Multimedia Tools and Applications (2019) 78:21245–21271
https://doi.org/10.1007/s11042-019-7456-0

An efficient authentication scheme for high efficiency video coding/H.265

Gagandeep Kaur[1] · Singara Singh Kasana[1] · M. K. Sharma[2]

Received: 12 July 2018 / Revised: 1 January 2019 / Accepted: 6 March 2019 /
Published online: 15 March 2019

Abstract

In this paper, a semi-fragile authentication scheme for high efficiency video coding is proposed. In this scheme, 4×4 intra luma transform blocks of I-frames are divided into two disjoint subsets. One subset is used for authentication code generation and another subset is used for embedding the generated code. From these subsets, blocks are selected on the basis of the quality and robustness thresholds. These thresholds are calculated at runtime by using a low-complexity spatial analysis. The authentication code is generated based on the relationship between number of positive and negative quantized discrete sine transformed coefficients in a block. The generated authentication code is embedded by altering the magnitudes of quantized discrete sine transformed coefficients during encoding of the video sequence. The process of authentication has low complexity as embedded authentication code can be extracted and verified without full decoding of the encoded bit-stream. Experimental results show that the proposed scheme is efficient in terms of imperceptibility, increase in bit-rate, computational complexity, robustness to re-compression, frame dropping, noise attacks and fragile to malicious attacks.

Keywords High efficiency video coding · DST · Video authentication · Video watermarking

1 Introduction

In today's digital world, multimedia contents like videos are widely used in many domains like entertainment, advertisements, social networking sites, etc. These digital videos can be

✉ Gagandeep Kaur
gagandeep.kaur@thapar.edu

Singara Singh Kasana
singara@thapar.edu

M. K. Sharma
mksharma@thapar.edu

[1] Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

[2] School of Mathematics, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

manipulated and redistributed very easily with imperceptible changes in visual quality. So, there is need of protection of the integrity of data to ensure the data is not tampered during the transmission process. Since 1954, digital watermarking has been used for authentication of digital multimedia data [2]. In digital watermarking secret information, i.e. watermark is embedded in the multimedia data to protect integrity of the content. The watermark is embedded using certain features of video coding standards like transform coefficients, intra prediction modes, motion vector information, block partitioning data, etc.

With the advancement in electronics and electrical engineering, more efficient portable multimedia devices with multi-core processors and high resolution displays are manufactured. These devices support digital videos with high definition and ultra-high definition resolutions, but existing transmission channels do not have sufficient bandwidth to transfer huge volumes of high definition multimedia data. This shortcoming of transmission channels demanded more efficient video codec with high compression and parallel processing capabilities. High Efficiency Video Coding (HEVC) is the latest video coding standard conceived in 2013 [16, 24] which supports parallel processing capability and have 50% more compression efficiency than H.264/AVC. The major enhancements made in HEVC standard with respect to H.264/AVC [23] standard are (1) Each Frame is divided into 64×64 Coding Tree Units (CTU) i.e. one luma Coding Tree Block (CTB) and corresponding two chroma CTB's instead of 16 × 16 macro blocks, (2) Each CTU can be further partitioned into quadtree structure of size 64 × 64, 32 × 32, 16 × 16 and 8 × 8 Coding Units (CU) i.e. one luma Coding Block (CB)and corresponding two chroma CB's, (3) Each CU can also be partitioned in quadtree of size 32 × 32, 16 × 16, 8 × 8 and 4 × 4 Transform Units (TU) i.e. one luma Tranform Block (TB) and corresponding two chroma TB's, (4) 4 × 4 intra luma TB use Discrete Sine Transform (DST) instead of Discrete Cosine Transform (DCT), (5) There are 35 intra prediction mode in HEVC as compared to only 9 modes in H.264/AVC.

The main features of an efficient authentication scheme are invisible degradation in visual quality, low computational complexity, minimal increase in bit-rate and should be robust against transcoding processes like re-compression with a different quantization parameter, frame dropping, noise attacks during transmission of video and on the other hand it should be fragile against malicious attacks, i.e. it can detect removal and insertion of objects. So, an efficient authentication scheme should be semi-fragile, i.e. robust against content preserving manipulations but fragile to content changing manipulations. There are plenteous techniques in literature for protecting the integrity, i.e. authentications of H.264/AVC standard. Unfortunately, all existing authentication techniques of H.264/AVC cannot be applied to HEVC encoded videos because of the encoding differences between the standards. There are a few authentication schemes in existing literature with respect to HEVC video coding standard. The research problem in this work is based on challenges faced with employing existing schemes for authentication of the HEVC encoded video sequences in terms of robustness against content preserving attacks, i.e. re-compression, frame dropping and noise attacks.

In the following text, the efficiency of existing schemes is analyzed. In [1] watermark bits are embedded into DST transformed coefficients of 4 × 4 intra luma blocks with drift compensation to avoid error propagation to neighbouring blocks. Although, this algorithm has a large embedding capacity with good visual quality, but it is not efficient in terms of non-malicious video processing manipulations because selection of 4 × 4 blocks for watermark embedding is done without any analysis for robustness. Dutta et al. [3] proposed an effective algorithm based on DCT transformed coefficients of 4 × 4 intra luma blocks. This algorithm is robust against re-compression and signal processing attacks like filtering, noise additions etc. Elrowayati et al. [4] proposed robust algorithm, where the watermark is embedded based on the parity of nonzero quantized transformed coefficients. This algorithm is robust against noise attacks and helps in improving performance of the HEVC encoder to detect and correct errors caused due to transmission over noisy channels. Though, it is robust against noise attacks, but it is fragile to re-compression as parities of coefficients are changed during re-encoding of video sequences. In [10] watermark is embedded by altering the selected set of DST coefficients based on the intra prediction mode used in 4 × 4 intra luma block. This scheme has high embedding capacity with acceptable degradation in visual quality. However, this scheme is not efficient in terms of re-compression and video processing attacks because selection of 4 × 4 blocks is done without any spatial analysis for robustness. In [14] watermark is embedded by altering CU split decision. In this technique split decision of only 32 × 32 and 16 × 16 CU's are used, which can easily be changed by the re-compression process.

Swati et al. [17] embedded watermark imperceptibly by changing the parity of Least Significant Bits (LSB) of quantized DCT transformed coefficients within the encoding loop. This algorithm is not robust against re-compression because LSB's are altered during the re-encoding of video sequeunces. In [19] authentication code is generated using syntax elements like CU type and prediction mode of CU. The generated code is repeatedly embedded into the LSB's of quantized DCT transformed coefficients, the quantization parameter of each CTU and motion vectors of each slice, which are used to detect the tampered regions at decoder end. In [20] joint encryption and authentication scheme for HEVC encoded video sequences is proposed. In this scheme syntax elements are divided into two separate sets, one set is used for authentication process while the other is utilized for encryption. The coding unit size in each slice is utilized for generating authentication code, which is then repeatedly embedded into LSB's of the quantized DCT transformed coefficients, quantization parameter of each CTU and motion vectors of each slice to detect the tampered regions. The schemes in [19, 20] utilizes the LSB's of the quantized DCT transformed coefficients, quantization parameter and motion vectors. The values of these syntax elements are modified by the encoder when video sequence is re-compressed with different quantization parameter. So, these schemes are fragile to the re-compression process of HEVC encoder. These schemes can detect malicious attacks, but these are fragile to content preserving manipulations like re-compression. In [22] intra prediction mode of 4 × 4 intra luma blocks are used for watermark embedding during encoding of video sequences. The 33 angular intra prediction modes are assigned angle values. The angle difference between intra prediction modes of two consecutive 4 × 4 intra luma blocks are used to embed watermark bit by replacing the best optimal mode with the sub optimal mode, based on mapping relationships of angle values. This scheme is not robust to re-compression because during re-encoding of video the sub optimal mode is replaced back to best optimal mode by HEVC encoder [8].

From the above analysis, it can be observed that the syntax elements used by Tew et al. schemes [19, 20] to embed authentication code are fragile to the re-compression process of HEVC encoder. So, there is no semi-fragile technique in the existing literature that is robust against content preserving manipulations like re-compression but fragile to content changing manipulations. Secondly, in [3] the watermarking algorithm is based on DCT transformed coefficients of 4 × 4 intra luma blocks, which is robust against re-compression and video processing attacks, but to reduce implementation complexity of HEVC encoder, mode dependent selection between DCT and DST is removed and only DST is used for 4 × 4 intra luma blocks [13]. There are differences in the properties of DST and DCT transforms, so DCT transformed coefficients based method does not give optimal results for DST transformed coefficients. The schemes of Chang et al. [1] and Liu et al. [10] are based on DST transformed coefficients of 4 × 4 intra luma blocks are not efficient in terms of re-compression, frame dropping and noise attacks because spatial analysis of 4 × 4 blocks is not done before embedding watermark bits. This motivated us to design DST based semi-fragile scheme that provides a practical solution for authentication of the HEVC encoded video sequences, which is robust against re-compression, frame dropping, addition of noise during transmission and simultaneously it can detect object insertion and deletion.

