H.264/AVC video authentication using skipped macroblocks for an erasable watermark

Dima Pröfrock*, Henryk Richter, Mathias Schlauweg, Erika Müller

University of Rostock, Institute of Communications Engineering,
Richard-Wagner Str. 31, 18119 Rostock, Germany

ABSTRACT

This paper presents a new watermarking framework, suitable for authentication of H.264 compressed videos. The authentication data is embedded as fragile, blind and erasable watermark with low video quality degradations. Because of using a fragile watermark, hard authentication is possible. In contrast to other approaches, the watermarking is done after the H.264 compression process. Hence, the authentication information can be embedded in already encoded videos. To reconstruct the original H.264 compressed video the watermark can be removed. The framework is based on a new transcoder, which analyses the original H.264 bit stream, computes a watermark, embeds the watermark and generates a new H.264 bit stream. To authenticate the video a hash value is used. This value is encrypted with a private key of an asymmetric cryptosystem. The payload of the watermark consists of the encrypted hash value and a certificate with the public key. Some skipped macroblock of the H.264 video are used to embed the watermark. A special process selects these macroblocks. This process sets the distribution and the number of skipped blocks as well as the number of embedded bits per block to achieve low video quality degradations and low data rate. To embed the watermark the performance of several approaches is discussed and analyzed. The result of the framework is a new watermarked H.264 bit stream. All data necessary for authentication are embedded and cannot get lost.

Keywords: digital watermarking, H.264, hard authentication, erasable watermark, skipped macroblocks

1. INTRODUCTION

With the rapid growth of current information technologies generating, editing and distributing of digital media data (audio, video, image, etc.) becomes increasingly trivial. The use of digital instead of analogue media data offers many advantages. It is now possible to make an exact copy of digital data or edit it without high effort. Simultaneous, these advantages result in problems. Because of perfect copies, cheap hardware and the World Wide Web, illegal copying and distribution of the media data is very easy. Everybody without special knowledge can edit and manipulate digital media data in a way, that a second person cannot recognize if the data is changed or not. Hence, there is a growing importance of applications such as data authentication, copyright and data hiding. Digital watermarking[1] offers contributions in these fields. Digital watermarking describes techniques to embed additional information into digital data. In this paper, we describe a watermarking framework to authenticate H.264 compressed digital video.

For authentication of digital media data, several fragile and semi-fragile watermarking techniques are known. The semi-fragile watermarks have two advantages. Firstly, they are robust to compression of digital data. Lin et al.[2] use S*pread Spectrum* watermarking to mark 8x8 blocks in images. The watermark is embedded in the DCT coefficients of middle frequency and is robust to JPEG compression. The second advantage is the property to detect altered regions in the digital data. Wolfgang and Delp use the *VW2D* Algorithm[3]. The image is divided into blocks, which are watermarked. For each block, the watermark detector computes a value, which shows whether or not the block has been altered. The disadvantage of semi-fragile watermarks is that they base on a threshold. Hence, the user can only estimate if the data is authentic. This is insufficient for the application of hard authentication like in medicine or forensic. Fragile watermarking offers the possibility of hard authentication.

Generally, hard authentication without watermarking is realized by using hash values (like MD5[4] etc.) and crypto-systems (like RSA[5] etc.). The video can be authenticated by a hash-value, the hash value can be protected with a key and the key can be verified from a trust centre. Hence, to authenticate a video, the user needs, additional to the video data, a

[*] dima.proefrock@uni-rostock.de, phone +49 (0)381 498 3596, fax +49 (0)381 498 3595

key or certificate and the encrypted hash value. Generally, these data are transmitted in a second file or are contained in the video metadata. Because of this, these data can get lost like in the example of Figure 1 a). User 2 does not need the authentication data. To save bandwidth user 1 transmit only the pure video data. Hence, user 3 has to contact the author of the video to get the authentication data. The use of a fragile watermarking approach can anticipate this loss like in the example in Figure 1 b). The authentication data is embedded in the pure video data. User 2 can only transmit the video file and delete all unnecessary metadata but the embedded authentication data can not get lost. We use a fragile watermarking technique to embed an authentication data set for hard authentication.

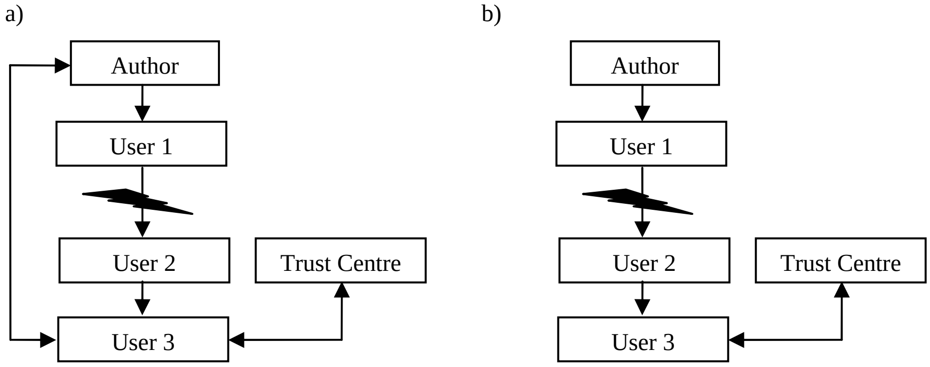

Figure 1. loss of authentication data a) and using watermarking to save the authentication data b)

