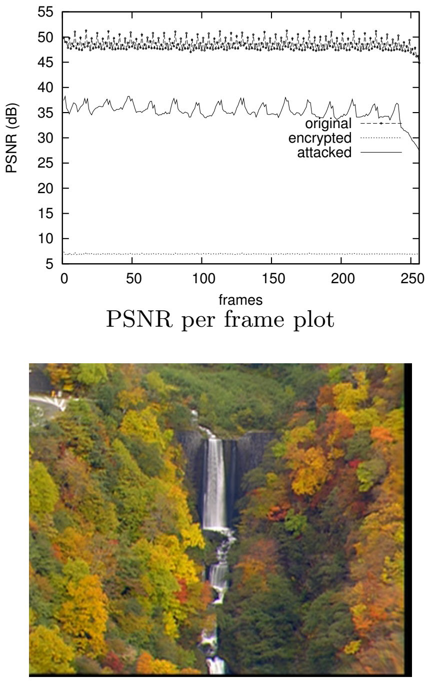
Waterfall

Figure 13: PSNR per frame plot for the Container and Waterfall sequence for encrypted and attacked high temporal frequencies.

highest four (half total) temporal frames are encrypted, seen in fig. 14 (both PSNR and visuals). When looking at the PSNR plots we can see some spikes in the preview image. These are the images in the temporal sequence which best fit the original sequence, the degradation following these frames stems from the fact that the still frames are simply not changed but the original sequence continues and moves away from the good fitting frame. Overall the visual viewing quality is heavily impaired since a bucking effect with blurring is introduced when a sufficient number of temporal frames are encrypted. However, when only the highest temporal layer is encrypted the skipping of every second frame is actually nearly not noticeable since the merging of temporally adjacent frames partly conceals the missing frame. It should also be noted that the highest temporal band is actually a full half of all frames, leading to a high amount of data to be encrypted when comparing this to the spatial case.

3.3 Partial Selective Encryption Performance

Instead of giving the information about encryption time and amount individually in each section, we have collected the information for all tests in table 2 for easier comparison. The sequence used was waterfall with 256 frames and GOP size 256 as well. The information given is for encryption only, scaling is not taken into account here since it was already discussed in section 2.2. The interesting thing to note here is that as soon as we take a step away from full selective encryption we are actually faster than full traditional encryption. While this was not our main concern it was a definite side target of sufficient encryption. Transparent en-



PSNR per frame plot

Frame 128 of the preview sequence

Figure 14: PSNR per frame plot for the Waterfall sequence and frame 128 of the preview (four highest spatial frequency bands encrypted) sequence.

Table 2: Performance comparison of the various selective encryption methods and full traditional encryption.

What was encrypted	time	% of Bitstream
Sufficient Encryption		
I-frames only	28ms	5.52%
lowest spatial band	99ms	34.79%
lowest temporal band	21ms	5.47%
Transparent Encryption		
highest spatial band	181ms	91.58%
two highest temporal bands	148ms	72.53%
four highest temporal bands	181ms	89.97%
Full Encryption		
full selective encryption	217ms	99.76%
full traditional encryption	201ms	100%

cryption, while faster than full selective or even traditional encryption, is slower than sufficient encryption. This is not surprising since the refinement layers, i.e. higher temporal bands, are significantly larger than the lower frames. Since transparent encryption has to target those high bands the amount of data to be encrypted is increased.

For sufficient encryption we have seen that the encryption of the lowest spatial bands performs best in terms of destroying visual quality followed closely by I-frames only. In terms of speedup we have about 2 for lowest spatial bands and more than 7 for I-frames when compared to full traditional encryption. When considering that even the sequence with encrypted I-frames is practically unusable, the choice is obviously the I-frame version since it gives the higher speedup.

For transparent encryption the speedup for spatial and temporal bands is about the same when we want to achieve a similar quality. Given that encryption speed is not even an objective for transparent encryption we can easily state that both versions are quite applicable. The real choice which to use thus is not performance but rather target quality.

4. CONCLUSION

We have introduced different ways to selectively encrypt the MC-EZBC bitstream with regard to transparent as well as sufficient encryption while being able to scale the bitstream in the encrypted domain. The proposed encryption schemes are fast and computationally cheap. Furthermore, the proposed encryption schemes meet all requirements of UMA while keeping security intact.

Concerning sufficient encryption we have shown that the destruction of the visual quality can easily and efficiently be achieved, but one has to be aware that encrypting low temporal bands is not enough, i.e. I frames have to be included. Overall the best practice is to either use I frames, low spatial bands or both combined, since I frames contain all the base layer information and low spatial frames contain the highest amount of energy from base and enhancement layers. For sufficient encryption we also achieved a gain in computational performance, e.g. when using only low spatial bands we require less than half the time of full traditional encryption.