To overcome the limitations of existing literature, an efficient authentication scheme is proposed for HEVC encoded videos based on transform coefficients of 4 × 4 intra luma blocks, which is robust against content preserving attacks and simultaneously it can detect object insertion and deletion. In this scheme, 4 × 4 intra luma transforms blocks of I-frames are used for authentication of the video because I-frames contain vital information as compared to inter frames. The 4 × 4 blocks are selected on the basis of spatial analysis. The spatial analysis is performed on the basis of Number of Non Zero Quantized Discrete Sine Transform (NNZQDST) coefficients in a 4 × 4 block to enhance the imperceptibility of the proposed scheme. Further, to increase the robustness of the proposed scheme, spatial analysis based on NNZQDST coefficients with absolute value greater than 1 (ABGR1) is performed. The blocks which are selected by spatial analysis are divided into two subsets, i.e. one for generating authentication code and other for embedding the generated authentication code. An authentication code is generated based on the signs of transformed coefficients present in the block. To enhance the security of the proposed scheme, the generated authentication code is encrypted using content based public key and pseudo randomly generated private key. Further, the scheme induced minimal increase in bit-rate as the encrypted code is embedded into the magnitudes of non-zero Quantized Discrete Sine Transform (QDST) coefficients of 4 × 4 intra luma blocks while preserving the signs of the coefficients during encoding of video sequence into bit-stream. The scheme has low computational complexity as extraction and verification process is done at the decoder end without full decoding of the encoded bit-stream.

The rest of the paper is organized as follows. The proposed scheme is explained in Section 2 and the experimental results are discussed in Section 3. Finally, the conclusion of paper is drawn in Section 4.

2 Proposed authentication scheme

In this section, the proposed scheme is explained in detail. First, spatial analysis is explained which is used for selecting blocks, subsequently the algorithms for authentication code generation and embedding are discussed. Finally, the extraction and verification algorithms are discussed.

2.1 Spatial analysis for imperceptibility and robustness

An I-Frame can be partitioned into TB of size 32 × 32, 16 × 16, 8 × 8 or 4 × 4 on the basis of texture of I-frame. Smooth areas are partitioned into either 32 × 32 or 16 × 16 i.e. areas with less texture are divided into large size and dense textured areas are partitioned into 8 × 8 or 4 × 4 i.e. areas with more texture are divided into small size. TB with size 4 × 4 are chosen for watermark embedding because Human Visual System (HVS) is less sensitive to more textured areas as explained in [11].

Spatial analysis for imperceptibility and robustness are done on the basis of NNZQDST coefficients present in a 4 × 4 intra luma block. Spatial analysis done by Mansouri et al. in [11] is based on number of Non-Zero Quantized Discrete Cosine Transform (NZQDCT) coefficients, but HEVC encoder uses DST instead of DCT for 4 × 4 intra luma blocks. So analysis proposed in [11] cannot be applied to HEVC encoded videos due to differences in energy compaction properties of these two transforms. The differences in energy compaction properties of DST and DCT as explained in [12] are: (1) DCT generates DC coefficient, which is the average value of the coefficients present in a block, but DST does not generate any DC coefficient and (2) In DCT transform, the values generated next to DC coefficient are called AC coefficients whose stability decreases as we move from low frequency to high frequency coefficients but DST coefficients do not follow any such phenomenon. An example of QDST coefficients of 4 × 4 intra luma block is shown in Table 1.

The imperceptibility of the scheme is achieved by selecting 4×4 intra luma blocks having more textured regions. The texture of the block is analysed by number of non-zero QDST coefficients present in a 4 × 4 intra luma block. The blocks with dense textured region will have more non-zero QDST coefficients. Mansouri et al. in [11] used quality threshold α which is calculated on the basis of number of non-zero QDCT AC coefficients of 4 × 4 intra luma block, but in HEVC DST is used instead of DCT in 4 × 4 blocks. There are no DC and AC coefficients in DST, so the quality threshold α is calculated on the basis of number of all non-zero QDST coefficients present in a 4 × 4 block.

The robustness of the proposed scheme is based on robustness threshold β i.e. based on the number of QDST coefficients in a 4 × 4 intra luma block with magnitude greater than one. The 4×4 blocks with more coefficients having magnitude greater than one means there is more irregular texture with minute details. The minute details can be accurately predicted from neighbouring blocks using the angular intra prediction process only when these types of blocks are partitioned into smaller blocks. So, there are very less chances that the size of these types of blocks will be changed to bigger size. Additionally, changes in the angular intra prediction mode to another mode are very less for areas with minute details. This results in stability in the relationship between the positive and negative QDST coefficients present in a block. To demonstrate this effect, we applied re-compression to several standard non watermarked video sequences, and then estimated the rate of changes with respect to different numbers of coefficients with absolute value greater than one (ABGR1). The cases of rate of changes includes (a) change in relationship of the sum of magnitudes of positive and sum of magnitudes of negative QDST coefficients and (b) change of size of 4 × 4 block to bigger size.

So, the value of beta is calculated on the basis of the number of QDST coefficients with ABGR1 present in a 4 × 4 intra luma block. Figure 1 shows the rate of change *w.r.t.* ABGR1 when different video sequences are re-compressed from Quantization Parameters (QP) 32 to 34. It can be observed from Fig. 1 that the rate of change decreases with increase in number of coefficients with ABGR1 present in a 4 × 4 block. The number of ABGR1 coefficients varies from one 4 × 4 blocks to another based on the texture of the I-frame in a

Table 1 QDST coefficients of 4 × 4 intra luma block in I-frame

0	1	1	0
4	1	0	0
−4	−3	−1	0
−1	1	0	0

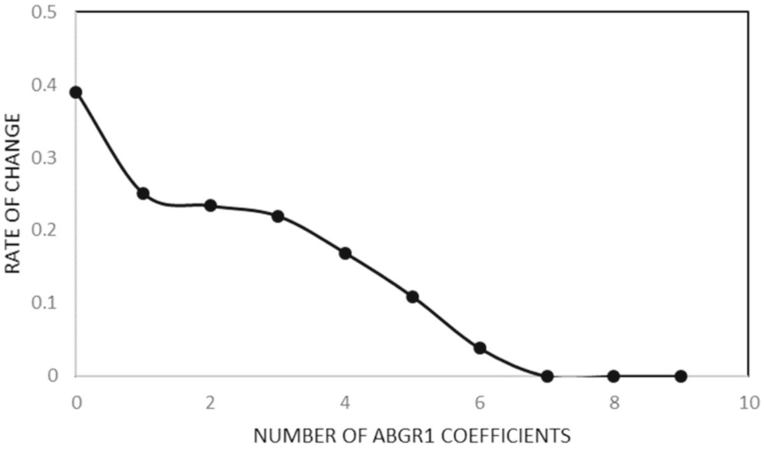


Fig. 1 Rate of change of 4 × 4 blocks when re-compression is done from Qp = 32 to Qp = 34

video sequence. Further, distribution of 4 × 4 blocks with a certain value of the number of QDST coefficients with ABGR1 varies on the basis of the texture of the video sequence. So, the optimal value of β is calculated at runtime based on texture of video sequence and number of blocks required to embed watermark bits. A survival function based on Complementary Cumulative Frequency Distribution (CCDF) of number of 4 × 4 blocks with different values of the number of coefficients with ABGR1 $\overline{F}$ is calculated using (1) and (2) respectively.

$$\overline{F}(\beta) > \mu \tag{1}$$

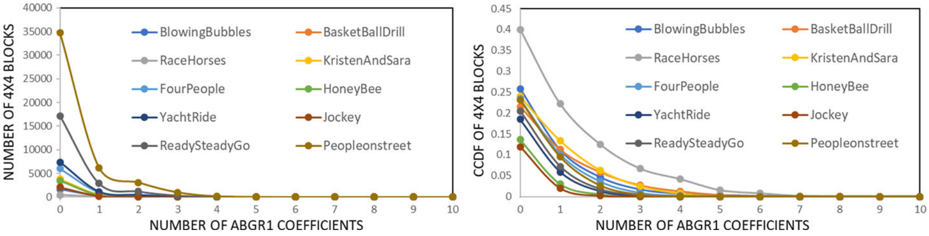
The value of $\overline{F}$ is calculated based on probability distribution (D) of number of 4 × 4 blocks with different values of ABGR1 coefficients, using the formula [11] which is defined as follows:

$$\overline{F}(\beta) = P(D > \beta) = 1 - F = \sum_{D>\beta} p(D) \tag{2}$$

Figure 2a shows distribution of 4 × 4 blocks with respect to number of ABGR1 coefficients in first I frame of video sequences with different textures. Figure 2b shows $\overline{F}$ of different video sequences with QP = 32. As it can be observed, for the same value of μ different values of β are obtained based on textures of different videos. For example, when value $\mu = 0.02$, the value of β for PeopleonStreet is 2 and RaceHorses is 3. Thus, the optimal value of β is derived from μ based on textures of videos at runtime. By implementing the above two spatial analysis for imperceptibility and robustness, the proposed scheme attains desired requirements of imperceptibility and robustness against re-compression.