Digital uncompressed video requires a plenty of memory to store it. Hence, digital video data normally is compressed. At present, H.264/AVC is the codec with the highest performance in video data compression[6]. In contrast to MPEG-2, H.264 offers about double compression rate at the same video quality. This is achieved by using an effective intra-coding and a quarter pixel accuracy motion estimation with multiple reference frames. There are three ways to embed the watermark in combination with video compression. The embedding can be done in the uncompressed domain, during or after the encoding. Watermarking in the uncompressed domain requires robust or semi-fragile watermarks. The approach of Serdean et al.[7] uses two watermarks. A first reference watermark in the spatial domain is used to obtain robustness to geometrical attacks. The payload is embedded as second watermark in the wavelet domain. Watermarking in the uncompressed domain does not allow hard authentication for the case of lossy video compression. Watermarking during the encoding process offers the possibility of hard authentication. Qui et al.[8] use motion vectors to embed fragile watermarks during a H.264 encoding. In dependence of the payload, the motion vectors are changed to realize odd or even motion vector prediction errors. Watermarking during the encoding process requires an implementation of the watermarking framework to the encoder. Our approach in contrast embeds the watermark after the H.264 encoding process. We achieve independence to H.264 implementations. The result is a “stand-alone” watermark framework to authenticate H.264 videos. Hence, already encoded H.264 videos can be watermarked.

An essential advantage of our approach is the possibility to erase the embedded watermark. Hence, the original compressed H.264 video data can be reconstructed. There are approaches, which embed watermark data by changing the source data slightly. For example, Byun et al.[9] insert the authentication data in the least significant bits of the original image to protect the other bit layers. However, some critical applications (i.e. medicine, forensic) require the original data set. In our approach, we change the video data in a reversible way to enable a complete video reconstruction. There are two ways to decode the watermarked video. Of course, the watermarked H.264 video conforms to standard H.264 decoder. In this case, the watermark results in video quality degradations. These degradations are minimized during the watermarking process. A second way of decoding and displaying the video is to consider the watermark during the decoding. Hence, the video can be displayed without any quality degradations. Therefore, a normal H.264 decoder can be modified with very low effort.

The organization of this paper is as follows. In Section 2, we introduce the developed framework. Next, Section 3 discusses the use of skipped macroblocks and shows the results of investigations to the frequency of occurrence of skipped macroblocks. Section 4 describes the principle of the macroblock selection process, the selection of skipped macroblock after a future analysis and the signal adaptation to optimize the relationship between PSNR and embedding overhead. In Section 5, we discuss the watermark payload, the embedding domain, the watermark detection and we investigate several watermarking techniques to embed the information bits in the skipped macroblocks.

2. THE PROPOSED FRAMEWORK

A block diagram of the proposed framework is shown in Figure 2. The framework is based on a transcoder, which analyses the original H.264 bit stream, computes a watermark, embeds the watermark and generates a new H.264 bit stream. The first step is an inverse entropy coding, which extracts video data out of the pure H.264 bit stream. In the next step, the video structure is analysed. NAL-Units, the largest parts of the structure, contain slices. Generally, I-, P- and B-slices are used. Depending on the type, a slice contains different macroblock types. Macroblocks contain the prediction error of the motion vectors and the DCT-coefficients. Some of the skipped macroblocks of the P- and B-slices are used for an erasable watermark. The macroblock selection process improves the ratio between video quality degradation and embedding overhead by varying distribution and number of skipped macroblocks and the number of embedded information bits per block. An encrypted hash value and a public key (with certificate) are embedded. Afterwards, the entropy coding of the video is done. The result is a new watermarked H.264 bit stream. All the data necessary for authentication is embedded in the video and cannot get lost.

