Multimed Tools Appl (2006) 31: 145–170
DOI 10.1007/s11042-006-0041-3

Mask-based fingerprinting scheme for digital video broadcasting

Sabu Emmanuel · Mohan S. Kankanhalli

Published online: 6 October 2006
© Springer Science + Business Media, LLC 2006

Abstract In this paper we propose a novel method to achieve video fingerprinting and confidentiality in a broadcasting environment. The fingerprinting technique can be used to generate unique copies for individual subscribers and can be used to identify the copyright violator. Thus for tracing the copyright violator, unique copy per subscriber is needed whereas broadcasting requires a single copy to be transmitted to everyone. The proposed method efficiently incorporates both these requirements. In addition to the fingerprinting requirement to trace the subscriber who is violating the copyright, a confidentiality requirement needs to be implemented against the non-subscribers in the broadcast region. The proposed algorithm efficiently combines both the fingerprinting requirement and confidentiality requirement into one single atomic process. The proposed algorithm uses robust invisible watermarking technique for fingerprinting and masking technique for confidentiality. The additional advantage of the proposed scheme is that it also supports MPEG-2 compressed domain processing, which is useful for many broadcasting standards.

Keywords Copyright protection · Digital watermarks · Digital video broadcasting · MPEG-2 · Pay TV · Video on demand · Video security · Video watermarking

1 Introduction

Video broadcasting systems employ digital techniques for processing, storing and transmission of video data. This is primarily due to ease of handling these functions in

S. Emmanuel (✉)
School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore 639798, Singapore
e-mail: asemmanuel@ntu.edu.sg

M. S. Kankanhalli
School of Computing, National University of Singapore,
Kent Ridge, Singapore 117543, Singapore
e-mail: mohan@comp.nus.edu.sg

digital domain than in analog domain. Being digital also brings in the advantage of easy and perfect replication of digital data. One of the major concerns of broadcasters is that, subscribers can easily make perfect copies of digital video and distribute to non-subscribers thus violating the copyright. Hence in digital video broadcasts, especially in pay channels, fingerprinting techniques are to be employed to identify the subscriber who is violating the copyright. For a broadcast type of transmission, everyone in the broadcast region receives broadcast data. In pay channels of digital video broadcasts, the broadcasters require that only subscribers should be able to view the video clearly. Hence a confidentiality requirement should be implemented against non-subscribers. The existing digital video pay channels employ conditional access system (CAS) for achieving confidentiality against non-subscribers [12, 23, 34]. Our proposed scheme supports both the confidentiality requirement and the copyright violator identification requirement (fingerprinting requirement). In our scheme, the copyright violator identification is made possible through the use of digital watermarking technique and confidentiality against no-subscribers is obtained through the use of the masking technique.

Several digital watermarking techniques have been devised to address the copyright concerns of the content owners, broadcasters and sellers. They are devised for variety of media viz., text, digital audio, digital image and digital video. Various copyright concerns include copyright violation detection/deterrence, copy protection, data authentication and data tamper proofing. A digital watermark is an embedded piece of information either visible or invisible (audible or inaudible, in the case of audio). In the case of visible watermarking, [2, 29, 33] the watermarks are embedded in a way that is perceptible to a human viewer. And hence the watermarks convey an immediate claim of ownership, providing credit to the owner and also deter copyright violations. In the case of invisible watermarking, [9, 13, 18] the watermarks are embedded in an imperceptible manner. The invisible watermarks can be fragile or robust. Fragile invisible watermarks [30] attempt to achieve data integrity (tamper proofing). The fragile invisible watermarks must be invisible to human observers, altered by the application of most common image processing techniques and should be able to be quickly extracted by authorized persons. The extracted watermark indicates where the alterations have taken place. However robust invisible watermarks attempt to achieve copyright violation detection. The desired properties for robust invisible watermarks are that they must be invisible to a human observer, the watermark should be detectable/extractable by an authorized person even after the media object is subjected to common signal processing techniques or after digital to analog and analog to digital conversions. The watermark must be robust against attacks and should resolve the rightful ownership problem (for that watermark must be non-invertible [11]). Many watermarking algorithms have been proposed in the literature [10, 14, 19, 28, 31, 36, 39, 40]. Thus for the purpose of fingerprinting, robust invisible watermarking techniques are suitable.

Digital video broadcasting uses broadcast type of transmission and therefore a single copy of video material is transmitted to everyone in the broadcast region. The broadcast region consists of subscribers and non-subscribers. The pay channels of digital video broadcasts require that a confidentiality requirement be implemented against the non-subscribers. Current pay channels employ CAS for this purpose. Primarily the CAS uses scrambling technique for providing the confidentiality [12, 23, 29, 42]. The broadcaster scrambles the video data using a control word (CW) and broadcasts the scrambled video data. The subscribers use the same CW to descramble the received scrambled video data to obtain clear video. The current CAS does not implement any fingerprinting technique to trace the copyright violator. Since the broadcasting requires a single copy to be transmitted, and the

copyright violator identification requires individual subscribers copy be unique (containing different watermarks), the watermarking for copyright violator identification should be performed at each subscriber end. This means that the current CASs, at the subscriber end has to implement the watermarking/fingerprinting process for copyright violator identification in addition to the descrambling process. It is more secure if the watermarking and descrambling processes are combined into a single atomic process. But it is hard to combine the current descrambling process with watermarking process into a single atomic process. Implementing descrambling and watermarking processes as two separate processes in a single IC chip is not secure as the control information to control the watermarking process can be selectively removed from the broadcast video stream while retaining the control information to control the descrambling process. Thus, one can obtain an unwatermarked clear video for viewing. Our proposed scheme uses masking technique for confidentiality and robust invisible watermarking for copyright violator identification. The masking and watermarking techniques are combined into a single atomic process, making it more secure against attacks.

It is even more challenging to support confidentiality and copyright violator identification in compressed (MPEG-2) domain broadcast. The challenges are due to the quantization, which is lossy and interframe coding, which makes use of the motion compensated predictions. However compressed domain processing for compressed domain broadcast is necessary for the following reasons. Firstly, the decompression, processing and recompression will have more computational overhead. Secondly, the MPEG-2 being lossy compression technique it would degrade the video due to the recompression. Our proposed scheme supports MPEG-2 compressed domain processing. The MPEG-2 intellectual property management and protection (IPMP) standard [20, 24–26] provides place holders for sending the information whether the packet is scrambled, the information about which CAS is used, control messages such as encryption control message (ECM) and encryption management message (EMM), and copyright identifiers. The IPMP specification only addresses confidentiality and watermarking. But the IPMP-X (intellectual property management and protection-extension) includes authentication function also along with the confidentiality and watermarking functions. The standard however does not specify the algorithms to be used for confidentiality, watermarking or authentication.

We in this paper describe a novel uncompressed (spatial) and compressed domain (MPEG-2) method to achieve the requirements—confidentiality against the non-subscribers, protection against the copyright violations, single and one time created copy for transmission for a broadcast scenario. Confidentiality is obtained by using an opaque blending mask while copyright violator identification is ensured by robust invisible watermarking. The proposed method efficiently combines the confidentiality requirement and copyright violator identification requirement (fingerprinting requirement) into a single atomic process. The method also supports dynamic join and leave. It requires less computational and network resources from the broadcaster yet it is easy to implement.

In Section 2 we discuss the past work in the area of conditional access systems and broadcast video watermarking. In Section 3 we describe the proposed scheme, in Section 4 implementation and results, in Section 5 discussion followed by conclusion in Section 6. A preliminary version of this paper appeared in [15].

2 Past work

In this section we discuss the past work related to the conditional access systems and digital watermarking of broadcast video.

2.1 Conditional access system (CAS)

A conditional access system (CAS) is the essential system to facilitate charging the subscriber some subscription fee. Existing conditional access TV modes are:

- pay TV (operated in the subscription mode)
- pay per view (payment for a single program feature as desired, which could be pre-booked or impulsive)
- pay per view per time (whereby the viewer's charges will be a function of the time spent on the channel).

The pay TV CAS already in existence comprises of two subsystems [12, 23, 29]. One subsystem implements the scrambling/descrambling system and the other subsystem implements the access control system. A scrambling system renders the basic service content i.e., audio and video useless for an unauthorized receiver. The scrambling system scrambles the basic service content, by suitably modifying the digital video data, or altering the audio data through digital processing with the help of a control word (*CW*). Several algorithms are available for scrambling purpose such as the data encryption standard (DES) and the digital video broadcasting common scrambling algorithm (DVB-CSA) [8, 23, 34]. The DES is widely used by the companies in USA. European legislation mandates the use of DVB-CSA in all digital TVs in Europe. The standards adopted for digital TV systems use MPEG-2 for the source coding. The transmission standard used in Europe is digital video broadcasting (DVB) standard, whereas in USA it is advanced television system committee (ATSC) standard. In DVB the scrambling can be carried out on the payload of MPEG-2 transport stream (TS) packet or on the payload of MPEG-2 packetized elementary stream (PES) packet. ATSC supports scrambling, only on the payload of MPEG-2 TS packets. The DVB supports the simulcrypt and multicrypt standards whereas ATSC supports only simulcrypt standard [8, 23]. The simulcrypt standard allows the co-existence of more than one CASs, simultaneously addressing different consumer bases, in one transmission. This is possible by using the same common scrambling algorithm with same *CW* by all CASs. Each CAS constructs their own entitlement control message (ECM) containing the description about the program and the encrypted *CW*. Each CAS encrypts the *CW* using their own service key (*SK*). The ECMs of each participating CAS are then multiplexed along with the MPEG-2 stream. The subscribers would use their respective CAS's *SK* to obtain the *CW* from the respective CAS's ECM. The *CW* is then used to descramble the received scrambled video to obtain the clear video. In case of multicrypt, no program is available through more than one CA system.

The entitlement management messages (EMM) are used to convey new entitlements, or service keys (*SK*s) to every subscriber. These EMMs are encrypted with a programmer distribution key (*PDK*). The EMMs can be transmitted along with the MPEG-2 stream or through a separate channel. The *SK*s are specific to each conditional access system whereas *PDK*s are specific to each subscriber. The *PDK*s are distributed to each subscriber by the program provider/broadcaster. This *PDK* can be transmitted over the same transmission channel by encrypting it using a key called issuer key (*IK*) or by an already available programmer distribution key. The issuer key *IK* is never transmitted on the transmission channel, but is directly distributed to the subscribers during the initialization phase. The *IK* is used to load/invalidate the *PDK* and *SK*. In fact these keys *IK*, *PDK* & *SK* along with subscribers entitlements are to be stored in a processor in the subscriber set top box. Therefore the processor used should be a secure processor [32].