2.2 Authentication code generation

An authentication code is generated from invariant features present in I-frames during encoding of video sequences. The 4 × 4 intra luma blocks in I-frame are divided into two disjoint sets, i.e. watermark generation region S_g and watermark embedding region S_e. These sets are generated based on the difference between the sum of magnitudes of



(a) (b)

Fig. 2 **a** Distribution of 4 × 4 blocks with respect to ABGR1 coefficients in first I frame of video sequences at Qp = 32, **b** CCDF of 4 × 4 blocks in first I frame

Positive Quantized Discrete Sine Transform (PQDST) Coefficients and sum of magnitudes of Negative Quantized Discrete Sine Transform (NQDST) coefficients. This is done to avoid interference between S_g and S_e regions, so that exact authentication code can be re-generated at the decoder end. A block belongs to S_e if the difference is in the range of -1 to 1, otherwise it belongs to S_g region. The flowchart of authentication code generation is given in Fig. 3 and the comprehensive description of the authentication code generation is explained in Algorithm 1. In this algorithm, the relationship between the number of PQDST coefficients and number of NQDST coefficients is used to generate authentication code. The outputs of this algorithm are palette PG_k and authentication code A_k, where k is the number of CTU's in an I-frame of a video sequence. The PG_k will be used by decoder to re-generate A_k during the verification process. The palette, PG_k is a file which contains the location of blocks used for authenticate code generation. It is used to avoid de-synchronization of code extraction and verification at the decoder end [11]. So, this scheme is semi-blind as instead of full original video only palette is required at the decoder end for extraction and verification of the authentication code. The palette can easily be transmitted through a secure channel.

To escalate the security of the proposed scheme, selection of blocks from S_g region is done using a pseudo random function for generating authentication code. The generated authentication code A_k is encrypted using a low complexity encryption function F_E and the content based key (Key) to generate watermark sequence W_k. The Key is used to encrypt generated authentication code to circumvent the intra collusion attack [6]. First, a public key is generated by pseudo randomly selecting a 16 × 16 intra luma block in each I-frame [5]. The DC coefficient and first two nonzero AC coefficients each of 8 bits are extracted from the selected 16 × 16 block. The public key which is a sequence of 24 bits is generated by concatenating these three coefficients (DC and two AC coefficients). The public key is further encrypted using pseudo randomly generated private key (a 24 bit pseudo-random number). The resultant key is used for encrypting A_k using a low complexity encryption function F_E to generate watermark i.e. W_k. The watermark sequence W_k is generated using (3). The private key and location of block from which public key is generated is written in palette along with location map of blocks. The palette is generated at the encoder end during authentication code generation and embedding process and is transmitted to the authorized user through secured channel. At the decoder end, location of block used for generating public key and private key is extracted from the palette. Then the public key is generated from DC and first two non-zero AC coefficients. Then this public key is encrypted using

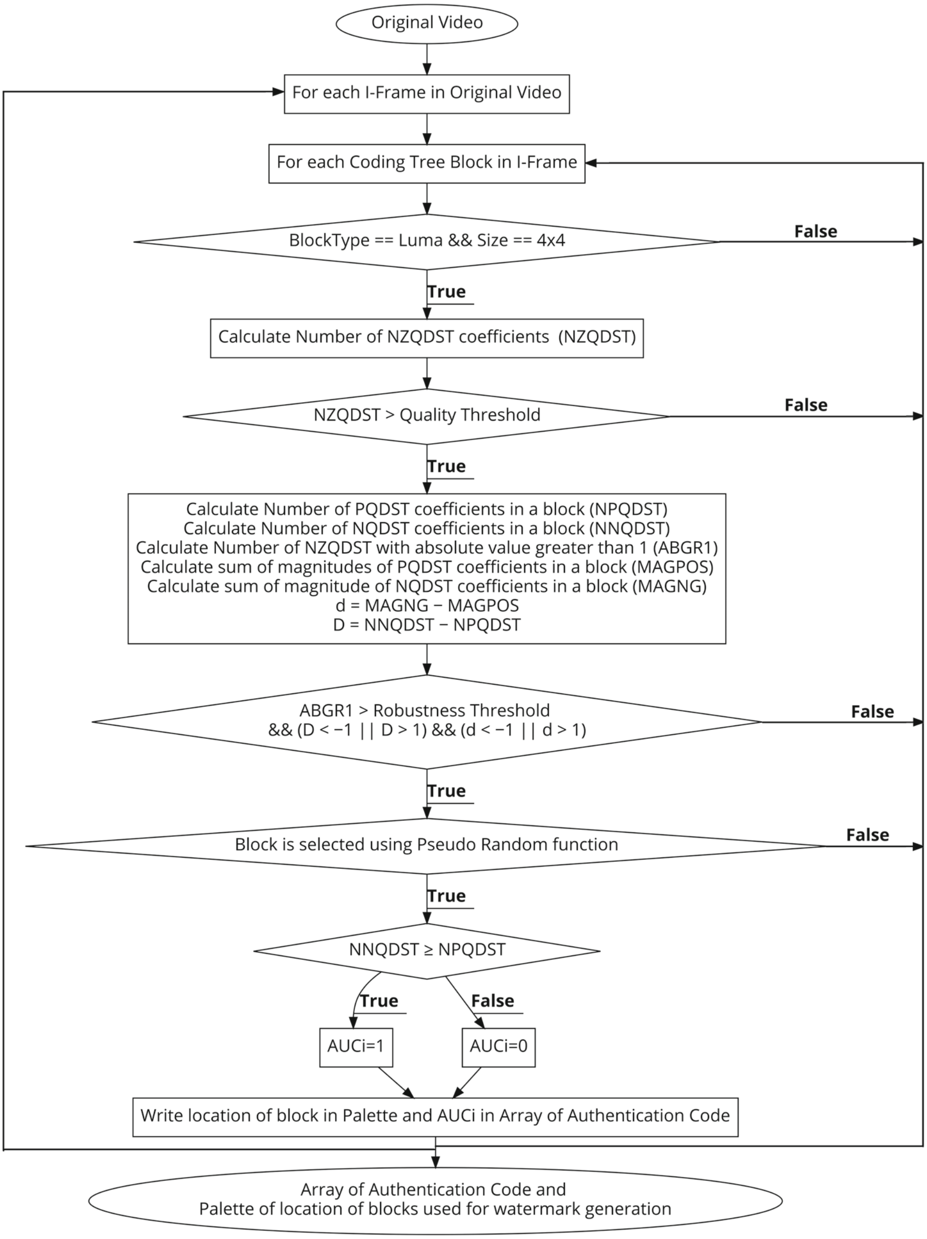

Original Video
For each I-Frame in Original Video
For each Coding Tree Block in I-Frame
BlockType == Luma && Size == 4x4
False
True
Calculate Number of NZQDST coefficients (NZQDST)
NZQDST > Quality Threshold
False
True
Calculate Number of PQDST coefficients in a block (NPQDST)
Calculate Number of NQDST coefficients in a block (NNQDST)
Calculate Number of NZQDST with absolute value greater than 1 (ABGR1)
Calculate sum of magnitudes of PQDST coefficients in a block (MAGPOS)
Calculate sum of magnitude of NQDST coefficients in a block (MAGNG)
d = MAGNG − MAGPOS
D = NNQDST − NPQDST
ABGR1 > Robustness Threshold
&& (D < −1 || D > 1) && (d < −1 || d > 1)
False
True
Block is selected using Pseudo Random function
False
True
NNQDST ≥ NPQDST
True
False
AUCi=1
AUCi=0
Write location of block in Palette and AUCi in Array of Authentication Code
Array of Authentication Code and
Palette of location of blocks used for watermark generation

Fig. 3 Flowchart of authentication code generation

the private key. The resultant key is used to decrypt authentication code for extraction and verification.

$$W_k = F_E(A_k, Key) = \{W_1, W_2, .., W_k\} \tag{3}$$

Algorithm 1 Authentication code generation.

```
Data: Original Video
Result: Array of Authentication Code $AUC_k$, Palette of location of blocks used for
        watermark generation $PG_k$
foreach Frame in Original Video do
    if I Frame then                                   /* If Frame Type is Intra */
        foreach Coding Tree Block in an I Frame do
            if $BlockType == Luma \&\& Size == 4 \times 4 \&\& Block \in S_g$ then
                Calculate NZQDST  /* Number of NZQDST coefficients */
                if $NZQDST > \alpha$ then   /* $\alpha$ is quality threshold */
                    Calculate NPQDST          /* Number of PQDST coefficients in a block */
                    Calculate NNQDST          /* Number of NQDST coefficients in a block */
                    Calculate $ABGR1$         /* Number of NZQDST with absolute value greater than 1 */
                    Calculate MAGPOS  /* sum of magnitudes of PQDST coefficients in a block */
                    Calculate MAGNG   /* sum of magnitude of NQDST coefficients in a block */
                    $d = MAGNG - MAGPOS$
                    $D = NNQDST - NPQDST$
                    if $ABGR1 > \beta \&\& (D < -1 \parallel D > 1) \&\& (d < -1 \parallel d > 1)$
                    then            /* $\beta$ is robustness threshold */
                        if Block is selected using Pseudo random function then
                            if $NNQDST \geq NPQDST$ then
                                $AUC_i = 1$
                            else
                                $AUC_i = 0$
                            Add location of block in Palette $PG_k$
                        else
                            Go to next block if block is not selected by pseudo
                            random function
```