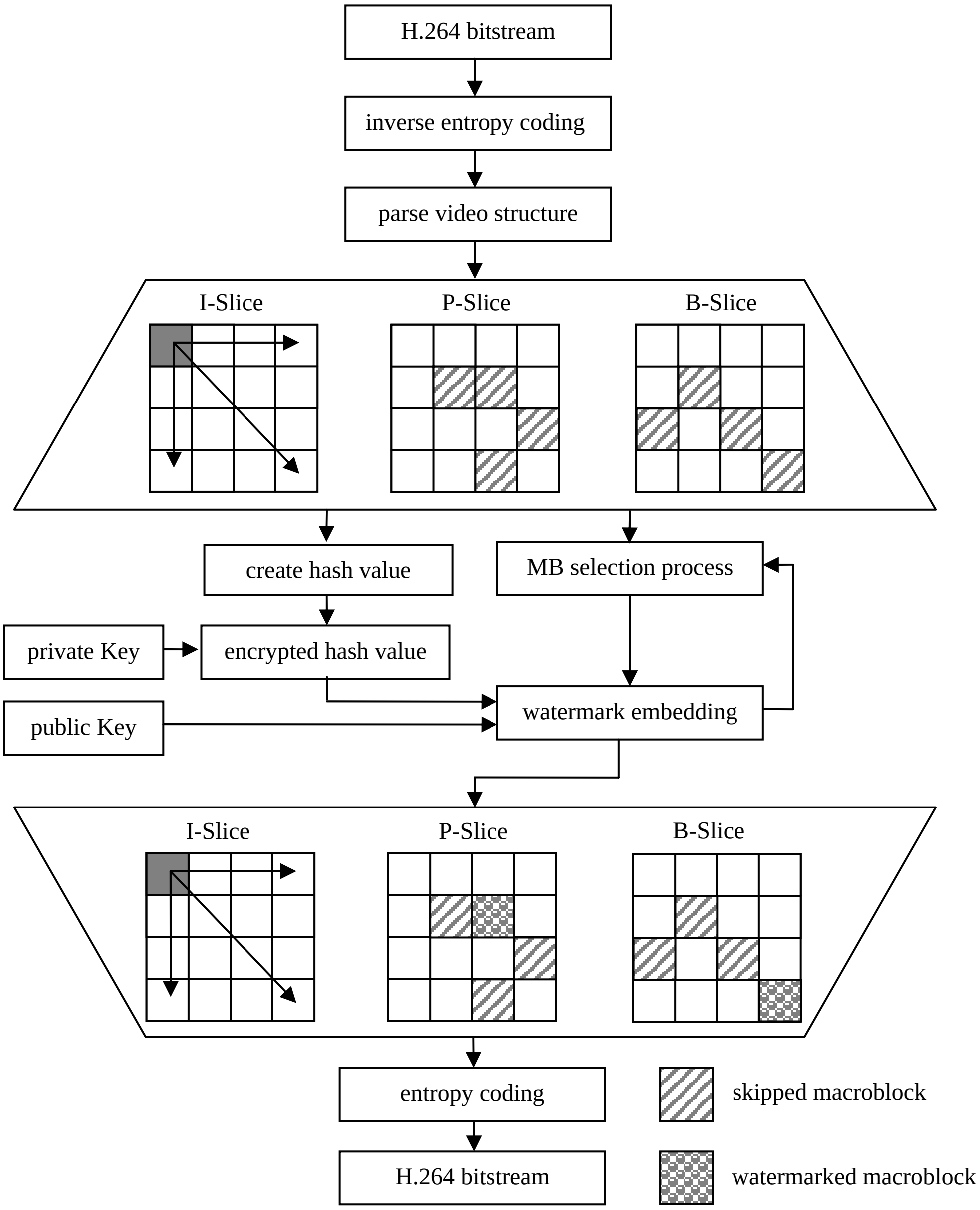

Figure 2. proposed Framework

3. SKIPPED MACROBLOCKS – THE BASIS

Macroblocks contain the prediction errors of their motion vectors and their DCT coefficients. The macroblock can be skipped, if these errors are zero. Only a counter, which describes the number of skipped macroblocks, exists in the H.264 bit stream. Our idea is to "reactivate" some of these skipped macroblocks. Because of using original skipped macroblocks, the watermark can be erased. Therefore, the watermarked macroblocks are converted to skipped macroblocks. The original video can be completely reconstructed.

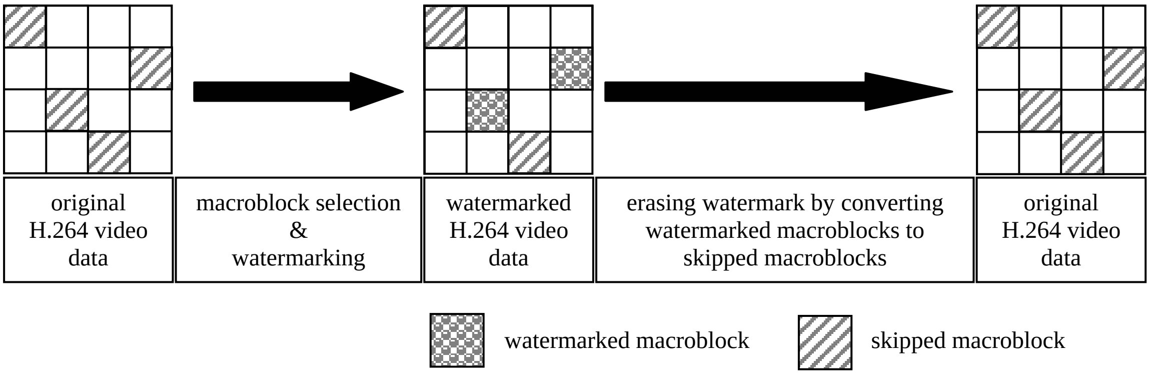
original H.264 video data
macroblock selection & watermarking
watermarked H.264 video data
erasing watermark by converting watermarked macroblocks to skipped macroblocks
original H.264 video data
watermarked macroblock
skipped macroblock

Figure 3. watermark embedding and erasing based on skipped macroblocks

During the "reactivation" of skipped macroblocks, some properties of H.264 are to consider. Under certain conditions, the motion vector prediction of skipped macroblocks differs from the motion vector prediction of unskipped macroblocks. Hence, a simple "reactivation" of skipped macroblocks results in a continuous motion prediction error, which results in heavy quality degradation. There are three ways to solve this problem. First, the problematic skipped macroblocks are not used. Second, the error of the motion prediction is compensated in the "reactivated" macroblock. Third, the error of the motion prediction is compensated in the succeeding macroblocks. To save the original video data, the third solution is not used. The error compensation of the second solution requires more bits as by using the unproblematic macroblocks. Hence, the problematic skipped macroblocks are not used for "reactivation".