The subscriber set top boxes can be considered as consisting of a host module and a CAS module. The host module typically contains an MPEG decoder, a tuner/demodulator for signal reception and input/output section. The conditional access module implements the descrambling function and management of various keys (*IK*, *PDK* & *SK*), ECM and EMM. The CAS module can be implemented in a removable secure processor such as a PCMCIA card or on a smart card [6, 12].

The MPEG-2 standard supports control information for intellectual property management and protection (IPMP), which includes conditional access and copyright management [20, 24–26]. The control information is handed over through IPMP elementary streams (IPMP-ESs) and IPMP descriptors (IPMP-Ds). But these IPMP messages are not made used by current conditional access systems for copyright management. The copyright management function is envisaged to be implemented along side the MPEG-2 decoder on the host module. Since the host module is not part of the removable secure processor we argue that the copyright management function can be turned off by removing the control bits belonging to the copyright management. Removal of these control bits is particularly easy due to the fact that the input to the MPEG-2 decoder and copyright management circuit are descrambled streams. Thus implementing descrambling and copyright management processes (watermarking process) as two processes is not secure. It is noted that any watermark bits embedded for copyright management before MPEG-2 decoding can be thought of as occurred errors in the MPEG-2 stream. These errors can cause drift problems while decoding. However our proposed algorithm combines watermarking and descrambling process into a single atomic process and performs unmasking and watermarking after MPEG-2 decoding. Therefore our proposed scheme is more secure and exhibits no drift problem. We next discuss the past work in the area of digital watermarking of broadcast video.

2.2 Digital watermarking of broadcast video

Techniques for hiding watermarks in digital data have grown steadily more sophisticated and increasingly robust against attacks. Many video researchers have used them to provide copyright management for video.

The European Esprit VIVA project [38] uses the watermarking technique for broadcast monitoring. The broadcast materials are watermarked in the spatial domain prior to broadcasting and the watermark is detected using the correlation detector. The broadcast chain consists of, D/A conversion, A/D conversion, MPEG-2 compression, MPEG-2 decompression, D/A conversion and A/D conversion. The watermark can be detected even though the watermarked video undergoes all these processing. This can be used for verification of commercial transmissions, assessment of sponsorship effectiveness, statistical data collection and analysis of broadcast content. But this scheme does not support individual watermarking for copyright violator identification in a broadcasting environment also cannot be used for subscription based video broadcasts where a confidentiality requirement is required against the non-subscribers.

Anderson & Manifavas have proposed the "Chameleon" scheme [1] that allows a single broadcast ciphertext to be decrypted to be slightly different plaintexts by users with lightly different keys. As acknowledged by the authors, the watermarking capability of this scheme is rather limited for MPEG video. This is because the encryption is done after MPEG encoding and decryption is done before decoding. The chameleon decryption leaves behind a watermark, which is some bit changes (equivalent to bit errors). The bit error rate lower than 0.1% is required for acceptable viewing quality. Since the watermark bits are very few, the number of distinct watermarks are also few. This affects scalability for broadcasting.

Our proposal does masking in compressed domain but unmasks leaving behind a watermark after MPEG decoding. Therefore the watermarking can be performed to the just noticeable distortion (JND) level of perceptual quality of the video.

Brown, Perkins & Crowcroft propose the "Watercasting" technique [4] that has each receiver in a multicast group receive a slightly different version of the multicast data. This scheme requires that the source watermark, encrypt and transmit n copies of the data. The network bandwidth requirement is high as the source transmits n copies. Each sender must trust the chain of network routers. A chain of trusted network providers is required. Each of them has to be willing to reveal their tree topology to each sender. It also does not offer a solution to distinguish the copies of receivers on the same subnet. Our scheme requires only one masked copy. Therefore the resource requirements at the source as well as the network are less compared to the above case. Our method does not ascribe any active role to the network routers and can distinguish every receiver.

Chu, Qiao & Nahrstedt presents a secure multicast protocol with copyright protection [7]. The protocol creates two watermarked streams, assigns a unique random binary sequence to each user and uses this sequence to arbitrate between the two watermarked streams. The efficiency is hampered by the need to watermark, encrypt and transmit two copies of the stream and by the significant amount of key message traffic. Also the authors state that it may be susceptible to collusion attacks.

Briscoe & Fairman present [3] a number of modular mechanisms "Nark" to enable secure sessions tailored to each individual multicast receivers. In addition to security it also proposes solution for non-repudiation and copyright protection essentially using the Chameleon scheme. Other than the limitations of Chameleon, it also requires a tamper resistant processor at each receiver.

Judge & Ammar propose the "WHIM" scheme [27], which makes use of a hierarchy of intermediaries for creating and embedding watermark. This scheme suffers from low watermark embeddability problem. Also each sender must trust the chain of active network intermediaries and network providers. Since the scheme does not combine the watermarking and decryption process at the receiver in one single process, the watermarking process can be bypassed.

The method presented by Parviainen and Parnes [35] creates two distinctly watermarked copies of each media packet. Both copies are then encrypted with two different randomly generated encryption keys and are then broadcast/multicasted. Any given receiver has access to the key of only one of the two encrypted packets of one media packet. For a media with k packets the method requires $2k$ keys and any one receiver possesses k keys. But this scheme has only limited collusion resistance as acknowledged by the authors.

In Table 1 we summarize the above discussed past works in digital broadcast video watermarking area.

This paper is an extension of our earlier work presented in [15] and [16]. The main contributions in this paper are proofs for compressed domain scaling and mask blending. It also presents a quantitative measure of degradation, computation overhead and compression overhead of the proposed algorithm.

We next discuss our proposed scheme.

3 The proposed scheme

The proposed scheme intends to provide copyright violator identification and confidentiality in a broadcasting environment. The scheme supports spatial (uncompressed) and compressed domain processing. We briefly describe our proposed scheme first.

Table 1 Digital broadcast video watermarking techniques and comparisons

	Confidentiality	Copyright violator identification	Supports MPEG-2 compression	Remarks
VIVA	No	No	Yes	– For broadcast monitoring
Chameleon	Yes	Yes	Drift problem	– Low watermark embedding capacity
Watercasting	Yes	Yes	Yes	– Active role by network routers needed – Cannot distinguish the copies of receivers on the same subnet
Chu et al.	Yes	Yes	Yes	– Susceptible to collusion attacks
Nark	Yes	Yes	Drift problem	– Low watermark embedding capacity – Tamper resistant processor needed
WHIM	Yes	Yes	Yes	– Low watermark embedding capacity – Active intermediaries required – Watermarking and decryption processes are not combined into one single process
Parviainen et al.	Yes	Yes	Yes	– Susceptible to collusion attacks
Proposed Algorithm	Yes	Yes	Yes	– High watermark embedding capacity – No active role to any intermediaries/ routers – Fingerprints every copy – Not susceptible to collusion attacks – No tamper resistance processor needed – Watermarking and decryption processes are combined into one single process

Brief description The broadcaster first creates a masked video by blending/embedding an opaque mask frame on to the original uncompressed video/compressed video, frame by frame. Mask blending process serves the purpose of confidentiality. The masked video is then broadcast. The subscribers unmask the received masked video using an unmasking frame (customized for each subscriber) leaving behind a residue in the form of a robust invisible watermark in the unmasked video. In addition to removing the masking effect, the unmasking process carries out the watermarking for copyright violator identification. The masking process is done in the transform (compressed) domain at the encoder for x the compressed domain processing and for spatial (uncompressed) domain the masking is performed in the spatial domain itself. The unmasking process is done in the spatial domain for both compressed and uncompressed domain processing by the decoder in the subscriber set-top boxes. The proposed scheme is depicted in figure 1, In figure 1, $\boldsymbol{x}_n^m$ is the nth masked video frame, $\boldsymbol{x}_n^{w_a}$ is the nth watermarked video frame of subscriber A, $\boldsymbol{x}_n^{w_b}$ is the nth watermarked video frame of subscriber B, $\boldsymbol{v}_a$ is the unmasking frame for subscriber A and $\boldsymbol{v}_b$ is the unmasking frame for subscriber B. We will now explain the method in detail.

3.1 Confidentiality requirement

The confidentiality requirement is intended to force the non-subscribers to join the broadcast and is obtained through the use of a mask blending/embedding procedure, which

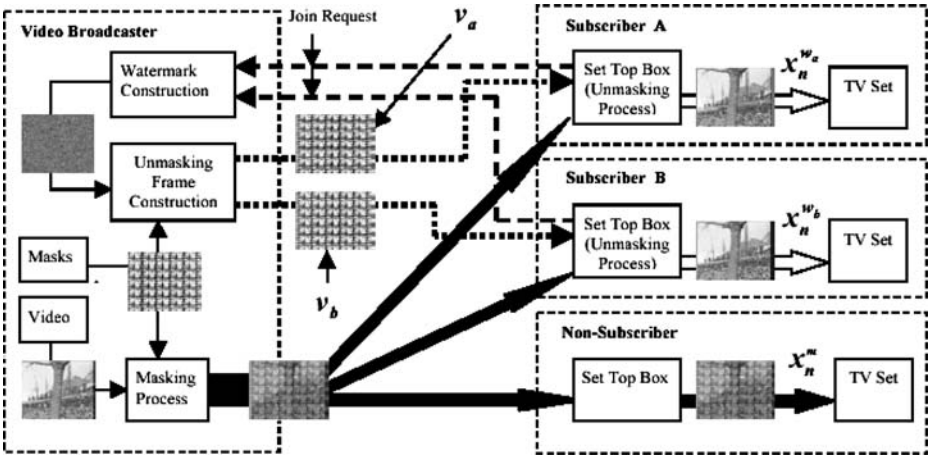

Fig. 1 The proposed scheme

can be performed in spatial domain or in compressed domain. The masked video is created only once and is then broadcast over air or network. We next explain the spatial domain mask blending followed by compressed domain mask blending.