Concerning transparent encryption we have shown that it is possible to achieve a reduction in quality by encrypting high spatial and frequency bands. While both methods are rather limited when it comes to possible output qualities, when combining both we have a sufficient number of possible quality steps. Assuming three spatial and eight temporal bands we would have a total of 24 possible output qualities. One should note however that, while reduction of visual quality through spatial encryption can easily be quantified this is not so simple for temporal bands, mainly because we are lacking a proper metric to measure bucking and lagging behavior in video sequences, except from the blurring which can be clearly seen in the PSNR plots. Concerning computational performance we can only register a slight improvement over full traditional encryption.

In future work we will look at the encryption of the motion fields, and closer investigate if it is possible to achieve full security, i.e. confidentiality, by encrypting motion fields as well as visual data. Furthermore, the use of a technique similar to the sliding window approach Stütz et al. introduced for JPEG2000 [24] would be beneficial to reduce the computational performance of transparent encryption.

5. 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, Sept. 2007.

[2] C. Bergeron and C. Lamy-Bergor. Compliant selective encryption for H.264/AVC video streams. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing, MMSP'05*, pages 1–4, Oct. 2005.

[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*, Barcelona, Spain, 2003.

[4] P. Chen and J. W. Woods. Bidirectional MC-EZBC with lifting implementation. In *IEEE Transactions on Circ. and Systems for Video Technology*, volume 14, pages 1183–1194, 2003.

[5] 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, Nov. 2007.

[6] D. Engel, T. Stütz, and A. Uhl. Format-compliant JPEG2000 encryption in JPSEC: Security, applicability and the impact of compression parameters. *EURASIP Journal on Information Security*, (Article ID 94565):doi:10.1155/2007/94565, 20 pages, 2007.

[7] D. Engel, T. Stütz, and A. Uhl. A survey on JPEG2000 encryption. *Multimedia Systems*, 2009. to appear.

[8] S.-T. 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.

[9] R. Iqbal, S. Shirmohammadi, and A. E. Saddik. Compressed-domain encryption of adapted H.264 video. In *Proceedings of the 8th International Symposium on Multimedia, ISM'06*, pages 979–984, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[10] 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*, Sept. 2008.

[11] H.-J. Lee and J. Nam. Low complexity controllable scrambler/descrambler for H.264/AVC in compressed domain. In K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, editors, *Proceedings of ACM Multimedia 2006*, pages 93–96. ACM, 2006.

[12] Y. Li, L. Liang, Z. Su, and J. Jiang. A new video encryption algorithm for H.264. In *Proceedings of the Fifth International Conference on Information, Communications and Signal Processing, ICICS'05*, pages 1121– 1124. IEEE, Dec. 2005.

[13] 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.

[14] T. D. Lookabaugh and D. C. Sicker. Selective encryption for consumer applications. *IEEE Communications Magazine*, 42(5):124–129, 2004.

[15] T. D. Lookabaugh, D. C. Sicker, D. M. Keaton, W. Y. Guo, and I. Vedula. Security analysis of selectiveley encrypted MPEG-2 streams. In *Multimedia Systems and Applications VI*, volume 5241 of *Proceedings of SPIE*, pages 10–21, Sept. 2003.

[16] X. Lu and A. M. Eskicioglu. Selective encryption of multimedia content in distribution networks: Challenges and new directions. In *Proceedings of the IASTED International Conference on on Communications, Internet and Information Technology (CIIT 2003)*, Scottsdale, AZ, USA, Nov. 2003.

[17] E. Magli, M. Grangetto, and G. Olmo. Conditional access to H.264/AVC video with drift control. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'06*. IEEE, July 2006.

[18] E. Magli, M. Grangetto, and G. Olmo. Conditional access techniques for H.264/AVC and H.264/SVC compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008. to appear.

[19] A. Massoudi, F. Lefèbvre, C. D. Vleeschouwer, B. Macq, , and J.-J. Quisquater. Overview on selective encryption of image and video, challenges and perspectives. *EURASIP Journal on Information Security*, 2008(Article ID 179290):doi:10.1155/2008/179290, 18 pages, 2008.

[20] M. Podesser, H.-P. Schmidt, and A. Uhl. Selective bitplane encryption for secure transmission of image data in mobile environments. In *CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002)*, Tromso-Trondheim, Norway, Oct. 2002. IEEE Norway Section. file cr1037.pdf.

[21] A. Said. Measuring the strength of partial encryption schemes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'05)*, volume 2, Sept. 2005.

[22] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, Oct. 1949.

[23] T. Shi, B. King, and P. Salama. Selective encryption for H.264/AVC video coding. In E. Delp and P. Wong, editors, *Proceedings of the SPIE, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 461–469, Feb. 2006.

[24] T. Stütz and A. Uhl. On efficient transparent JPEG2000 encryption. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '07*, pages 97–108, New York, NY, USA, Sept. 2007. ACM Press.

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

[26] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. 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.