2.3 Watermark embedding

The generated watermark W_k is embedded into 4×4 intra luma blocks which are selected on the basis of spatial analysis explained in Section 2.1. The security of watermark is provided by selecting candidate blocks from the S_e region pseudo randomly. In the proposed scheme watermark is embedded based on the relationship between the sum of magnitude of PQDST coefficients and sum of magnitude of NQDST coefficients present in 4×4 intra luma block. The blocks that are to be used for watermark embedding are selected from the S_e regions based on quality threshold α i.e. NNZQDST coefficients present in a block and robustness threshold β i.e. number of coefficients with ABGR1 present in a block. These thresholds are selected based on spatial analysis explained in Section 2.1. Further, a subset of candidate blocks is pseudo randomly selected from the S_e region. The flowchart of the process of candidate block selection is given in Fig. 4 and the comprehensive description of the candidate block selection is explained in Algorithm 2. This algorithm generates a palette PE_k that

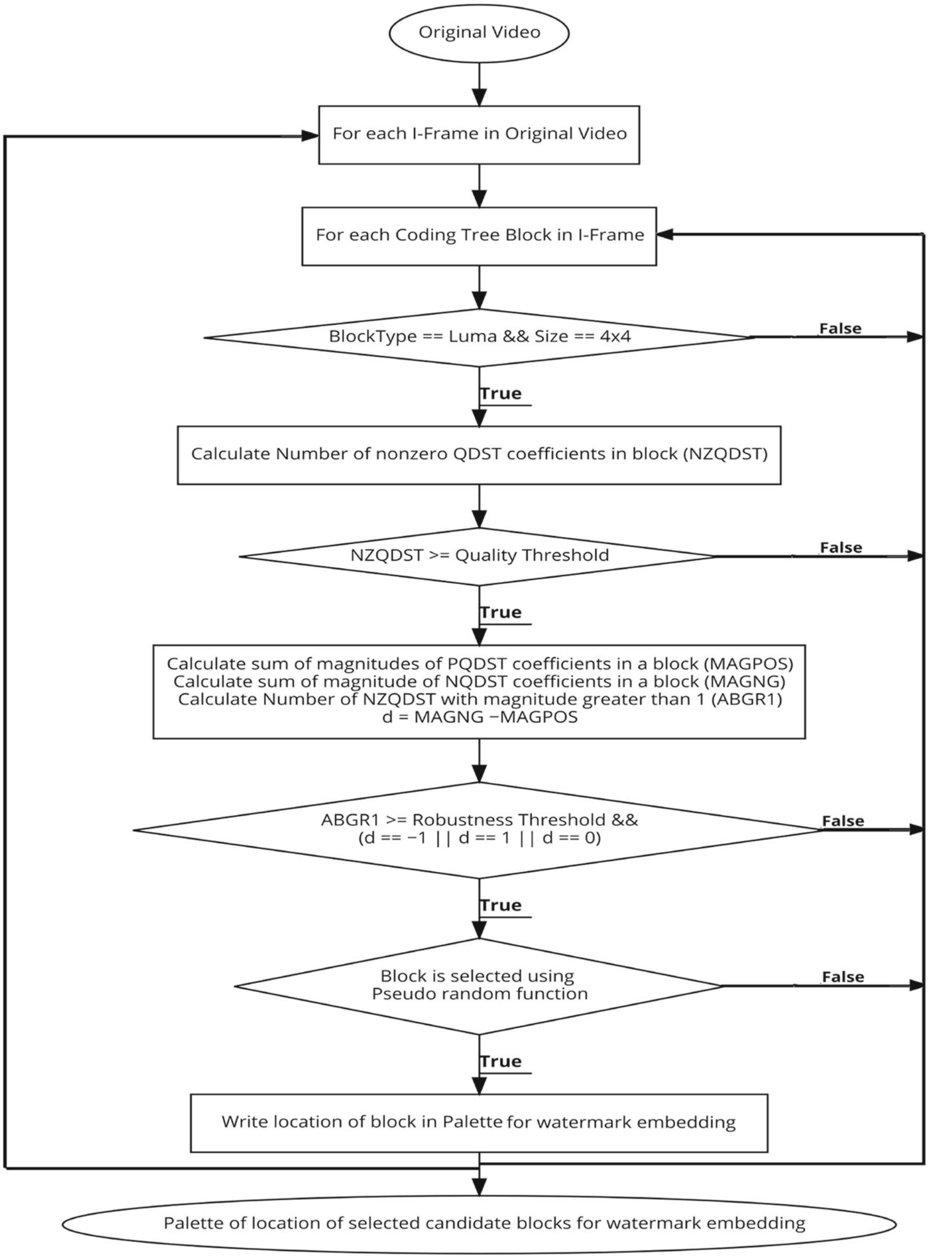

Original Video
For each I-Frame in Original Video
For each Coding Tree Block in I-Frame
BlockType == Luma && Size == 4x4
False
True
Calculate Number of nonzero QDST coefficients in block (NZQDST)
NZQDST >= Quality Threshold
False
True
Calculate sum of magnitudes of PQDST coefficients in a block (MAGPOS)
Calculate sum of magnitude of NQDST coefficients in a block (MAGNG)
Calculate Number of NZQDST with magnitude greater than 1 (ABGR1)
d = MAGNG −MAGPOS
ABGR1 >= Robustness Threshold &&
(d == −1 || d == 1 || d == 0)
False
True
Block is selected using
Pseudo random function
False
True
Write location of block in Palette for watermark embedding
Palette of location of selected candidate blocks for watermark embedding

Fig. 4 Flowchart of candidate blocks selection for embedding watermark

contains the locations of the candidate blocks which are selected for embedding generated watermark sequence W_k. This palette, PE_k, will be used by Algorithm 3 for embedding the watermark W_k and Algorithm 4 for extraction and verification of the embedded watermark at decoder end.

Algorithm 2 Selection of candidate blocks.

```
Data: Original Video
Result: Palette of location of selected candidate blocks for watermark embedding $PE_k$
foreach Frame in Original Video do
    if I Frame then                    /* If Frame Type is Intra */
        foreach Coding Tree Blocks In Frame do
            if $BlockType == Luma$ && $Size == 4 \times 4$ && $Block \in S_e$ then
                Calculate NZQDST        /* Number of nonzero QDST coefficients in block */
                if $NZQDST \geq \alpha$ then   /* $\alpha$ is quality threshold */
                    Calculate MAGPOS  /* sum of magnitudes of PQDST coefficients in a block */
                    Calculate MAGNG   /* sum of magnitude of NQDST coefficients in a block */
                    Calculate $ABGR1$       /* Number of NZQDST with magnitude greater than 1 */
                    $d = MAGNG - MAGPOS$
                    if $ABGR1 \geq \beta$ && $(d == -1 \| d == 1 \| d == 0)$ then /* $\beta$ is robustness threshold */
                        if Block is selected using Pseudo random function then
                            Add location of block in Palette $PE_k$ for watermark embedding
                        else
                            Go to next block if block is not selected by pseudo random function
```

Watermark bits are embedded into QDST coefficients during the last stage of I-frame encoding. This embedding creates the problem of accumulation of drift as the changes made in QDST coefficients in current block can propagate to the neighbouring blocks during the prediction process at decoder end. To solve this problem, QDST coefficients of 4 × 4 blocks are divided into two sets, i.e. first set of coefficients which will be used for watermark embedding and second set which will not be used for watermark embedding. Table 2 shows indices of the QDST coefficients present in 4 × 4 block. The 7 indices

Table 2 Indices of QDST coefficients in a 4 × 4 block

C_{00}	C_{01}	C_{02}	C_{03}
C_{10}	C_{11}	C_{12}	C_{13}
C_{20}	C_{21}	C_{22}	C_{23}
C_{30}	C_{31}	C_{32}	C_{33}

$\{C_{i,3}\}_{i=0,1,2,3} \bigcup \{C_{3,j}\}_{j=0,1,2,3}$ belong to protected set from a total of 16 indices of a 4×4 block which are not used for embedding watermark bits to avoid I-frame error drift problem [1].

The flowchart of the watermark embedding process is given in Fig. 5 and is explained in detail in Algorithm 3. The inputs to this algorithm are the palette of selected candidate blocks PE_k and watermark W_k i.e. generated using Algorithm 1 as discussed is Section 2.2. The watermark bit is embedded on the basis of the magnitudes of PQDST and NQDST, so to embed watermark bit only non-zero QDST coefficients are altered to restrict the increase in bit-rate of watermarked video sequence. Further, watermark is embedded only in those blocks in which difference between MAGPOS and MAGNG is -1, 0 or 1. This is to avoid visual distortion as by restricting to this difference at most magnitudes of two QDST coefficients are altered in a block.