3.1.1 Spatial domain mask blending process

A video is a set of $K \times L$ sized frames whose nth frame is denoted by $\boldsymbol{x}_n$. Next the broadcaster constructs a $K \times L$ sized mask frame $\boldsymbol{v}$. The opaque mask frame's purpose is to severely degrade the viewing experience by obscuring the video. The intended effect is similar to that of video scrambling. The mask frame $\boldsymbol{v}$ is blended onto every frame of the video:

$$\boldsymbol{x}_n^m(k,l) = \alpha x_n(k,l) + \beta \boldsymbol{v}(k,l) \quad \forall\ k,l \tag{1}$$

where $\boldsymbol{x}_n^m$ is the nth masked video frame and α, β are scaling factors such that $\alpha + \beta = 1$ and $0 < \alpha, \beta \leq 1$. The scaling factors can be used to adjust the strength of the mask. The Eq. 1 defines the mask blending process in the spatial domain. The masked video is then broadcast. The receivers who are non-subscribers would only be able to view $\boldsymbol{x}_n^m$, which is obscured. The requirement here is that the output of the decoder/set top box to the display should be $\boldsymbol{x}_n^m$ for non-subscribers. Next we explain how we meet this requirement in MPEG-2 compressed domain.

3.1.2 Compressed domain processing

In our scheme we assume that the MPEG-2 compressed video stream is available for broadcasting. The decoder/set top box at the receiver decompresses the received MPEG-2 data before sending to the display. For a non-subscriber the data send to the display must be $\boldsymbol{x}_n^m$ the obscured video.

This can be achieved by appropriately processing the video in the compressed domain at the encoder before broadcasting. The obscured output at the non-subscriber's decoder would be given by the Eq. 1 $\boldsymbol{x}_n^m(k,l) = \alpha \hat{x}_n(k,l) + \beta \boldsymbol{v}(k,l)\ \forall\ k,l$. The term $\hat{\boldsymbol{x}}_n$ is used instead of $\boldsymbol{x}_n$ to reflect the loss caused by MPEG-2 compression. We first show how $\alpha \hat{\boldsymbol{x}}_n$ is

obtained by processing the quantized error Discrete Cosine Transform (DCT) coefficients and then the addition of $\beta \boldsymbol{v}$ to $\alpha \hat{\boldsymbol{x}}_n$ is shown. The figure 2 depicts the compressed domain mask blending process. Acronyms in figures 2 and 3 and their expansions are as follows: VLC—variable length code, SW—switch, MC—motion compensation, FDCT—forward discrete cosine transform, IDCT—inverse discrete cosine transform, DC DCT—DC coefficient of discrete cosine transform [22].

We use the following *notation* in this section: $\phi[\{\langle a;b\rangle\}]$ implies that, $\langle a;b\rangle$ refers to an 8×8 pixel block, where "a" is the DC DCT coefficient of the 8×8 pixel block, "b" represents the 63 AC DCT coefficients of the 8×8 pixel block, $\{\langle a;b\rangle\}$ refers to the set of all 8×8 pixel blocks consisting a frame and $\phi[\{\langle a;b\rangle\}]$ represents the operator "ϕ" applied on to all the blocks of the frame. $\phi[\{\langle c\rangle\}]$ implies that, $\langle c\rangle$ refers to an 8×8 pixel block, where "c" is the 8×8 pixel block. $\{\langle c\rangle\}$ refers to the set of all 8×8 pixel blocks consisting a frame and $\phi[\{\langle c\rangle\}]$ represents the operator "ϕ" applied on to all the blocks of the frame.

3.1.2.1 Compressed domain scaling of video frames

In this subsection we describe how we compute the $\alpha \hat{\boldsymbol{x}}_n$. We can observe in figure 2 that the MPEG-2 compressed video stream is first passed to the VLC decoder and demultiplexer box which outputs error DCT (discrete cosine transform) coefficients, motion vectors, control parameters. We use these motion vectors and control parameters for scaling the video frames and also for mask blending. The quantized error DCT coefficients are then scaled by a factor α as required by Eq. 1. The box 'SW' in figures 2 & 3 is switch. The next box, which implements 'subtract $\frac{(1-\alpha)*128*N}{s}$ from DC_DCT' where 'N' comes from the $N\times N$ point DCT and here $N=8$, 's' is the intra_DC_differential quantization step-size (a control parameter, which was used during MPEG-2 encoding of video frames), is necessary due to the use of fixed prediction value of 128 for the MB_intra blocks at the decoder for

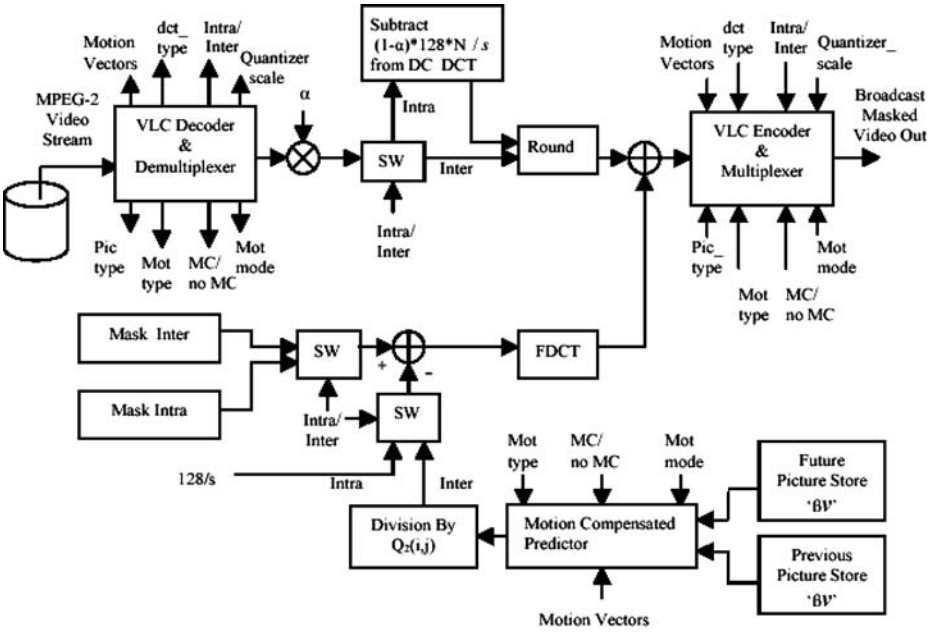

Fig. 2 Compressed domain mask blending process

MPEG-2 video streams. We now prove that this sequence of computation would result in $\alpha\hat{\boldsymbol{x}}_n$.

Proof The input to each box in figures 2 & 3 is frame by frame but each box implements block by block processing on the input frame and the blocks consist of 8×8 pixels. As per the encoding/decoding order the first frame in a group of pictures (GOP) is encoded as an I frame. All the blocks in an I frame are intra coded. Let $E_n(f_1, f_2)\, f_1 = 0,1, ..7\, f_2 = 0,1,..7$ be the quantized error DCT coefficients of a block of nth original video frame, which is encoded as an I frame. The intra coded "MB_intra" blocks use a fixed prediction value of 128 and hence we use the term "error" and the notation $E_n(f_1, f_2)$. ■

Let us assume that we transmit the VLC coded *scaled and shifted* quantized error DC DCT coefficient $\left(\alpha E_n(0,0) - \frac{(1-\alpha)*128*N}{s}\right)$ and *scaled* quantized error AC DCT coefficients $\alpha E_n(f_1, f_2)$ for $f_1 = 0,..7\, f_2 = 0,..7$ except $(f_1, f_2) = (0,0)$ of all the blocks in the I frame block by block i.e., we transmit the VLC code for,

$$\left\{\left\langle\left(\alpha E_n(0,0) - \frac{(1-\alpha)*128*N}{s}\right); \alpha E_n(f_1,f_2) \text{ for } f_1 = 0,..7 f_2 = 0,..7 \text{ except } (f_1,f_2) = (0,0)\right\rangle\right\} \tag{2}$$

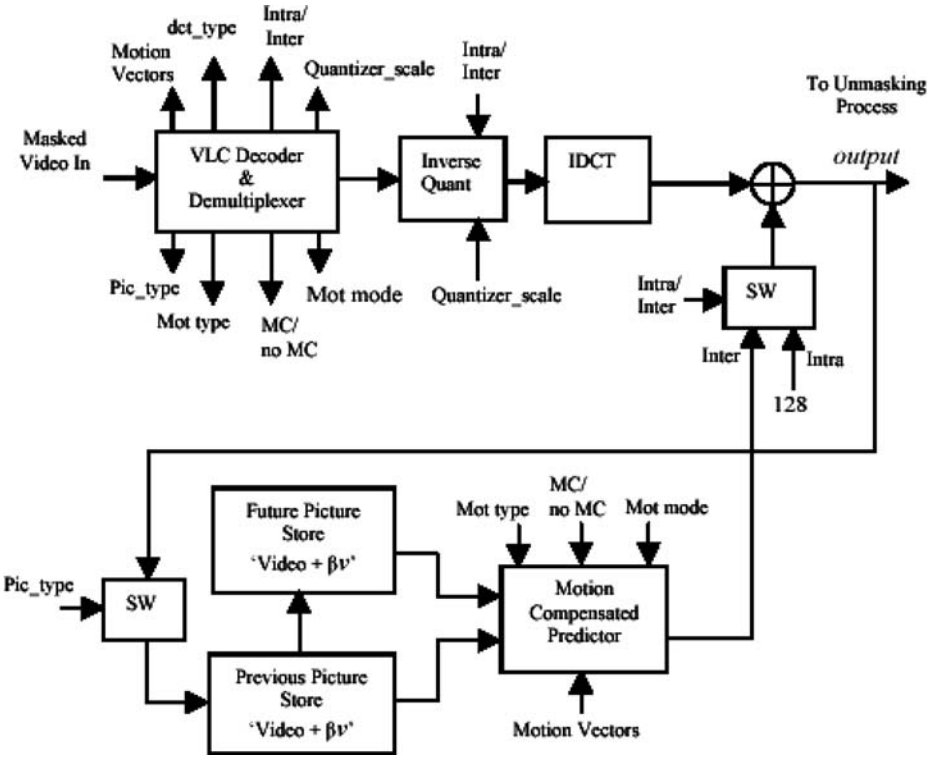

Fig. 3 MPEG-2 decoder

Since VLC and Inverse VLC are lossless we have at the output of the inverse quantizer at the MPEG-2 decoder (figure 3),

$$Q^{-1}\left[\left\{\left\langle\left(\alpha E_n(0,0)-\frac{(1-\alpha)^*128^*N}{s}\right);\alpha E_n(f_1,f_2)\text{ for }f_1=0,..7 f_2=0,..7\text{ except }(f_1,f_2)=(0,0)\right\rangle\right\}\right]$$
$$=\left\{\left\langle\alpha\hat{E}_n(0,0)-(1-\alpha)^*128^*N;\alpha\hat{E}_n(f_1,f_2)\text{ for }f_1=0,..7 f_2=0,..7\text{ except }(f_1,f_2)=(0,0)\right\rangle\right\} \tag{3}$$

where $\hat{E}_n(f_1,f_2)$ is the inverse quantizer output of E_n (f_1,f_2).