Using skipped macroblocks implies that there are enough skipped macroblocks for "reactivation". Number and distribution of skipped macroblocks depend on the statistical characteristics of the video data and the H.264 encoder settings. To describe these dependence investigations have been made. The results imply that the two-dimensional correlation coefficient TC_i between successive frames i and $i+1$ is a suitable statistical characteristic. An important H.264 encoder setting is the quantization parameter QP . Hence, the correlation coefficients mean

$$TC_m = \frac{1}{n} \cdot \sum_{i=1}^{n} TC_i \qquad (1)$$

and QP can be used to estimate the averaged number of skipped macroblocks per frame.

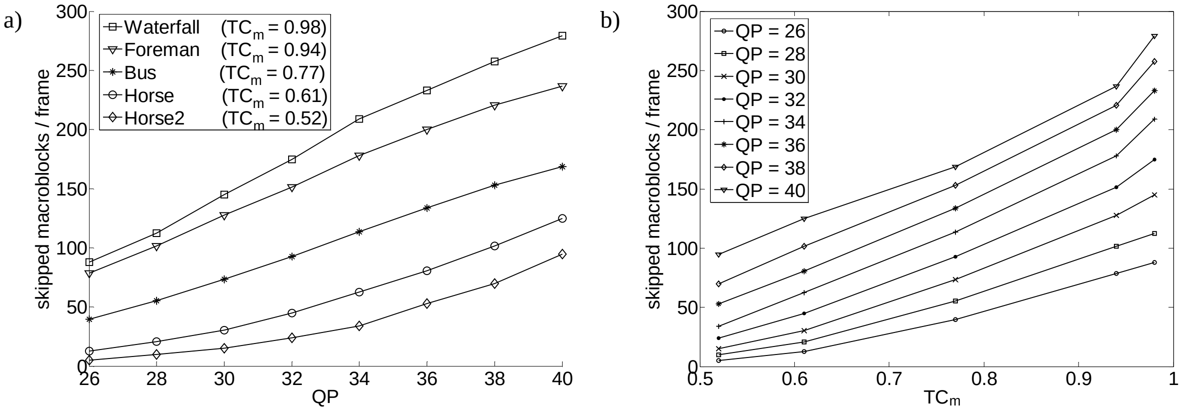

Figure 4. averaged skipped macroblocks per frame in dependence on QP a) and TC_m b)

4. MACROBLOCK SELECTION PROCESS

Generally, the watermarking results in video quality degradation and an increased video data rate. Quality and data rate are interdependent and can be varied. Generally, the quality decreases and increases with the data rate. The aim of the selection process is to improve the ratio between quality degradation and data rate. Because the data rate depends on the watermark payload, we use the embedding overhead instead the video data rate as a significant characteristic of our approach. The embedding overhead is the relationship between embedded bits and increased video data rate. For example, an embedding overhead of two and a watermark payload of one bit per frame results in an increased bit rate of two bits per frame.

In the selection process, three parameters are varied. The number and distribution of used macroblocks and the number of embedded information bits per macroblock. These parameters depend on each other. Assume we use a fixed watermark payload rate per frame. Hence, we can use many macroblocks to embed few information bits per block or few macroblocks to embed many information bits per block. Because we have to encode the macroblock type and other macroblock data, every additional macroblock results in an increased embedding overhead. Using many information bits per macroblock results in a higher video quality degradation. The macroblock distribution affects only the quality degradation. It is important to use macroblocks, which are used as few as possible for prediction of other macroblocks.

4.1. Selection Process - the Basis

The selection process is based on the value mb, which describes how much a macroblock is used for the prediction of other macroblocks. To compute mb for each skipped macroblock, the transcoder uses an additional H.264 decoder. The decoder pre-decodes the video and collects information to skipped macroblocks and to the motion vectors of all macroblocks.

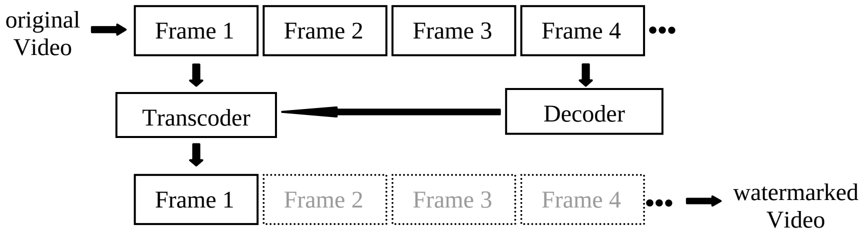

Figure 5. use of a decoder to collect additional information

By using the information of the decoder, the transcoder creates a mb-value-map and a list of all skipped macroblocks. Combining the map and the list, the average mb-value for each skipped macroblock can be computed.