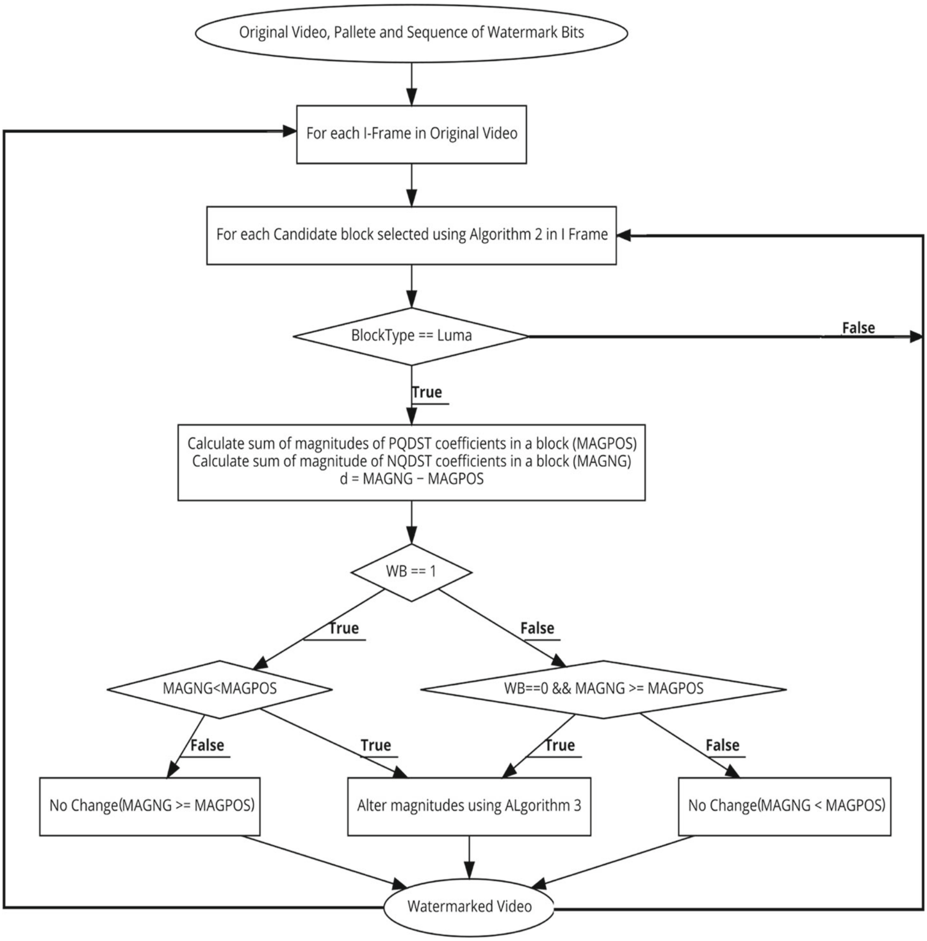

Original Video, Pallete and Sequence of Watermark Bits
For each I-Frame in Original Video
For each Candidate block selected using Algorithm 2 in I Frame
BlockType == Luma
False
True
Calculate sum of magnitudes of PQDST coefficients in a block (MAGPOS)
Calculate sum of magnitude of NQDST coefficients in a block (MAGNG)
d = MAGNG − MAGPOS
WB == 1
True
False
MAGNG<MAGPOS
WB==0 && MAGNG >= MAGPOS
False
True
True
False
No Change(MAGNG >= MAGPOS)
Alter magnitudes using ALgorithm 3
No Change(MAGNG < MAGPOS)
Watermarked Video

Fig. 5 Flowchart for embedding of watermark

Algorithm 3 Watermark embedding algorithm.

```
Data: Palette of location of selected candidate blocks for watermark embedding $PE_k$
      and $W_k$(Sequence of Watermark Bits(WB)
Result: Watermarked Video
foreach Frame in Original Video do
    if I Frame then                          /* If Frame Type is Intra */
        foreach Candidate block selected using Algorithm 2 in I Frame do
            if $BlockType == Luma$ then
                Calculate MAGPOS     /* sum of magnitudes of PQDST
                coefficients in a block */
                Calculate MAGNG      /* sum of magnitude of NQDST
                coefficients in a block */
                $d = MAGNG - MAGPOS$
                if $WB == 1\&\&MAGNG >= MAGPOS\|WB ==$
                $0\&\&MAGNG < MAGPOS$ then
                    No change
                else
                    if $WB == 1\&\&MAGNG < MAGPOS$ then
                        while $MAGNG < MAGPOS$ do
                            for $i <= 3$ do
                                for $j <= 3$ do
                                    if $C_{ij} < -1\&\&C_{ij}$ does not belong to protected
                                    set of pixels then
                                        Increase magnitude of $C_{ij}$ by one
                    else if $WB == 0\&\&MAGNG >= MAGPOS$ then
                        while $MAGNG >= MAGPOS$ do
                            for $i <= 3$ do
                                for $j <= 3$ do
                                    if $C_{ij} > 1\&\&C_{ij}$ does not belong to protected
                                    set of pixels then
                                        Increase magnitude of $C_{ij}$ by one
```

2.4 Watermark extraction and verification

The watermark extraction and verification is performed at decoder side after entropy decoding of QDST coefficients of watermarked video sequence. The watermark extraction process is exactly reverse of watermark embedding and the flowchart of watermark extraction process is given in Fig. 6. The QDST coefficients are entropy decoded for two purposes (1) to regenerate authentication code A'_k (2) extraction of the watermark W'_k embedded during encoding of the video sequence. The content based authentication code A'_k is re-generated using Algorithm 1 and palette PG_k. The watermark W'_k is extracted

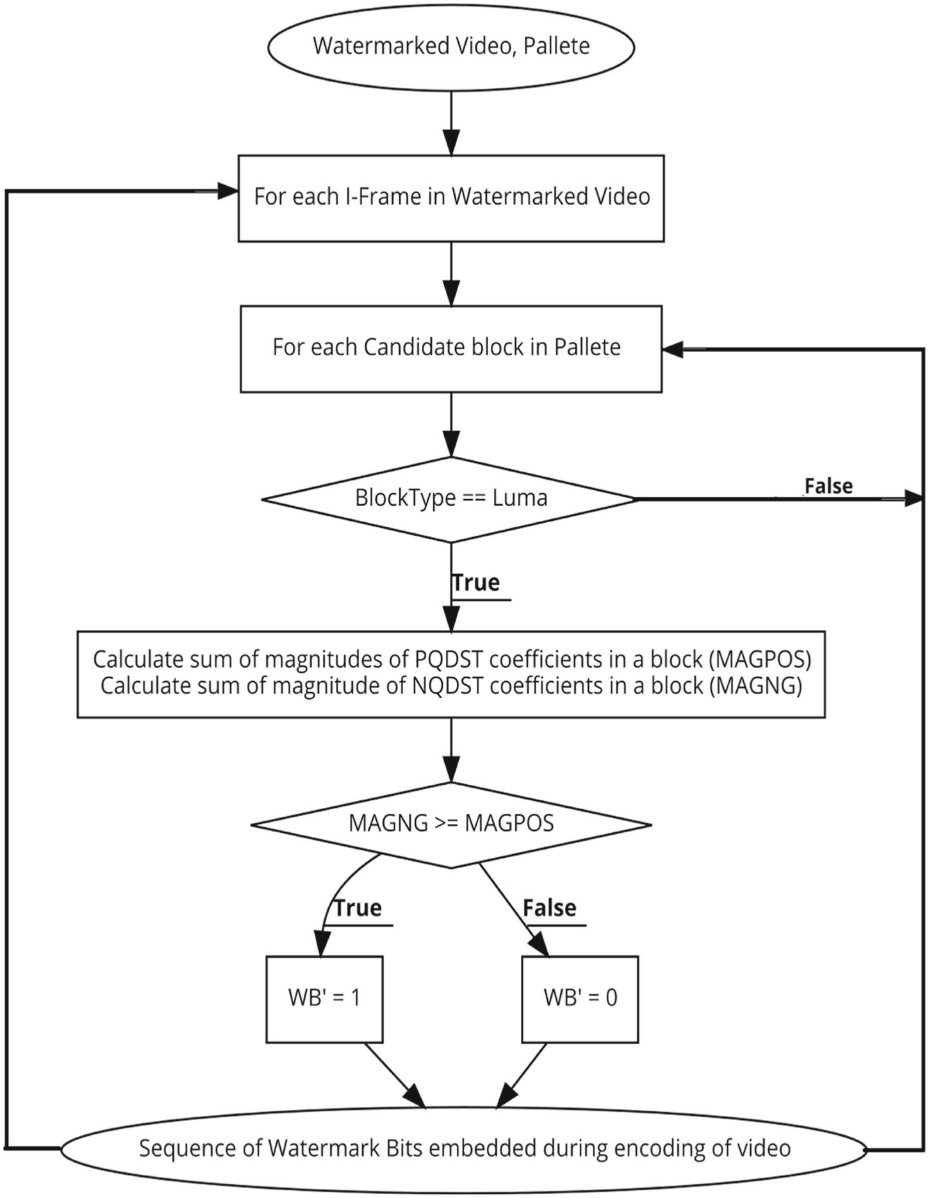

Watermarked Video, Pallete
For each I-Frame in Watermarked Video
For each Candidate block in Pallete
BlockType == Luma
False
True
Calculate sum of magnitudes of PQDST coefficients in a block (MAGPOS)
Calculate sum of magnitude of NQDST coefficients in a block (MAGNG)
MAGNG >= MAGPOS
True
False
WB' = 1
WB' = 0
Sequence of Watermark Bits embedded during encoding of video