After the Inverse DCT (IDCT),

$$IDCT\left[\left\{\left\langle(\alpha\hat{E}_n(0,0)-(1-\alpha)^*128^*N;\alpha\hat{E}_n(f_1,f_2)\text{ for }f_1=0,..7 f_2=0,..7\text{ except }(f_1,f_2)=(0,0)\right\rangle\right\}\right]$$

$$\begin{aligned}&=(\alpha\hat{x}_n(k,l)-\alpha^*128)-(1-\alpha)^*128 \quad \forall\, k,l\\ &=\alpha\hat{x}_n(k,l)-128 \quad \forall\, k,l\end{aligned} \tag{4}$$

The intra coded blocks use a fixed prediction value of 128. The output of the MPEG-2 decoder, after adding the fixed prediction 128 we have,

$$\begin{aligned}\text{output}&=\alpha\hat{x}_n(k,l)-128+128 \quad \forall\, k,l\\ &=\alpha\hat{\boldsymbol{x}}_n\end{aligned} \tag{5}$$

We see that the MPEG-2 decoder output is scaled by a factor α. This scaled version is used as a prediction for the subsequent P and B frames of the current GOP (group of pictures). The above described processing is applied to the intra coded blocks of P and B frames as well to obtain the scaled version.

Let us assume that the following frame is encoded as P frame. Let it be the $(n+1)$th frame. P frames consists of inter/intra coded blocks. The intra coded blocks are processed exactly the same way as described above to obtain the scaled version. But for inter coded blocks we note that there is a scaled version of the I frame at the MPEG-2 decoder for prediction. For inter coded blocks we transmit the VLC coded, *scaled* quantized error (DC and AC) DCT coefficients $\alpha E_{n+1}(f_1, f_2)$ for $f_1=0,1,..7$ $f_2=0,1,..7$. For simplicity of description we assume the P frame consists only of inter coded blocks. At the MPEG-2 decoder, after the inverse quantizer we have,

$$\begin{aligned}&Q^{-1}[\{\langle\alpha E_{n+1}(f_1,f_2)\text{ for }f_1=0,1,..7 f_2=0,1,...7\rangle\}]\\ &=\left\{\left\langle\alpha\,\hat{E}_{n+1}(f_1,f_2)\text{ for }f_1=0,1,..7 f_2=0,1,...7\right\rangle\right\}\end{aligned} \tag{6}$$

where $\hat{E}_{n+1}(f_1,f_2)$ is the inverse quantizer output of $E_{n+1}(f_1,f_2)$.

After the Inverse DCT (IDCT), since $\hat{E}_{n+1}(f_1,f_2)$ is the motion compensated prediction error DCT coefficients, we can write,

$$\begin{aligned}&IDCT\left[\left\{\left\langle\alpha\hat{E}_{n+1}(f_1,f_2)\text{ for }f_1=0,1,..7 f_2=0,1,...7\right\rangle\right\}\right]\\ &=\alpha(\hat{x}_{n+1}(k,l)-\hat{x}_n(k-d_u,l-d_v)) \quad \forall\, k,l\\ &=\alpha\hat{x}_{n+1}(k,l)-\alpha\hat{x}_n(k-d_u,l-d_v) \quad \forall\, k,l\end{aligned} \tag{7}$$

where d_u, d_v are motion vectors.

The output of the MPEG-2 decoder, after adding motion compensated prediction we have,

$$\begin{aligned} \text{output} &= (\alpha\hat{\boldsymbol{x}}_{n+1}(k,l) - \alpha\hat{\boldsymbol{x}}_n(k-d_u, l-d_v)) + \alpha\hat{\boldsymbol{x}}_n(k-d_u, l-d_v) \quad \forall\, k,l \\ &= \alpha\hat{\boldsymbol{x}}_{n+1} \end{aligned} \tag{8}$$

We can see that the output is scaled version of $\hat{\boldsymbol{x}}_{n+1}$. The scaled version of I and P frames are used as a prediction for the next B frames. The B frames can be consisting of inter/intra coded blocks. The intra coded blocks are processed exactly the same way as that of intra coded blocks in I frames to obtain the scaled version of the B frames at the output of the MPEG-2 decoder. Inter coded block would mean the predictions to be forward, backward or interpolated motion compensated predictions. The inter coded blocks are processed exactly the same way as that of inter coded blocks in P frames to obtain the scaled version of the B frames at the output of the MPEG-2 decoder. In case of interpolated motion compensated predictions, there are two motion vectors but it can be easily shown that the method for forward/backward motion compensated prediction applies for this case also.

After multiplying the error DCT coefficients by the scaling factor (if intra coded blocks, the DC is shifted) we round the result to an integer as required by the MPEG-2 as shown in figure 2. The proof above did not take into account the loss due to rounding. This loss is small, which can be observed from figure 5b & d, and is inevitable in any case of additive watermarking (followed by MPEG compression) where scaling is applied before adding the appropriate strength of watermark. Next, we discuss how we obtain the addition of $\beta\boldsymbol{v}$ to the $\alpha\hat{\boldsymbol{x}}_n$ as required.

3.1.2.2 Compressed domain mask blending

The appropriately generated mask error DCT coefficients are then added to the rounded scaled quantized video error DCT coefficients as shown in figure 2 (at the broadcaster site). These values along with the motion vectors, control parameters and other control information are then VLC encoded and transmitted. We use the same motion vectors, control parameters and control information obtained at the output of the VLC decoder and demultiplexer (figure 2) for generating the appropriate mask error DCT coefficients. This means that the same motion vectors, control parameters and control information which were used for video encoding is used for generating the appropriate mask error DCT coefficients. Further, any constraint placed on the control parameters would apply to the video encoding as well as to the mask.

The $\alpha\hat{\boldsymbol{x}}_n$ and $\beta\boldsymbol{v}$ at the output of the MPEG-2 decoder (subscriber site) can be considered as result of separate inputs to the MPEG-2 decoder system. One input (due to video) to the MPEG-2 decoder causes $\alpha\hat{\boldsymbol{x}}_n$ and the other input (due to mask) causes the output $\beta\boldsymbol{v}$. Therefore the masking process at the broadcaster site, which causes $\beta\boldsymbol{v}$ at the output of the MPEG-2 decoder (subscriber site), can be treated separately (and it needs to be shown, how we obtain the presence of $\beta\boldsymbol{v}$ at the output of the MPEG-2 decoder). The input, which causes $\alpha\hat{\boldsymbol{x}}_n$ at the output is decoded properly by the MPEG-2 decoder. The masking process must be done with the aim that the masked video should be obscured and also when the MPEG-2 decoder decodes the masked video, it must result in an output frame $\beta\boldsymbol{v}$ added to every frame at the output of the MPEG-2 decoder as seen in figure 3.

So we begin with a mask frame $\boldsymbol{v}$ (luminance component only), which is then scaled by β to obtain $\beta\boldsymbol{v}$. Further all the processing is done block by block on the frame and the blocks consists of 8×8 pixels. Note that the embedding process is done only for the luminance blocks. From this $\beta\boldsymbol{v}$, we create two scaled versions of this frame, one frame ***mask_i*** which consists of blocks to mask corresponding intra-coded blocks (which have no

motion compensation) of the video and another frame ***mask_n*** which consists of blocks to mask corresponding inter-coded blocks of the video using the following expressions:

$$\boldsymbol{mask\_i} = IDCT\left[\left\{\left\langle \frac{\beta V(0,0)}{s}; \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i=f_1=0,...7\, j=f_2=0,...7 \text{ except } (i,j)=(f_1,f_2)=(0,0)\right\rangle\right\}\right] \tag{9}$$

$$\boldsymbol{mask\_n} = IDCT\left[\left\{\left\langle \frac{\beta V(0,0)}{Q_2(0,0)}; \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i=f_1=0,...7\, j=f_2=0,...7 \text{ except } (i,j)=(f_1,f_2)=(0,0)\right\rangle\right\}\right] \tag{10}$$

Where $\beta V(0,0)$ is DC DCT coefficient of one block of $\beta \boldsymbol{v}$ and $\beta V(f_1, f_2)$ for $f_1 = 0,..7\ f_2 = 0,..7$ except $(f_1, f_2) = (0,0)$ is 63 AC DCT coefficients of the same block of $\beta \boldsymbol{v}$. We assume that the intra and inter quantization matrix values are the same for the AC DCT coefficients i.e., *Intra Qmat*(i,j) = *Inter Qmat* (i,j) for $i = 0,..7\ j = 0,7$ except $(i,j) = (0,0)$ where *Intra Qmat*(i,j) is the intra quantization matrix and *Inter Qmat*(i,j) is the inter quantization matrix. (This assumption can be relaxed if the watermark used is robust. We use robust spread spectrum based watermark.) Therefore,

$$Q_1(i,j) = Q_2(i,j) = Q(i,j) \text{ for } i = 0,..7\, j = 0,..7 \text{ except } (i,j) = (0,0) \tag{11}$$

where,

$$Q_1(i,j) = \frac{2 \times q\_scale \times Intra\ Qmat(i,j)}{32} \text{ for } i = 0,..7\, j = 0,..7 \text{ except } (i,j) = (0,0) \tag{12}$$

$$Q_2(i,j) = \frac{2 \times q\_scale \times Inter\ Qmat(i,j)}{32} \text{ for } i = 0,1,..7\, j = 0,1,..7 \tag{13}$$

hence,

$$Q(i,j) = \frac{2 \times q\_scale \times \{Intra\ or\ Inter\} Qmat(i,j)}{32} \text{ for } i = 0,..7\, j = 0,..7 \text{ except } (i,j) = (0,0) \tag{14}$$

The factor *q_scale* is the quantization scale factor and is assumed to be constant. The intra_DC_Differential_quantization step size 's' can be 2, 4 or 8. And,

$$Q_2(0,0) = \frac{2 \times q\_scale \times Inter\ Qmat(0,0)}{32} \tag{15}$$

For masking an intra type of block at (x,y) location in the video frame, we just need to add the DCT of the prediction error between the block in ***mask_i*** (at the same location x,y) and $\frac{128}{s}$ as can be seen from figure 2. For masking the forward or backward or the interpolated types of block at location (x,y) in the video frame, we just add the DCT of the prediction error between the block in ***mask_n*** at the same location and the motion compensated prediction for the inter coded block. The motion compensated prediction for inter coded block is divided by $Q_2(i,j)$. This will not be lossy as long as 's' is divisible by $Q_2(0,0)$. For the skipped blocks, nothing needs to be added, just the macroblock skip

information is to be transmitted. But for the MB_pattern coded blocks, one has to use the union of the coded block pattern of the video and the masking process.