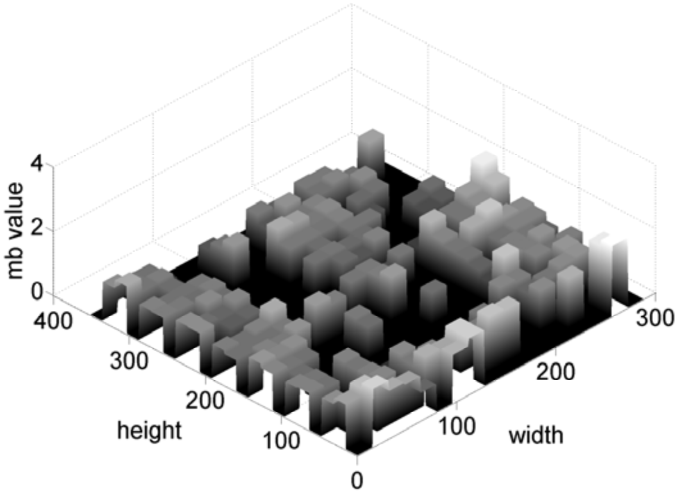

Figure 6. *mb*-value-map of skipped macroblocks

4.2. Future Analysis

The mb-value-map of the skipped macroblocks is used for a future analysis. The macroblocks with the lowest mb-values are used as few as possible for the prediction of other macroblocks. These blocks are selected for watermarking. The mb-value precision depends on the number of pre-decoded frames. For example, see Figure 7. Three skipped macroblocks of every P- and B-Slice are "reactivated" and the 16 luma DC-coefficients of every block are changed by ± 1. We describe the quality degradation with the PSNR between the unwatermarked compressed video and the watermarked compressed video. The PSNR value increases with the number of considered frames. The IDR-I-slice-period[13] is 10. Therefore, after 10 considered frames no improvement appears.

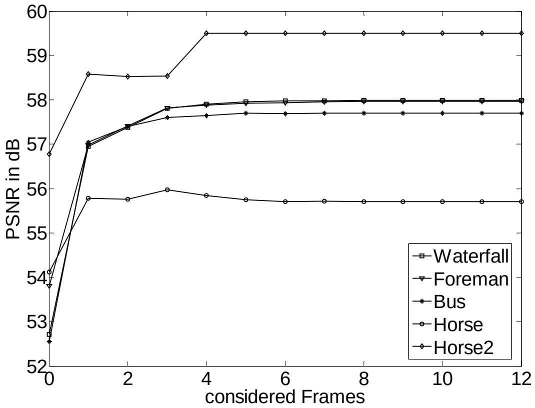

Figure 7. PSNR improvement by selecting skipped macroblocks with a low *mb*-value

4.3. Signal Adaptation

The mb-value not only is used to select the distribution of skipped macroblocks. Additionally, it is used to adapt the signal strength or the number of embedded bits per skipped macroblock. For example, two macroblocks per frame are used to embed four bytes. The two blocks with the lowest mb-value are selected. The mb-value of block one is zero, this block is not used to predict other macroblocks. The mb-value of block two is five, the error of block two propagates five blocks. Therefore, it is better to embed all bytes in block one and do not watermark block two. For example, see Figure 8. Three blocks per frame are watermarked with two bytes per block. As threshold to decide whether one or two blocks are used for four bytes, we use the mb-value difference mbd between two blocks.

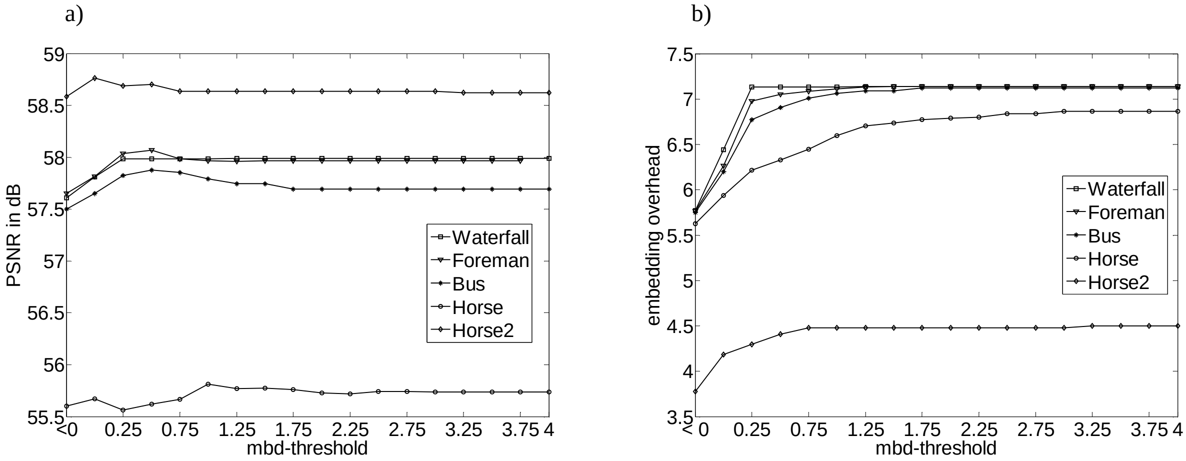

Figure 8. PSNR improvement by signal adaptation in a) and the resulting embedding overhead in b)

Figure 8 a) shows that there is an optimal PSNR value for every video. Figure 8 b) shows that the use of fewer skipped macroblocks results in a lower embedding overhead. Watermarking without signal adaptation results in the worst case concerning the embedding overhead. The use of signal adaptation offers the possibility to set a working point with an optimal relationship between PSNR and embedding overhead. Figure 9 shows an example for the bus video.