Fig. 6 Flowchart for extraction of watermark

using Algorithm 4 and the palette PE_k generated using Algorithm 2. The extracted watermark W_k' is decrypted using decryption function F_D which is reverse of encryption function used at encoder end and Key (key generated at encoder end) to extract the authentication code A_k''.

$$A_k'' = F_D(W_k', Key) \tag{4}$$

The verification process of each I-Frame in a video sequence is done by comparing the extracted authentication code A''_k at decoder end and the re-generated authentication code A'_k at decoder end by using (5).

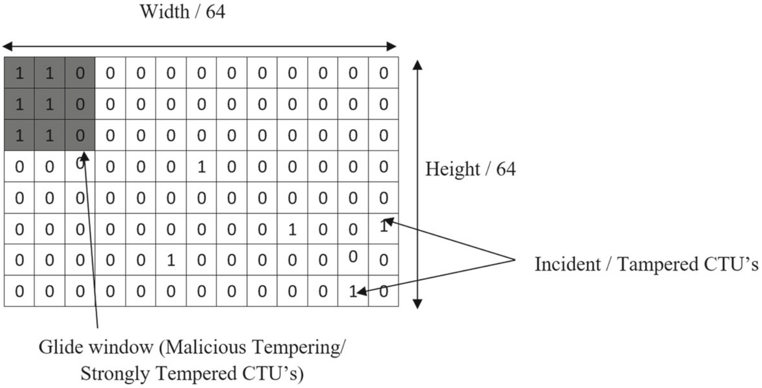
$$\begin{aligned} &if\ A'_k \oplus A''_k == 1,\ I - Frame\ is\ Unauthentic \\ &if\ A'_k \oplus A''_k == 0,\ I - Frame\ is\ Authentic \end{aligned} \tag{5}$$

Based on the (5), generate an Attack Identification Matrix (AIM) for each $I - Frame_i$ (where i varies from 1 to n and n is total number of I-frames in a video sequence), in which 1 denotes authentic CTU and 0 denotes unauthentic CTU. In [25], a gliding window of 3 × 3 and a tamper detection threshold T_γ (sum of 1's in a gliding window) is used to detect maliciously tampered area in an $I - Frame_i$. The value of T_γ is set to 5 based on theory of probability using binomial distribution. In another algorithm [5], a median filter is applied on generated AIM for two purposes: (1) To remove scattered incidental 1's and (2) Treat non-error elements i.e. 0 as error element i.e. 1 if 0 is surrounded by five or more 1's to ensure clustering to locate malicious tampering. As found in [5, 25], tampered locations in AIM are spread in two ways: (a) In case of re-compression and noise attack, the 1's are distributed in the form of random noise and (b) In case of malicious tampering, 1's are accumulated in some locations. So,we are using algorithms of [25] and [5] for tamper detection. An Example of AIM 3 × 3 gliding window is shown in Fig. 7.

By observing the AIM in Fig. 7 the tampered CTU's (1's) inside the glide window are called Strongly Tampered Elements (STE) because they are clustered, on the other hand the 1's outside the gliding window are called Incidental/Tampered Elements (TE) because they are mostly scattered. The size of the glide window and the T_γ i.e. number of 1's inside glide window are selected empirically in [25] based on application requirement i.e. authentication sensitivity.

Following are the steps for malicious tamper detection as in [5]:

1. Calculate Tampered Percentage (TP) i.e. sum of tampered CTU's divided by total number of CTU's in AIM.



1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0

Fig. 7 Attack Identification Matrix (AIM) and 3 × 3 glide window

2. Apply 3 × 3 median filter on AIM.
3. Calculate Strongly Tampered Percentage (STP) i.e. the sum of all strongly tampered elements by the total number of tampered elements in processed AIM.
4. Compare the STP with a malicious tampering threshold T_m as follows:

$$\begin{cases} I - Frame_i \text{ is unauthnetic, if STP} \geq T_m \\ I - Frame_i \text{ is authnetic, if STP} < T_m \end{cases} \tag{6}$$

$$where\ P(T_m \geq \Omega) = 1 - P(T_m < \Omega) \approx 1 - \left(\left(\frac{E(STE)}{\sqrt{Var(STE)}}\right)/E(TE)\right) \tag{7}$$

We know that the value of each element in the AIM can be either 1 or 0, then the probability of each element is 0.5. Hence, it can be considered as a random variable which follows binomial probability distribution. Assumed that for any random element in AIM, the probability of an error can happen is 0.5. Similarly, the probability of any five or more random errors can occur in a glide window of size $n \times n$ is calculated using cumulative probability as follows :

$$P(i \geq 5) = \sum_{i=5}^{n \times n} \binom{n \times n}{i} (0.5)^i (0.5)^{n \times n - 1} \tag{8}$$

Where value of n is 3, the calculated value of (8) is 0.5. Then, the expected value of STE i.e. E(STE) and variance of STE i.e. Var(STE) are 0.5 × *numel* and 0.5 × (1 − 0.5) × *numel* respectively, where *numel* denotes the number of tampered elements in AIM. The probability of any element to detected as tampered in a gliding window is $0.5(1 - (0.5)^8) = 0.498$. Then the expected value of tampered elements i.e. E(TE) is 0.498 × *numel*.

Algorithm 4 Watermark extraction algorithm.

```
Data: Watermarked Video, Palette of location of selected candidate blocks for
      watermark embedding $PE_k$
Result: $W_k'$ (Sequence of Watermark Bits(WB) embedded during encoding of video)
foreach Frame in Watermarked video do
    if I Frame then                         /* If Frame Type is Intra */
        foreach Candidate block in palette $PE_k$ do
            if $BlockType == Luma$ then
                Calculate MAGPOS      /* sum of magnitudes of PQDST
                coefficients in a block */
                Calculate MAGNG        /* sum of magnitude of NQDST
                coefficients in a block */
                if $MAGNG >= MAGPOS$ then
                    $WB' = 1$
                else
                    $WB' = 0$
```