The above procedure for blending the mask in the compressed domain will result in a constant $\beta \boldsymbol{v}$ frame to be at the output of the MPEG-2 decoder:

Proof We discuss the mask blending process, which causes $\beta \boldsymbol{v}$ at the output of the MPEG-2 decoder. Input to each box in figures 2 & 3 is frame by frame but each box implements the block by block processing on the input frame and the blocks consists of 8×8 pixels. ■

Let us mask an I frame consisting of intra coded blocks. Then the input of the FDCT (forward DCT) in figure 2 is the prediction error frame $\{\langle mask\_i\_b(u,v) - \frac{128}{s} \text{for } u = 0,1,...7 v = 0,1,...7\rangle\}$. Where ***mask_i_b*** is one block of ***mask_i***. The output of the FDCT is

$$FDCT\left[\left\{\left\langle mask\_i\_b(u,v) - \frac{128}{s} \text{for } u = 0,1,...7 v = 0,1,...7 \right\rangle\right\}\right]$$
$$= \left\{\left\langle \left(\frac{\beta V(0,0)}{s} - \frac{128^* N}{s}\right); \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i = f_1 = 0,..7\, j = f_2 = 0,..7 \text{ except } (i,j) = (f_1,f_2) = (0,0)\right\rangle\right\} \tag{16}$$

Since VLC and Inverse VLC are lossless we have at the input of the inverse quantizer at the MPEG-2 decoder (figure 3) is same as that at the output of the FDCT in figure 2. The output of the inverse quantizer is

$$Q^{-1}\left[\left\{\left\langle \left(\frac{\beta V(0,0)}{s} - \frac{128^* N}{s}\right); \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i = f_1 = 0,..7\, j = f_2 = 0,..7 \text{ except } (i,j) = (f_1,f_2) = (0,0)\right\rangle\right\}\right]$$
$$= \{\langle (\beta V(0,0) - 128^* N); \beta V(f_1,f_2) \text{ for } f_1 = 0,..7\, f_2 = 0,..7 \text{ except } (f_1,f_2) = (0,0)\rangle\} \tag{17}$$

Output of the Inverse DCT

$$IDCT[\{\langle (\beta V(0,0) - 128^* N); \beta V(f_1,f_2) \text{ for } f_1 = 0,..7 f_2 = 0,..7 \text{ except } (f_1,f_2) = (0,0)\rangle\}]$$
$$= \beta v(k,l) - 128 \quad \forall\, k,l \tag{18}$$

Since intra coded blocks use a fixed prediction value of 128. The output of the MPEG-2 decoder, after adding the fixed prediction 128 we have,

$$\begin{aligned} \text{Output} &= (\beta v(k,l) - 128) + 128 \quad \forall\, k,l \\ &= \beta \boldsymbol{v} \end{aligned} \tag{19}$$

This $\beta \boldsymbol{v}$ is used as a prediction for the next P and B frames. The above described processing is applied to the intra coded blocks of P and B frames as well.

Let us consider the masking of P or B frame. The P or B frame consists of inter/intra coded blocks. The intra coded blocks are processed exactly the same way as described above. Inter coded block would mean the predictions to be forward, backward or interpolated motion compensated predictions. All these would refer to the picture stores containing previous and future stores for prediction. But both the previous and future picture stores contain the same $\beta \boldsymbol{v}$ frame. Therefore the analysis for forward, backward or interpolated prediction is the same. In case of interpolated motion compensated predictions, there are two motion vectors but it can be easily shown that the method for forward/

backward motion compensated prediction applies for this case also. For simplicity of description we assume the P or B frames consists only of inter coded blocks (with forward/backward motion compensated predictions).

For masking the inter coded blocks we use ***mask_n*** instead of ***mask_i***. Then the input of the FDCT in figure 2 is the prediction error frame, $\{\langle mask\_n\_b(u,v) - \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{ for } i = u = 0, ...7\ j = v = 0, ...7\rangle\}$ Where, ***mask_n_b*** refers to one block of ***mask_n***, β***v_b*** refers to one block of β***v*** and d_u, d_v are motion vectors. The output of the FDCT is

$$
\begin{aligned}
&FDCT\left[\left\{\left\langle mask\_n\_b(u,v) - \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{for } i = u = 0,1,...7\ j = v = 0,1,...7 \right\rangle\right\}\right] \\
&= FDCT[\{\langle mask\_n\_b(u,v) \text{ for } u = 0,...7 v = 0,...7\rangle\}] \\
&\quad - FDCT\left[\left\{\left\langle \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{ for } i = u = 0,...7\ j = v = 0,...7 \right\rangle\right\}\right] \\
&= \left\{\left\langle \frac{\beta V(0,0)}{Q_2(0,0)}; \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i = f_1 = 0,..7\, j = f_2 = 0,..7 \text{ except } (i,j) = (f_1,f_2) = (0,0) \right\rangle\right\} \\
&\quad - FDCT\left[\left\{\left\langle \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{for } i = u = 0,...7\ j = v = 0,...7 \right\rangle\right\}\right]
\end{aligned}
\tag{20}
$$

Since VLC and Inverse VLC are lossless we have at the input of the inverse quantizer at the MPEG-2 decoder (figure 3) is same as that at the output of the FDCT in figure 2. The output of the inverse quantizer is

$$
\begin{aligned}
&Q^{-1}\begin{bmatrix} \left\{\left\langle \frac{\beta V(0,0)}{Q_2(0,0)}; \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{ for } i = f_1 = 0,..7\, j = f_2 = 0,..7 \text{ except } (i,j) = (f_1,f_2) = (0,0) \right\rangle\right\} \\ -FDCT\left[\left\{\left\langle \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{ for } i = u = 0,...7\, j = v = 0,...7 \right\rangle\right\}\right] \end{bmatrix} \\
&= Q^{-1}\left[\left\{\left\langle \frac{\beta V(0,0)}{Q_2(0,0)}; \frac{\beta V(f_1,f_2)}{Q(i,j)} \text{for } i = f_1 = 0,..7\, j = f_2 = 0,..7 \text{ except } (i,j) = (f_1,f_2) = (0,0) \right\rangle\right\}\right] \\
&\quad - Q^{-1}\left[FDCT\left[\left\{\left\langle \frac{\beta v\_b(u-d_u, v-d_v)}{Q_2(i,j)} \text{ for } i = u = 0,...7\, j = v = 0,...7 \right\rangle\right\}\right]\right] \\
&= \{\langle \beta V(0,0); \beta V(f_1,f_2) \text{ for } f_1 = 0,..7 f_2 = 0,..7 \text{ except } (f_1,f_2) = (0,0)\rangle\} \\
&\quad - FDCT[\{\langle \beta v\_b(u-d_u, v-d_v) \text{ for } u = 0,...7\, v = 0,...7\rangle\}]
\end{aligned}
\tag{21}
$$

Output of the Inverse DCT,

$$
\begin{aligned}
&IDCT\begin{bmatrix} \{\langle \beta V(0,0); \beta V(f_1,f_2) \text{ for } f_1 = 0,..7 f_2 = 0,..7 \text{ except } (f_1,f_2) = (0,0)\rangle\} \\ -FDCT[\{\langle \beta v\_b(u-d_u, v-d_v) \text{ for } u = 0,...7\, v = 0,...7\rangle\}] \end{bmatrix} \\
&= \beta v(k,l) - IDCT[FDCT[\{\langle \beta v\_b(u-d_u, v-d_v) \text{ for } u = 0,...7\, v = 0,...7\rangle\}]] \quad \forall\ k,l \\
&= \beta v(k,l) - \beta v(k-d_u, l-d_v) \qquad \forall\ k,l
\end{aligned}
\tag{22}
$$

Output of the MPEG-2 decoder after adding the fixed prediction the motion compensated prediction

$$\text{Output} = (\beta v(k,l) - \beta v(k-d_u, l-d_v)) + \beta v(k-d_u, l-d_v) \quad \forall k,l = \beta \boldsymbol{v} \tag{23}$$

The presence of $\beta \boldsymbol{v}$ will cause the video to be obscured. Subscribers would be provided with an unmasking frame to view the video clearly. Unmasking is done in the spatial domain (i.e., after the decoding) as explained in Section 3.2.2. We will now explain the methods of obtaining copyright violator identification.

3.2 Copyright violator identification

The copyright violator identification property is obtained through the use of a robust invisible watermark created specifically for each subscriber by the broadcaster and is kept secret by the broadcaster. The unmasking frame carries this robust invisible watermark to the subscriber set top box and gets embedded in the video during unmasking process. The unmasking process is a single atomic process, which combines the watermarking for copyright violator identification and unmasking for confidentiality requirement. We now explain the unmasking frame construction and unmasking process.

3.2.1 Unmasking frame construction

Whenever a new subscriber wants to subscribe to the broadcast, the subscriber sends a join request containing verifiable subscriber's identity and makes arrangements to pay the necessary subscription fee. The broadcaster then verifies the subscriber's identity and then creates a robust invisible watermark $\boldsymbol{W}_{bi}$ specifically for the subscriber 'R_i'. The robust invisible watermark construction is explained in Section 3.3. The robust invisible watermark is kept secret by the broadcaster. The broadcaster then creates an unmasking frame $\boldsymbol{v}_i$ for subscriber 'R_i' (in figure 1 we use $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$ instead of $\boldsymbol{v}_i$. The subscripts 'a' and 'b' indicate that the unmasking frames $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$ are for subscriber A and subscriber B, respectively):

$$v_i(k,l) = \beta v(k,l) - \alpha W_{bi}(k,l) \quad \forall\, k,l \tag{24}$$

The broadcaster then transmits $\boldsymbol{v}_i$ to the subscriber through a secure (encrypted) channel. the broadcaster also stores the subscriber's identity and watermark $\boldsymbol{W}_{bi}$ In a table named '*SubscriberInfoTable*'.