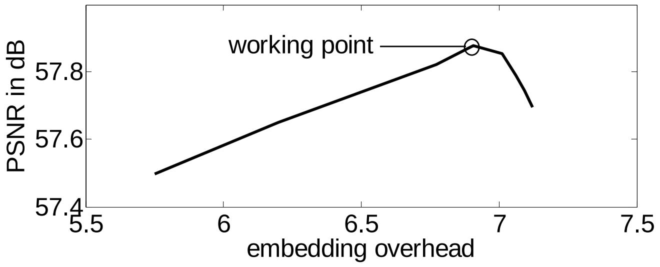

Figure 9. working point for the bus video

5. WATERMARK EMBEDDING

5.1. Watermark Payload

The watermark payload is the authentication data of the video. It consists of an encrypted hash value and a certificate with a public key. To generate the hash value we can use the pure H.264 bit stream or data of slices and macroblocks (H.264 video data). Using the bit stream results in a low effort generating the hash value. But the watermarking changes the bit stream. To reconstruct the hash value the watermark has to be erased. Therefore, a high effort is necessary. Using the H.264 video data results in a high effort generating the hash value. An inverse entropy coding and parsing of the video structure is necessary. During the normal decoding process, the hash value can be reconstructed without erasing the watermark. To generate the hash value we use the H.264 video data. We have a one-time high effort generating the hash value but every user can reconstruct this value with a low effort.

To encrypt the hash value we use a usual certificate system. The hash value is encrypted with a private 2048 bit key. The certificate contains the public key and additional information to verify the key over a trust centre. The size of the certificate depends on the contained information. We assume a certificate size of 3.5 kB. The sum of the encrypted hash value and the certificate is the resulting watermark payload size of 3.756 kB.

An important point is the decision, which video parts are to authenticate. Of course, one way is to authenticate the complete video as shown in Figure 10 a). Another way is to authenticate different video parts. For example, every video scene can be authenticated. The advantage is that after a video manipulation the manipulated scenes can be detected. Therefore, the encrypted hash value and the certificate are embedded in every scene. To authenticate the scene order a second hash value, which cover the complete video, can be embedded. Figure 10 b) shows the principle. Several hash value planes allow a hierarchical authentication of a video.

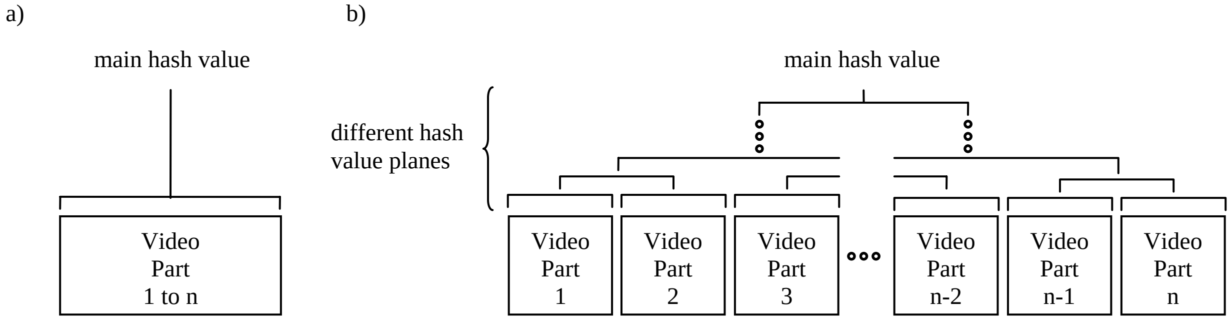

Figure 10. complete authentication of the video data in a) and hierarchical authentication in b)

We authenticate video parts with a length of 30 seconds. This offers the possibility for a hierarchical authentication approach of larger videos. With a size of 3.756 kB for one authentication data set and 25 frames per second we have to embed at least 5.008 Bytes per frame. By using 3 skipped macroblocks with two bytes per block and an I-frame period of ten we can embed 5.4 Bytes per frame. The following investigation is based on this watermark payload rate per frame.

5.2. Embedding Domain and Detection

The macroblock selection process chooses the skipped macroblocks and sets the number of embedded bytes per macroblock. One macroblock contains some metadata (type, coded-block pattern, etc.), the motion vector prediction errors and the coefficients prediction errors. Using the metadata to embed the payload, the watermark capacities are very limited. Using the motion vector prediction error to embed the payload, the motion vector prediction of the neighbour macroblocks is affected. The result is a heavy distortion, which propagate the complete frame. Hence, we use the coefficient prediction error to embed the watermark payload. The error propagation is considered and minimized by the macroblock selection process. To embed the payload we use the luma coefficient prediction errors. Therefore, 16x16 coefficients divided in 4x4 blocks are available as shown in Figure 11. One 4x4 block contains 16 coefficient prediction errors in a zig-zag order. Because we use "reactivated" skipped macroblocks, all coefficient prediction errors are zero. By using one coefficient prediction error per 4x4 block two bytes can be embedded per macroblock.