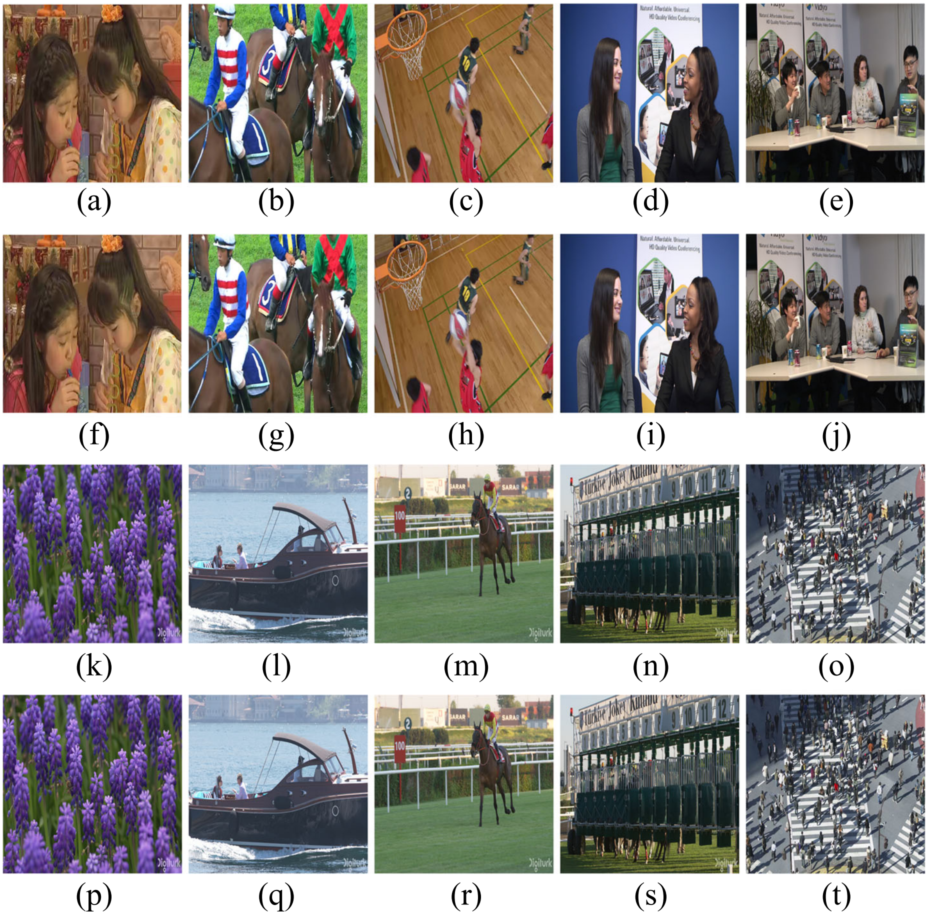
3 Experimental results

To evaluate and validate the proposed authentication scheme is implemented by using HEVC reference software HM16 [7] on a system with Intel i7-8550U CPU @1.80 GHz, 8G RAM and Windows 10 operating system. Experimental results are obtained by using the intra main configuration with IntraPeriod=8, DecodingRefereshType=2, GOPSize=4, QP =32, IPPPPPPPI closed GOP structure and the other all parameters remains as the default configuration. To show the efficiency of the proposed authentication scheme, various video sequences with different resolutions and textures are tested. The details of different video sequences for testing are given in Table 3. The performance of the proposed scheme is evaluated in terms of visual imperceptibility, Bit Increase Rate (BIR), computational complexity, robustness against re-compression, addition of noise, frame dropping and fragile to malicious attacks. There is an intersecting relationship between the robustness, visual quality, embedding capacity and bit rate. A higher embedding capacity results in high visual distortion, BIR and decrease in robustness against attacks and vice versa. As a fundamental requirement of authentication scheme is transparency, robustness against re-compression and fragile against malicious attacks. So to acquire these requirements and to overcome the above trade-off, we restricted embedding capacity by bounding the number of bits embedded into each CTU to one. Further, the required transparency and robustness is obtained by using quality threshold α and robustness threshold β respectively. The values of these thresholds vary based on the texture of different video sequences and are calculated at run-time by using CCDF as explained in Section 2.1. The higher the value of α and β results in high robustness and low visual distortion, but it decreases the candidate blocks for generating and embedding watermark bits and vice versa. So based on experimental results of different video sequences the optimal value of the CCDF is set to 0.1 for α and 0.01 for β obtaining the optimal values at run time. The proposed scheme uses the transformed coefficients of 4 × 4 intra luma blocks for the authentication process. The performance of the proposed scheme is compared with the schemes of Chang et al. [1], Dutta et al. [3] and Liu et al. [10] because these schemes are also based on transformed coefficients of 4 × 4 intra luma blocks.

The visual quality of the proposed scheme is evaluated in terms of subjective and objective measurements. For subject evaluation, Fig. 8 shows original and watermarked frames of different video sequences. The Fig. 8a, c, b, d, e, k, l, m, n and o show the original frames

Table 3 Configuration parameters of video sequences used for experimentation

Serial number	Video sequence	Resolution	Number of frames	Frame rate
1	BlowingBubbles	416 × 240	500	50
2	BasketBallDrill	832 × 480	500	50
3	RaceHorses	832 × 480	500	30
4	KristenAndSara	1280 × 720	600	60
5	FourPeople	1280 × 720	600	60
6	HoneyBee	1920 × 1080	600	120
7	YachtRide	1920 × 1080	600	120
8	Jockey	1920 × 1080	600	120
9	ReadySteadyGo	1920 × 1080	600	120
10	PeopleOnStreet	2560 × 1600	150	30



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

Fig. 8 Original and watermarked frames of different video sequences with QP=32

of tested video sequences. The Fig. 8f, g, h, i, j, p, q, r, s and t show the watermarked frames of tested video sequences. So, it can be observed from the Fig. 8, there are no visual artifacts in the watermarked frames of video sequences. In addition to subjective evaluation, Peak Signal to Noise Ratio ($PSNR$), Structural SIMilarity index ($SSIM$) and Visual Information Fidelity in Pixel Domain ($VIfp$) are used as objective visual quality measurements. The $PNSR$ for YUV video sequence is calculated using (9) given in [18]. The difference in the $PSNR$ of original and watermark video is calculated using (10). The δ_{PSNR} of the proposed scheme and the schemes in [1, 3] and [10] are shown in Table 4.

$$PSNR_{YUV} = \frac{6 \times PSNR_Y + PSNR_U + PSNR_V}{8} \tag{9}$$

where $PSNR_Y$ is $PSNR$ of luma component, $PSNR_U$ is $PSNR$ of chroma Cb and $PSNR_V$ is $PSNR$ of Chroma Cr.

$$\delta_{PSNR} = PSNR_{YUV} - PSNR^{'}_{YUV} \tag{10}$$

Table 4 Comparison of proposed scheme with different schemes in terms of δ_{PSNR}, $SSIM$ and $VIfp$

Video sequence name	Proposed scheme			Chang et al. [1]			Dutta et al. [3]			Liu et al. [10]		
	δ_{PSNR}(dB)	$SSIM$	$VIfp$	δ_{PSNR}(dB)	$SSIM$	$VIfp$	δ_{PSNR}(dB)	$SSIM$	$VIfp$	δ_{PSNR}(dB)	$SSIM$	$VIfp$
BlowingBubbles	0.1054	0.99252	0.93885	0.1542	0.99203	0.91068	0.1941	0.99002	0.90217	0.1697	0.99173	0.91022
BasketBallDrill	0.1179	0.99259	0.96648	0.1749	0.99213	0.95212	0.2141	0.99012	0.94361	0.1876	0.99159	0.95182
RaceHorses	0.1329	0.99319	0.96791	0.1966	0.99303	0.95425	0.2397	0.99102	0.94574	0.2024	0.99217	0.95335
KristenAndSara	0.1186	0.99344	0.98518	0.1886	0.99341	0.98017	0.2317	0.99142	0.97166	0.1971	0.99197	0.97964
FourPeople	0.1391	0.99284	0.97356	0.1951	0.99252	0.96274	0.2382	0.99051	0.95423	0.1983	0.99086	0.96209
HoneyBee	0.1474	0.99231	0.95806	0.2136	0.99172	0.93949	0.2567	0.98971	0.93098	0.2253	0.99133	0.93852
YachtRide	0.1284	0.99237	0.96695	0.2027	0.99181	0.95282	0.2458	0.98981	0.94431	0.2218	0.99125	0.95215
Jockey	0.1414	0.99194	0.96293	0.2041	0.99116	0.94679	0.2472	0.98915	0.93828	0.2189	0.99017	0.94629
ReadySteadyGo	0.1301	0.99295	0.96884	0.1969	0.99268	0.95566	0.2402	0.99067	0.94715	0.2066	0.99219	0.95493
PeopleOnStreet	0.0919	0.99326	0.97157	0.1471	0.99314	0.95976	0.1902	0.99113	0.95125	0.1546	0.99255	0.95894
Average	0.1253	0.99274	0.96603	0.1874	0.99236	0.95145	0.2298	0.99035	0.94294	0.1982	0.99158	0.95079

where $PSNR_{YUV}$ and $PSNR'_{YUV}$ are $PSNR$ of original and watermarked video sequences respectively calculated using (9).

The $SSIM$ and $VIfp$ [15, 21] are also used for evaluating visual quality of watermarked video sequences since $PSNR$ does not reflect the temporal activity of video sequences with different textures. The value of $SSIM$ and $VIfp$ index varies from 0 to 1, where 1 indicates that both the compressed video sequence with watermark and without watermark are completely identical on the other hand 0 indicates both video sequences are entirely different from each other. Table 4 shows that the results of $SSIM$ and $VIfp$ values for different video sequences for the proposed scheme are better than the schemes in [1, 3] and [10]. These results show that the performance of the proposed scheme is better than the schemes in [1, 3] and [10] in terms of imperceptibility because 4 × 4 intra luma blocks which are used for watermark embedding are selected on the basis of quality threshold α i.e. calculated using the number of non-zero QDST coefficients. So, based on the subjective evaluation in Fig. 8 and the objective evaluation in terms of $PSNR$, $SSIM$ and $VIfp$, the proposed scheme is visually imperceptible, which meets the demand of video authentication system.

Bit-rate is the total number of bits used for encoding a video during compression. In the proposed scheme, the watermark bits are embedded by altering only non-zero QDST coefficients and by preserving signs of the coefficients, which in return induce a very slight increase in bit-rate of encoded videos after watermark embedding. The performance of the proposed scheme is evaluated in terms of BIR which is calculated by using (11) where BR and BR' are bit-rates of original and watermarked videos respectively. BIR of the different video sequences for the proposed scheme and the schemes in [1, 3] and [10] are shown in Fig. 9 with QP = 32. It can be observed from the results that BIR of the proposed scheme is better than the schemes in [1, 3] and [10].

$$BIR = \frac{BR' - BR}{BR} \times 100 \tag{11}$$

Firstly, the performance of the proposed scheme is evaluated in terms of robustness against content-preserving attacks, i.e. robustness against re-compression with different Qp values, noise attacks and frame dropping attacks. The robustness is evaluated in terms of correlation between the extracted watermark bits and embedded watermark bits. Secondly, the performance of the scheme is evaluated in terms of sensitivity to the malicious tampering.