3.2.2 Unmasking process

To view the unobscured channel broadcast, the subscriber R_i's set top box, performs this computation:

$$x_n^{w_i}(k,l) = \left(x_n^m(k,l) - v_i(k,l)\right)(1/\alpha) \quad \forall\, k,l \tag{25}$$

where $\boldsymbol{x}_n^{w_i}$ is the watermarked video frame for 'R_i'. Eq. 25, defines the unmasking process. in the case of compressed domain broadcasts, the output of the mpeg-2 decoder in the set top box is the masked video $\boldsymbol{x}_n^m$. The unmasking is applied after the mpeg-2 decoding.

Notice that $\boldsymbol{x}_n^{w_i}$ contains the robust invisible watermark $\boldsymbol{W}_{bi}$ left behind as a residue i.e.,

$$x_n^{w_i}(k,l) = x_n(k,l) + W_{bi}(k,l) \quad \forall\, k,l \tag{26}$$

Thus the unmasking process defined by Eq. 25 is a single atomic process, which combines the watermarking for copyright violator identification and unmasking for confidentiality requirement. in the case of compressed domain broadcasts the Eq. 26 will contain $\widehat{\boldsymbol{x}_n}$ instead of $\boldsymbol{x}_n$ to reflect the fact that the mpeg-2 is a lossy compression technique. the unmasking frame $\boldsymbol{v}_i$ is for the exclusive use of subscriber 'R_i'. If subscriber 'R_i' leaks/sells $\boldsymbol{v}_i$ or $\boldsymbol{x}_n^{w_i}$ to a non-subscriber, the illegal video would contain R_i's watermark. thus, any piracy done by 'R_i' is easily detected because of the invisible watermark present in the pirated video.

We use the spread spectrum watermark proposed by Hartung et. al. [18] in our implementation. However one could use any robust invisible watermark. We use the correlation receiver technique for the watermark detection [18]. We will now provide details about the watermark construction and detection procedures.

3.3 Watermark construction

We construct a watermark frame [18] of dimension K pixels by L pixels, same as that of the video frames. we consider the watermark frame as a 1-dimensional signal acquired by raster-scanning (scanning left to right and then top to bottom). Assume that the information to be embedded consists of bits having values $\{-1,1\}$. let us create a sequence a_j out of it (the watermark information to be embedded).

Let

$$a_j, \quad a_j \in \{-1,1\}, \quad j = 0,1,....N \tag{27}$$

be a sequence of bits, which is then spread using the chip rate C_r to obtain the spread sequence b_i. The C_r and N are selected in such a way that $C_r \times N = K \times L$, the frame dimension.

$$b_i = a_j \quad jC_r \le i < (j+1)C_r, \quad \forall j \tag{28}$$

The spreading provides redundancy and improves the robustness to geometrical attacks such as cropping. the spread sequence is then multiplied with a pseudo random noise sequence p_i where $p_i \in \{-1,1\}$. It is then amplified by a scaling factor 'κ' (a positive number, selected in such a way that watermark still remains invisible in the watermarked frames, and is also detectable) to get the watermark.

$$w_i = \kappa b_i p_i \quad \forall i \tag{29}$$

The watermark w_i could be arranged as a frame (dimension $K \times L$), which is the watermark frame.

3.4 Watermark detection

The detection of the hidden information a_j is done by employing the correlation receiver [18]. The correlation receiver does not require the original unwatermarked video signal for the detection. to detect a_j we multiply the watermarked video x_i^w by the same pseudo random noise sequence p_i that was used for watermark construction followed by a

summation over the window for each embedded information, yielding the correlation s_j. The sign (s_j) is the a_j.

$$s_j = \sum_{i=jc_r}^{(j+1)c_r-1} p_i x_i^w = \sum_{i=jc_r}^{(j+1)c_r-1} p_i x_i + \sum_{i=jc_r}^{(j+1)c_r-1} p_i w_i = \sum_{i=jc_r}^{(j+1)c_r-1} p_i x_i + \sum_{i=jc_r}^{(j+1)c_r-1} p_i^2 \kappa b_i \quad (30)$$

where x_i is the original unwatermarked video. The first term in Eq. 30 is zero if p_i and x_i is uncorrelated. However this is not always the case in real. So to obtain a better result we first prefilter the watermarked video x_i^w and remove most of the unwatermarked video content. But if we have the original unwatermarked video we just need to subtract the original unwatermarked video from the watermarked video x_i^w. Assuming that the first term in Eq. 30 is almost zero,

$$s_j = \sum_{i=jc_r}^{(j+1)c_r-1} p_i^2 \kappa b_i = \sum_{i=jc_r}^{(j+1)c_r-1} p_i^2 \kappa a_j = \kappa a_j \sum_{i=jc_r}^{(j+1)c_r-1} p_i^2 = a_j \kappa \sigma_p^2 \quad (31)$$

Since κ and σ_p^2 are positive, we have

$$sign(s_j) = sign\left(a_j \kappa \sigma_p^2\right) = a_j \quad (32)$$

Next we explain the protocol used to identify the copyright violator from the unauthorized copy found with the non-subscriber.

3.5 Copyright violator identification protocol

Suppose a legal recipient makes multiple copies of the unmasked watermarked video $\boldsymbol{x}_n^{w_i}$ or the unmasking frame $\boldsymbol{v}_i$ and redistributes to non-subscribers. The broadcaster can identify the subscriber who has redistributed the video by detecting the watermark $\boldsymbol{W}_{bi}$ present in the unauthorized copy found with the non-subscriber. For this purpose the broadcaster picks up one by one the watermarks created by him, the $\boldsymbol{W}_{bi}$ s from the *SubscriberInfoTable* and then correlates it with the copy found with the non-subscriber. The highest correlation value with certain minimum threshold value is used to identify the watermark $\boldsymbol{W}_{bi}$ present in the copy of the video. If the correlation value is smaller than the minimum threshold we declare that the watermark is not found. Once the watermark $\boldsymbol{W}_{bi}$ is identified it could be obtained from the *SubscriberInfoTable* the identity of the subscriber R_i who is the legal recipient. The broadcaster can then initiate necessary legal measures and prove to the judge the existence of $\boldsymbol{W}_{bi}$ in the unauthorized copy.

4 Implementation and results

We have implemented our technique for spatial and MPEG-2 compressed domain, and tested it on several video clips. However we show here only the results of compressed domain as the results of spatial domain is similar. We apply the proposed scheme only on the luminance channel of the video frames. However it is possible to implement it on the

chrominance channels as well. We have worked with various sets of scaling factors α, β and also various mask images. The higher the β value is set, the more is the obscurity and mask frames with high saturation values will also have more obscurity. The unmasked watermarked video frames for 'R_i' contains the invisible watermark for 'R_i'. The watermarks in these unmasked watermarked video frames can be detected using the correlation receiver. Figure 4 depicts the full masking and unmasking results for one frame of one of the test videos with frame dimension 720×576.

4.1 Quantitative measure of degradation

To evaluate the degradation caused and to evaluate the performance of the proposed scheme a quantitative measurement is required. For this purpose the signal to noise ratio in dB (SNR_{indB}) is used and is defined by

$$SNR_{indB} = 10\log_{10}\frac{E\left\{\left(x_n'(k,l)\right)^2\right\}}{E\left\{\left(x_n'(k,l) - x_n^{w_i}(k,l)\right)^2\right\}} \quad \forall\, k,l \tag{33}$$

Where $E\{\cdot\}$ is the expectation operator. The $\hat{\boldsymbol{x}}_n$ is used instead of $\boldsymbol{x}_n$ to reflect the loss caused due to MPEG-2 compression. This quantitative value however does not truly reflect the perceptual quality of the unmasked watermarked video. The numbers give us a quantitative measure of the degradation. The signal to noise ratio has been calculated for several video clips of MPEG-7 video categories and is plotted in figure 5. The MPEG-7 video set consists of ten categories with 30 items. But our test covers only eight categories, which have 27 items, 12:50:47 h duration and 1,210,642 number of frames. The video clips used are with frame dimension 352×288 and the transmission rate used is 5 Mbits/s.

The signal to noise ratio calculation is performed frame wise and the SNRmax refers to the highest signal to noise ratio, SNRavg refers to the average of the signal to noise ratios and SNRmin is the minimum signal to noise ratio. The signal to noise ratio in dB with watermark is plotted in figure 5c and that without watermark in figure 5d. It can be observed that the degradation is very small. The degradation has two components one due to the rounding operation and the other due to the watermark. The watermark amplitudes