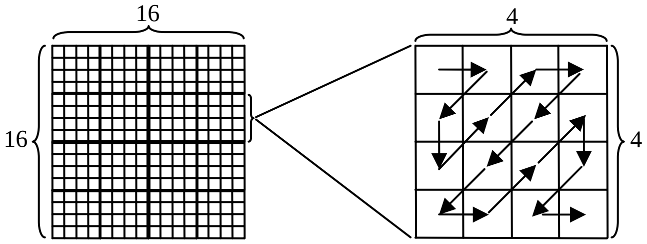

Figure 11. 16x16 block of coefficients consists of 16 4x4 separate transformed blocks

To recover the watermark payload the watermarked macroblocks are to detect. There are different ways to realize this. One way is the embedding of a special pattern. The coefficients are correlated with the pattern to decide whether or not a macroblock is watermarked. This way is inaccurate. Non-watermarked blocks can accidentally contain the pattern, which results in a false detection. To detect the watermarked blocks we use the coded-block pattern. The coded-block pattern defines, in which of the four 8x8 luma and two 8x8 chroma blocks the coefficient prediction errors are unequal to zero. We don't use the chroma coefficients to embed the watermark payload. Hence, all coefficients are zero. By setting the coded-block pattern for one 8x8 chroma block we mark the macroblock. To detect a watermarked macroblock only a comparison of the coded-block pattern and the coefficient prediction errors of this 8x8 chroma block is necessary.

5.3. Embedding Techniques

Generally, there are several approaches to embed the watermark in the selected skipped macroblocks, for instance using *Spread Spectrum*[10], *Last Significant Bit* (LSB)[11] or *Quantization Index Modulation* (QIM)[12]. Often, watermarking approaches are designed to be robust to several attacks like noise, compression, geometrical distortions etc. In our approach, we do not need robust watermarking. Hence, we do not apply these approaches directly but adopt it for our approach.

Spread Spectrum watermarking can be used in spatial and in frequency domain. The watermark bits are spread and modulated with a pseudo-noise signal. The result is added to the original image. To recover the watermark bits, the pseudo-noise signal is required. The principle of *Spread Spectrum* is to achieve robustness by spreading a signal with high energy. For our approach of a fragile watermarking, using *Spread Spectrum* results in disadvantages. We do not need robustness and using a spread signal results in deteriorate video quality and an increased embedding overhead. Generally, LSB watermarking replaces the last significant bit layer of pixel or coefficient values with the watermarking payload. In our approach, we replace the last significant bit of the transmitted skipped macroblock coefficient prediction errors. Attention should be paid, that the coefficients can be negative and positive values. Because of using skipped macroblocks, the coefficient prediction errors are zero. To embed a bit value "0" the coefficient prediction error is left zero. To embed a bit value "1" the coefficient prediction error is set to one. Generally, QIM watermarking is used to achieve high watermark robustness to several attacks. The quantization step size defines this robustness. Because we aim hard authentication, robustness is not needed. Therefore, we use the smallest quantization step size of two. By changing the coefficient prediction errors we quantize the resulting coefficient. To embed a bit value "0" the coefficient prediction error is changed to achieve an even valued coefficient. To embed a bit value "1" the coefficient prediction error is changed to achieve an odd valued coefficient. The absolute change per coefficient prediction error can be zero or one. To achieve a mean offset of zero the sign of the absolute change of one is rotated. To compare the LSB and the QIM approach, we use different frequency of the coefficient prediction errors. We vary them from the lowest to the highest frequency. The coefficient prediction errors are ordered in a zig-zag-line as shown in Figure 11. The investigation delivers following results:

The embedding overhead depends on the relationship between the number of one and zero bits in the watermark payload. A relationship of 1:1 results in an equal embedding overhead for the LSB and QIM approaches. More zeros than ones result in a lower embedding overhead for the LSB approach and more ones than zeros result in a higher embedding overhead for the LSB approach. The embedding overhead also depends on the used frequency of the coefficients. Higher frequency results in a higher embedding overhead as shown in Figure 12 b). Reason is the entropy coding. Using the high-frequency coefficient prediction errors requires entropy coding of the low-frequency coefficient prediction errors even though all are zero.

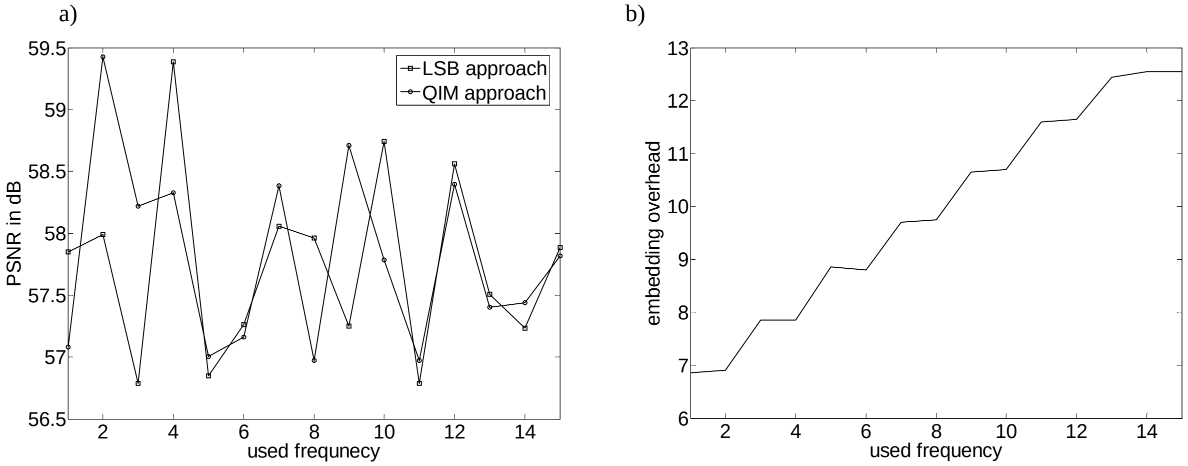

Figure 12. resulting Bus video PSNR for the LSB and QIM approach in dependence on the used frequency in a) and the resulting embedding overhead for both approaches in b)