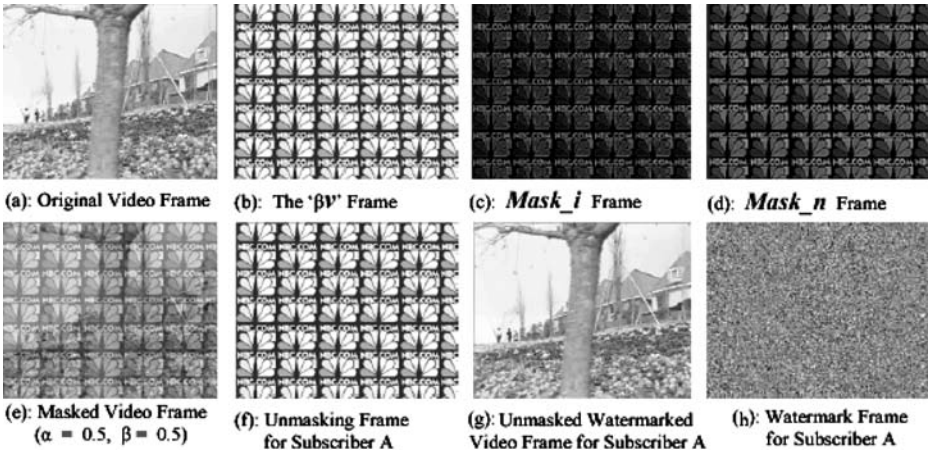

Fig. 4 Masking and unmasking results

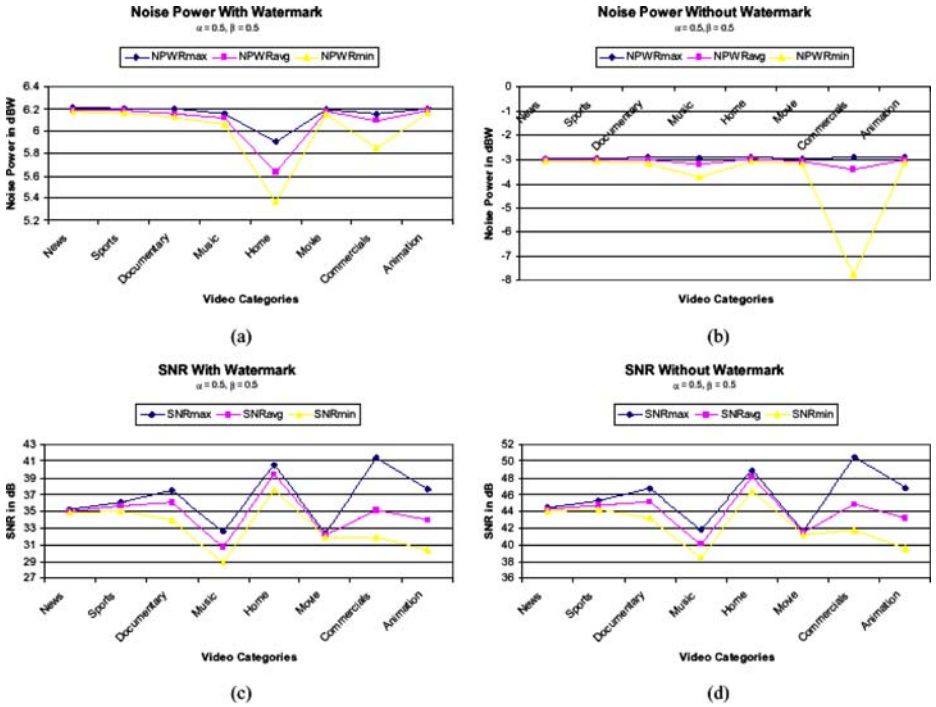

Fig. 5 Plots of noise power and signal to noise ratio

used are +2, −2 (i.e., κ used is 2). This degradation is not visible in the unmasked watermarked video in any of the clips and is perceptually similar to the original video. The amplitude of the watermark should be selected in such a way that the degradation should not be visible. The degradation due to other processing errors with zero watermark strength shows that the processing degradation is negligible. The figure 5a & b are the plots of noise power with and without watermark respectively. The NPWRmax, NPWRavg and NPWRmin refer to maximum, average and minimum noise powers respectively.

4.2 Computation overhead

The computation overhead of the mask blending process in comparison to the MPEG-2 compression was investigated. A *macroblock* size of 16 pixels row by 16 pixels column (consisting of four *blocks*) was taken as a unit of our investigation. Except the motion estimation all other processing are done *block* (eight pixels raw by eight pixels column) wise.

The following assumptions were made while finding the computational costs. We assume that the two dimensional forward discrete cosine transform (FDCT) and the inverse discrete cosine transform (IDCT) are implemented using the Haque's fast block matrix decomposed algorithm [17, 37]. This algorithm requires for a *block* size of N pixels raw by N pixels column, $(3/4)\,N^2 \log_2 N$ real multiplications and $3N^2 \log_2 N - 2N^2 + 2N$ real additions. The inter coded *macroblocks* can be forward, backward or interpolated coded. We assume the interpolation function used is pixel averaging function. For motion vector estimation, we assume that a fast six step algorithm is used and further we assume the use

of sum of absolute difference (SAD) as a measure of best match. The SAD is defined for a 16 pixel row by 16 pixel column *macroblock* as follows:

$$SAD(k,l) = \sum_{i=0}^{15}\sum_{j=0}^{15} |x_n(k+i, l+j) - x_m(k+dk+i, l+dl+j)| \tag{34}$$

where $x_n(k+i, l+j)$ is pixel intensity value at *macroblock* position (k, l) in source picture n and $x_m(k+dk+i, l+dl+j)$ is the pixel intensity at *macroblock* position $(k+dk, l+dl)$ in reference picture m. The 16×16 array in picture m is displaced horizontally by dk and vertically by dl. By convention (k, l) refers to the upper left corner of the *macroblock*, indices (i, j) refer to values to the right and down, and displacements (dk, dl) are positive when to the right and down. For a six step algorithm, in the first step we evaluate SAD at nine displacements, and subsequent four steps we calculate SAD at eight displacements followed by the last step for 0.5 pixel precision at four displacements. The last step SAD is computed between interpolated values of the *macroblock*. We ignore the computation cost of this interpolation as is not substantial. We also do not consider the computation cost involved in the variable length coding which is a part of MPEG-2 compression. With these assumptions the computational costs of MPEG-2 compression and mask blending in MPEG-2 compressed domain are found and are depicted in Table 2.

We see that when compared to the computation requirement of MPEG-2 compression, the mask blending process require less computation. The main contributor of computation to the MPEG-2 compression is the motion estimation, which is not present in the mask blending process. The actual computation cost of MPEG-2 compression would have been more had we considered the computation cost due to variable length coding and also the motion estimation for 0.5 pixel accuracy. The mask blending is done only once, irrespective of the number of subscribers.

4.3 Compression overhead

In case of raw video broadcasting, the mask blending does not increase the message size i.e., the original video and the masked video are of the same size. But in case of compressed domain processing, the compression ratio would be affected as seen from figure 6. The compression ratio is defined as follows:

$$Compression\ Ratio = \frac{Size\ Of\ Compressed\ Masked\ Video}{Size\ Of\ Compressed\ Original\ Video} \tag{35}$$

Table 2 The computation overhead

	MPEG-2 compression		Mask blending process	
Macroblock	Multiplications/ divisions	Additions/ subtractions	Multiplications/ divisions	Additions/ subtractions
Intra coded I & P	1,664	3,968	832	2,368
Intra coded B	832	2,112	832	2,368
Forward coded P	1,664	26,963	1,088	2,368
Forward or backward coded B	832	25,107	1,088	2,368
Interpolated coded B	1,920	50,470	1,344	2,624

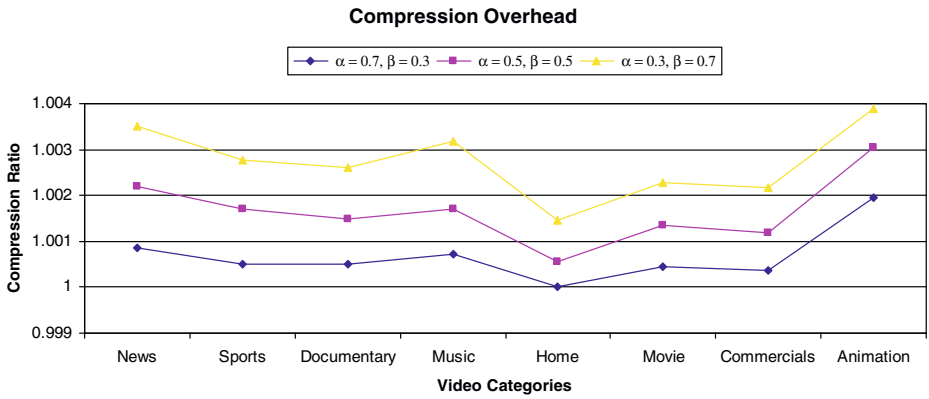

Fig. 6 Plot of compression overhead with ($\alpha = 0.7$, $\beta = 0.3$), ($\alpha = 0.5$, $\beta = 0.5$) and ($\alpha = 0.3$, $\beta = 0.7$)

For various video categories we find out the compression ratio defined by the Eq. 35 and is plotted in figure 6. We see that there is a small compression overhead ranging from 0.001 to 0.4%. The compression overhead is found to be increased when the mask strength is increased from $\beta = 0.3$ to $\beta = 0.7$.

5 Discussion

We now discuss in detail some of the salient aspects of our scheme.

Key Management The unmasking frame acts as access control key in our proposed scheme. For a frame dimension of 352×288, the size of the unmasking frame in bytes is 101,376 bytes (99 kB). In order to have better security and also to support dynamic leave feature, the mask could be changed frequently. A receiver can join the broadcast anytime he wishes by sending the join request to the broadcaster. Therefore the proposed scheme supports dynamic join and leave feature. The key revocation to close off a subscriber who is no longer paying can be done at the time of mask change. Whenever the mask is changed the corresponding unmasking frame for each subscriber is to be generated and transmitted. The unmasking frame is handed over to the subscriber through a secure channel. Assuming the mask frame validity period 30 min, video frame dimension 352×288, video frame rate 25 frames/s and the typical MPEG-2 compression ratio 1:10, the size of the MPEG-2 video in bytes is $(352 \times 288) \times 25 \times 60 \times 30/10 = 456{,}192{,}000$ bytes. Therefore, the key message size is just 0.022% of the 30 min MPEG-2 video size, which is small. The control word size of the current CAS system is in bits (168 for ATSC), which is very small in comparison to the unmasking frame size. But the current CAS systems do not support fingerprinting. It is hard to integrate the fingerprinting and the current descrambling process of CAS into a single atomic process as discussed in Section 2.1. In case of digital cinema broadcast to the cinema halls, the fingerprinting requirement is as important as that of CAS due to the high value of new digital movie releases.

Security Concerns The security of the watermarks has been well studied [19, 21]. We, in our scheme, make use of robust invisible watermark by Hartung et al. [18], which is a spread spectrum based watermark. Therefore the security of the watermark in our scheme is

similar to that of [18]. There are many remedies and counter attacks presented in [21] to make the spread spectrum based watermarks more resistant against attacks.

Suppose 'n' subscribers collude to make an unwatermarked video by averaging the corresponding frames of each subscriber's video, or they collude by averaging the unmasking frames to make an unmasking frame which does not carry watermark (more precisely inverse watermark) information. Boneh & Shaw [5, 21] have shown, how to construct watermark signals to defeat this kind of averaging collusion attacks. In the event of collusion, Boneh & Shaw's schemes would point out the colluding parties. There is another kind of collusion where the colluding parties can assemble video by randomly selecting frames from each of their watermarked videos. But since watermarking is frame wise, this would reveal the colluding parties. The subscribers can assemble an unmasking frame strip by strip, by switching between different unmasking frames or can create an unmasked video, strip by strip from each corresponding frame, by switching between different unmasked watermarked video. In either case to defeat this kind of collusion the watermark signal/information bits have to be pseudo-randomly distributed to pixels using pseudo random noise sequence p_i [21].

The inversion/ambiguity attack can be carried out for false ownership claim of watermarked data. The attacker first guesses a false watermark and then derives a false original data from the watermarked data using the guessed watermark. By producing the guessed watermark and the derived false original the attacker then claims ownership of the watermarked data. This attack can be defeated by designing a non-invertible watermark [21]. One of the ways of designing non-invertible watermark is by using the cryptographically secure time stamps provided by trusted third parties and encoded in the watermark [41]. Another way is by making the watermark to be dependent on the original data in a one way fashion (for example using hash function) [11, 41].