Figure 12 a) shows the dependence of the PSNR from the frequency. The non-uniform PSNR curve is caused by the different quantization of the single frequency coefficients[13]. Results of investigations show that the optimal PSNR values are achieved at the fourth frequency for the LSB approach and at the second frequency for the QIM approach.

Table 1. PSNR values by using optimal frequency for the LSB and the QIM approach

Video	**LSB-PSNR**	**QIM-PSNR**
Waterfall	59.27 dB	59.01 dB
Foreman	60.23 dB	60.43 dB
Bus	59.37 dB	59.42 dB
Horse	57.84 dB	57.81 dB
Horse2	60.09 dB	60.29 dB

Table 1 shows that there are no relevant differences between the PSNR values of the LSB and QIM approaches. However, the QIM approach achieves the best PSNR values at a lower frequency, which results in lower embedding overhead for the QIM approach as opposed to the LSB approach. Hence, we can say that the QIM approach is suited than the LSB approach for embedding the watermark payload in the skipped macroblocks.

6. CONCLUSION

We propose a new watermarking framework, suitable for authentication of H.264 videos. The use of watermarking to authenticate digital video data and the necessity of hard authentication are discussed. Our approach is based on the latest video compression standard H.264. An essential advantage of our approach is the possibility to erase the watermark and to reconstruct the original H.264 video. The watermark is embedded by “reactivating” some of the skipped macroblocks of the H.264 video data. Number and distribution of skipped macroblocks per frame depends on the statistical characteristics of the video data and the H.264 encoder settings. We present results of investigations to describe these dependences. A special macroblock selection process chooses several skipped macroblocks out of a frame and sets the number of bits, which are to embed. The basis of the selection process and the single sub-processes future analysis and signal adaptation are explained. The watermark payload consists of a cryptosystem certificate and an encrypted hash value. To embed the payload, the luma coefficient prediction errors of the “reactivated” skipped macroblocks are used. By modifying the coded-block pattern, the watermarked macroblocks can be detected. At last, we discuss and investigate several basic watermarking approaches to embed the payload in the luma coefficient prediction errors. We found out that the *Spread Spectrum* approach is an unsuitable approach. The comparison of the LSB and the QIM approach shows that the QIM approach achieves better results than the LSB approach.

REFERENCES

1. I. J. Cox and M. L. Miller, “The First 50 Years of Electronic Watermarking”, EURASIP J. of Applied Signal Processing, No. 2, 126-132, 2002

2. E. T. Lin, C. I. Podilchuk, E. J. Delp, “Detection of Image Alterations Using Semi-Fragile Watermarks”, Proc. of the SPIE, vol. 3971, pp. 152-163, San Jose, CA, 2000

3. R. B. Wolfgang, E. J. Delp, “Fragile Watermarking Using the VW2D Watermark”, Proc. of the IS&T/SPIE, pp. 204-213, San Jose, California, 1999

4. R. L Rivest, “The MD5 Message Digest Algorithm”, RFC 1321, Internet Activities Board, Internet PrivacZ Task Force, 3RIPEMD-1281, April 1992

5. R. L. Rivest, A. Shamir, L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Communications of the ACM, Vol.21, Nr.2, pp.120-126, Feb. 1978

6. T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264 / AVC Video Coding Standard”, IEEE Transactions, Vol. 13, pp. 560-576, July 2003

7. C. Serdean, M. Ambroze, M. Tomlinson, G. Wade, “DWT Based Video Watermarking for Copyright Protection”, *Invariant to Geometrical Attacks*, CSNDSP 2002, Staffordshire, UK, 15-17 July 2002

8. G. Qui, P. Marziliano, A. T.S. Ho, D. He, Q. Sun, “A hybrid watermarking scheme for H.264/AVC video”, Proc. of the ICPR, Vol. 4, pp. 865-868, Cambridge, UK, August 2004

9. Sung-Cheal Byun, Sang-Kwang Lee, Ahmed H. Tewfik, Byung-Ha Ahn, “A SVD-Based Fragile Watermarking Scheme for Image Authentication”, IWDW 2002, 170-178, Seoul, Korea, 2002

10. G. Le Guelvouit, S. Pateux, “Wide spread spectrum watermarking with side information and interference cancellation”, Proc. SPIE/ IS&T Volume 5020, pp. 278-289, Santa Clara, CA, June 2003

11. C. Rajaratnam, N. Memon, “Analysis of LSB-based Image Steganography techniques”, Proc. of International Conference on Image Processing, Vol. 3, pp. 1019-1022, Thessaloniki, Greece, October 2001

12. B. Chen, G. W. Wornell, “Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding”, IEEE Transaction on Info. Theory, Vol. 47, no. 4, pp. 1423-1443, 2001

13. “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14486-10 AVC”, in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2003