Another attack is to estimate the mask frame from the masked video frames or watermark frame from the unmasked watermarked video frames. In order to make the estimation of mask frame or watermark frame using Wiener estimator more difficult, the power spectrum of mask frame and watermark frame should be a scaled version of the video signal power spectrum [21]. It can be shown that it is hard to separate mask from masked video through brute force technique as it requires enormous amount of computing power coupled with human interaction to identify the correct/acceptable original video frame and mask frame. The broadcaster must make sure that the mask frame should not be made known (either through applying mask on a blank frame or through some other means) or should not be easily guessed by the receivers (subscribers & non-subscribers). To make the proposed method more robust against the attacks, one needs to design a porous mask and robust invisible watermark, which means we mask and watermark only some random pixel locations (substantially large in number in order to have the opacity effect) or change the mask often or use multiple masks and use them interchangeably.

Other Advantages In the proposed scheme the operations performed for masking and unmasking are simple additions. Therefore the scheme is not compute-intensive. It can be easily seen that all operations are O(n) where 'n' is the size of the frame. The masked video frames are created only once for a video but the unmasking frames and the robust invisible watermark are computed whenever a new subscriber subscribes to the service. The unmasking frames are also computed whenever the mask frames are changed (for better security) but the robust invisible watermark need not be created then. The unmasking frame is handed over to the subscriber through a secure channel and becomes the access control mechanism. The proposed scheme combines the unmasking and watermarking process into

one single atomic process, making the attacker to put more effort to get away with an unwatermarked video.

The proposed scheme supports MPEG-2 compressed domain processing. The masking for confidentiality requirement is carried out on quantized error DCT coefficients of MPEG-2 stream. Therefore, in our proposed scheme, in order to mask an already MPEG-2 compressed video, the broadcaster needs only partial decoding (run length decoding) to be done. After masking, the masked error DCT coefficients are VLC coded transmitted. Though the masking is carried out on quantized error DCT coefficients at the broadcaster site, it is done in such a way that unmasking can be done after MPEG-2 decoding (uncompressed domain) at the subscriber site. The unmasking of masked video using the unmasking frame (which carries watermark information) results in watermarking of video. Since, the watermarking is carried out after MPEG-2 decoding there is no drifting problem during decoding.

6 Conclusion

We have developed a scheme to simultaneously obtain confidentiality and fingerprinting for spatial and MPEG-2 compressed broadcast video. Broadcasting demands a single copy for transmission where as fingerprinting demands several individually watermarked copies. The proposed scheme satisfies both the demands. Confidentiality is achieved by blending an additive mask frame. By selecting a proper mask and by controlling the masking operation one could achieve a transparency continuously ranging from fully transparent to absolutely opaque. Fingerprinting is obtained through additive robust invisible watermarking. The proposed scheme combines the unmasking and watermarking process into one single process, which makes the attacker to put more effort to get away with an unwatermarked copy. The proposed scheme supports dynamic join and leave, requires low resources in terms of computing power and bandwidth and not complex in terms of implementation. Our future directions are to extend the scheme to audio stream of the broadcasts.

References

1. Anderson R, Manifavas C (1997, January) Chameleon—a new kind of stream cipher. Encryption in Haifa
2. Braudaway GW, Magerlein KA, Mintzer F (1997) Protecting publicly available images with a visible image watermark. Int Conf Image Proc 1:524–527
3. Briscoe B, Fairman I (1999, June) Nark: receiver based multicast key management and non-repudiation. BT Technical Report
4. Brown I, Crowcroft J, Perkins C (1999, November) Watercasting: distributed watermarking of multicast media. Networked group communications. Italy, pp 286–300
5. Boneh D, Shaw J (1998, September) Collusion-secure fingerprinting for digital data. IEEE Trans Inf Theory 44:1897–1905
6. Buer M, Wallace J (1996, August) Integrated security for digital video broadcast. IEEE Trans Consum Electron 42(3):500–503
7. Chu HH, Qiao L, Nahrstedt K (1999, January) A secure multicast protocol with copyright protection. Proceedings of SPIE symposium on electronic imaging: science and technology
8. Clayson PL, Dallard NS (1997, September) Systems issues in the implementation of DVB simulcrypt conditional access. Int Broadcast Conv 470–475
9. Cox IJ, Killian J, Leighton T, Shamoon T (1997, December) Secured spread spectrum watermarking for multimedia. IEEE Trans Image Process 6(12):1673–1687
10. Cox IJ, Miller ML, Bloom JA Digital watermarking. Morgan Kaufmann Publishers, Inc., San Francisco, 2001

11. Craver S, Memon N, Yeo B, Yeung MM (1998, May) Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks and implications. IEEE J Sel Areas Commun 16(4):573–586
12. Cutts DJ (1997, February) DVB conditional access. Electron Commun Eng J 9(1):21–27
13. Dittman J, Stabenau M, Steinmetz R (1998) Robust MPEG video watermarking technologies. ACM International Multimedia Conference, pp 71 – 80
14. Doërr G, Dugelay J-L (2003) A guide tour of video watermarking. Signal Process, Image Commun 18 (4):263–282
15. Emmanuel S, Kankanhalli MS (2001, August) Copyright protection for MPEG-2 compressed broadcast video. Proceedings of the IEEE international conference on multimedia and expo (ICME 2001), Tokyo
16. Emmanuel S, Kankanhalli MS (2003) A digital rights management scheme for broadcast video. Multimedia Systems Journal 8(6):444 – 458
17. Haque MA (1985, December) A two-dimensional fast cosine transform. IEEE Trans Acoust Speech Signal Process 33(6):1532–1539
18. Hartung F, Girod B (1998, May) Watermarking of uncompressed and compressed video. Signal Process 66(3):283 – 301
19. Hartung F, Kutter M (1999, July) Multimedia watermarking techniques. Proc IEEE 87(7):1079 – 1107
20. Hartung F, Ramme F (2000, November) Digital rights management and watermarking of multimedia content for M-commerce applications. IEEE Commun Mag 38(11):78 – 84
21. Girod B, Hartung F, Su JK (1999, January) Spread spectrum watermarking: malicious attacks and counterattacks. Proceedings of the SPIE, Electronic Imaging 99, San Jose, USA, vol 3657, pp 147–158
22. Haskell BG, Netravali AN, Puri A (1997) Digital video: an introduction to MPEG-2. International Thomson, Chapman & Hall
23. http://www.atsc.org/standards.html. ATSC Standard A/70: Conditional Access System for Terrestrial Broadcast with Amendment
24. http://www.chiariglione.org/mpeg/index.htm. The MPEG Home Page. Text of ISO/IEC 13818-1
25. ISO/IEC 13818-1: Generic coding of moving pictures and associated audio: systems. (MPEG-2 Systems)
26. ISO/IEC 14496-1: Coding of audiovisual objects: systems. (MPEG-4 Systems)
27. Ammar M, Judge P (2000, June) WHIM: watermarking multicast video with a hierarchy of intermediaries. NOSSDAV, North Carolina
28. Kalker T (1999) System issues in digital image and video watermarking for copyright protection. IEEE Int Conf Multimedia Comput Syst 1:562–567
29. Kankanhalli MS, Rajmohan, Ramakrishnan KR (1999) Adaptive visible watermarking of images. IEEE Int Conf Multimedia Comput Syst 1:568–573
30. Kundur D, Hatzinakos D (1999, July) Digital watermarking for telltale tamper proofing and authentication. Proc IEEE 87(7):1167–1180
31. Linnartz JP, Depovere G, Kalker T Philips electronics response to call for proposals issued by the data hiding subGroup copy protection technical working group
32. Macq BM, Quisquater JJ (1995, June) Cryptology for digital TV broadcasting. Proc IEEE 83 (6):944 – 957
33. Meng J, Chang SF (1998) Embedding visible video watermarks in the compressed domain. Int Conf Image Proc 1:474 – 477
34. Mooij W (1997, September) Advances in conditional access technology. Int Broadcast Conv, pp 461–464
35. Parviainen R, Parnes P (2001, May) Large scale distributed watermarking of multicast media through encryption. Proceedings of the CMS 2001, Germany
36. Piva A, Barni M, Bartolini F, Cappellini V (1997) DCT-based watermark recovering without resorting to the uncorrupted original image. Int Conf Image Proc 1:520 – 523
37. Rao KR, Yip P (1990) Discrete cosine transform algorithms advantages applications. Academic
38. Strycker LD, Termont P, Vandewege J, Haitsma J, Kalker A, Maes M, Depovere G (2000, August) Implementation of a real-time digital watermarking process for broadcast monitoring on a TriMedia VLIW processor. IEE Proceedings, Visual Image Signal Processing 147(4)
39. Su K, Kundur D, Hatzinakos D (2004) Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance. IEEE Trans Multimedia
40. Voyatzis G, Pitas I (1999, July) The use of watermarking in the protection of digital multimedia products. Proc IEEE 87(7)
41. Wolfgang RB, Delp EJ (1997, June) A watermarking technique for digital imagery: further studies. Proceedings of the international conference on imaging science, systems and applications (CISST 97), Las Vegas, NV, USA, pp 279 – 287
42. Zeng W, Lei S (1999, November) Efficient frequency domain digital video scrambling for content access control. ACM Multimedia '99 Proceedings, Orlando, Florida

Sabu Emmanuel received his B.E. (Electronics & Communication Engineering) from Regional Engineering College, Durgapur (1988), M.E. (Electrical Communication Engineering) from Indian Institute of Science (IISc.), Bangalore (1998), and Ph.D. (Computer Science) from National University of Singapore (NUS) (2002). He is an Assistant Professor at the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests are in media forensics, digital rights management (DRM), and wireless communication security. He has been a member of the technical program committee of several international conferences.

Mohan Kankanhalli obtained his BTech (Electrical Engineering) from the Indian Institute of Technology, Kharagpur and his MS/PhD (Computer and Systems Engineering) from the Rensselaer Polytechnic Institute. He is a Professor at the School of Computing at the National University of Singapore. He is on the editorial boards of several journals including the ACM Transactions on Multimedia Computing, Communications, and Applications, IEEE Transactions on Multimedia, ACM/Springer Multimedia Systems Journal, Pattern Recognition Journal and the IEEE Transactions on Information Forensics and Security. His current research interests are in Multimedia Systems (content processing, retrieval) and Multimedia Security (surveillance, authentication and digital rights management).