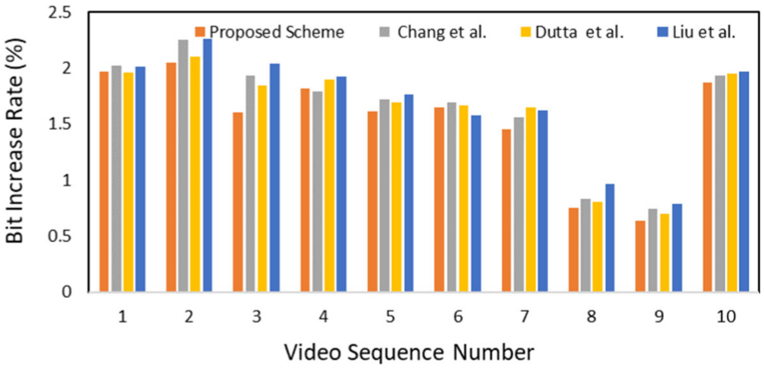


Fig. 9 Comparison of the BIR for the proposed scheme with different schemes

Table 5 Correlation comparison of proposed scheme with different schemes in terms of re-compression attack

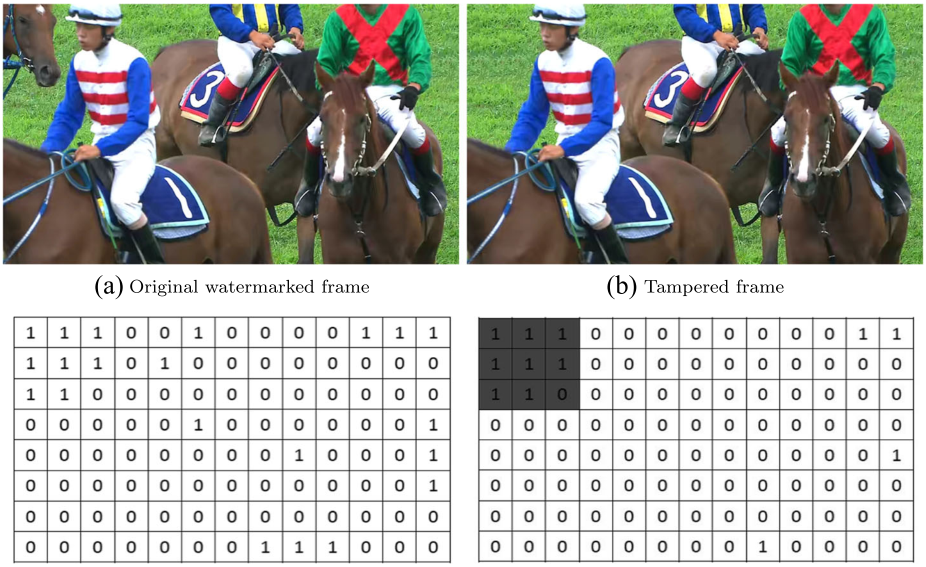
Video sequence	Proposed scheme			Chang et al. [1]			Dutta et al.[3]			Liu et al.[10]		
	Re-compression			Re-compression			Re-compression			Re-compression		
	QP = 30	QP = 32	QP = 34	QP = 30	QP = 32	QP = 34	QP = 30	QP = 32	QP = 34	QP = 30	QP = 32	QP = 34
BlowingBubbles	0.8194	0.8805	0.7786	0.5124	0.814	0.4214	0.4695	0.7487	0.3733	0.5267	0.8281	0.4287
BasketBallDrill	0.8018	0.8705	0.7622	0.5608	0.89	0.4612	0.5179	0.8247	0.4131	0.5747	0.8806	0.4716
RaceHorses	0.7975	0.8606	0.6503	0.4353	0.8857	0.3979	0.3924	0.8204	0.3498	0.4379	0.8928	0.4061
KristenAndSara	0.7534	0.8738	0.7255	0.4018	0.8825	0.391	0.3589	0.8172	0.3429	0.4076	0.8982	0.4172
FourPeople	0.7864	0.8761	0.7172	0.4279	0.8274	0.5334	0.385	0.7621	0.4853	0.4328	0.8395	0.5385
HoneyBee	0.8524	0.8763	0.7519	0.4379	0.539	0.4261	0.395	0.4737	0.378	0.4415	0.5441	0.4276
YachtRide	0.8231	0.8875	0.8134	0.5033	0.8324	0.4896	0.4604	0.7671	0.4415	0.5088	0.8247	0.4967
Jockey	0.7962	0.8857	0.7323	0.4986	0.8122	0.4644	0.4557	0.7469	0.4163	0.5021	0.8147	0.4689
ReadySteadyGo	0.7928	0.8829	0.7823	0.469	0.8561	0.4342	0.4261	0.7908	0.3861	0.4642	0.8621	0.4676
PeopleOnStreet	0.8031	0.8813	0.7244	0.4471	0.882	0.4521	0.4042	0.8167	0.404	0.4662	0.8918	0.4636
Average	0.8026	0.8775	0.7438	0.4694	0.8221	0.4471	0.4265	0.7568	0.399	0.4763	0.8277	0.4587

Table 6 Correlation comparison of proposed scheme with different schemes in terms of frame dropping and noise attacks

Video sequence	Proposed scheme			Chang et al. [1]			Dutta et al. [3]			Liu et al. [10]		
	Gaussian filter	Gaussian noise	Frame dropping	Gaussian filter	Gaussian noise	Frame dropping	Gaussian filter	Gaussian noise	Frame dropping	Gaussian filter	Gaussian noise	Frame dropping
BlowingBubbles	0.6543	0.7044	0.7229	0.6247	0.6748	0.3542	0.4434	0.4854	0.3197	0.6383	0.6607	0.3615
BasketBallDrill	0.6153	0.7024	0.6892	0.5994	0.6691	0.3011	0.4246	0.4644	0.257	0.6133	0.6797	0.2907
RaceHorses	0.7044	0.7132	0.7736	0.6609	0.6807	0.3633	0.4464	0.5762	0.3441	0.6635	0.6978	0.3814
KristenAndSara	0.7005	0.7108	0.7069	0.6386	0.672	0.3619	0.4497	0.4798	0.3288	0.6438	0.6563	0.3981
FourPeople	0.7618	0.7174	0.7077	0.7426	0.6923	0.3563	0.4788	0.4484	0.3222	0.7475	0.6802	0.3512
HoneyBee	0.8245	0.7171	0.7526	0.7846	0.6848	0.3414	0.5885	0.5581	0.3334	0.8082	0.6998	0.3429
YachtRide	0.7141	0.6943	0.7217	0.6957	0.6693	0.2973	0.4771	0.4083	0.2396	0.7112	0.6816	0.2809
Jockey	0.8002	0.7119	0.7692	0.7296	0.6792	0.3138	0.5362	0.5019	0.2896	0.7431	0.6817	0.3093
ReadySteadyGo	0.7581	0.7189	0.6545	0.7445	0.6905	0.3017	0.5381	0.4359	0.2432	0.7695	0.6845	0.2683
PeopleOnStreet	0.7126	0.6941	0.7393	0.6966	0.6535	0.3158	0.5036	0.4841	0.2954	0.7157	0.6733	0.3043
Average	0.7246	0.7085	0.7238	0.6917	0.6766	0.3307	0.4886	0.4843	0.2973	0.7054	0.6796	0.3289

The robustness against re-compression attacks is performed by re-encoding watermarked video sequence with different Qp values of 30, 32 and 34 respectively. The results of the re-compression attack of the proposed scheme and the schemes in [1, 3] and [10] are shown in Table 5. The proposed scheme is also evaluated in terms of various video processing attacks i.e. frame dropping, Gaussian noise with mean = 0 and variance = 0.001, Gaussian filtering with radius of 3 × 3 and sigma = 0.3. The frame dropping attack is performed by dropping 25% random frames similar to the scheme in [9]. In [9] 15 bit binary sequence, i.e. frame number, is inserted before the actual watermark. At the decoder end, first the binary sequence is extracted to identify the frame number and then the embedded watermark is extracted. Table 6 shows the results of the proposed scheme and the schemes in [1, 3] and [10] in terms of different video processing attacks. The results in Tables 5 and 6 show that the robustness of the proposed scheme is better than the schemes in [1, 3] and [10]. The robustness of the proposed scheme obtained by selecting blocks for embedding watermark bits on the basis of the robustness threshold beta calculated at runtime using CCDF based on the spatial analysis of each video sequence explained in Section 2.1. The schemes in [1] and [10] done not follow any criteria to select the blocks for watermark embedding and the scheme in [3] is based on the QDCT coefficients but in HEVC DST is used for 4 × 4 intra luma blocks. The DCT coefficients based scheme in [3] in not efficient for DST based coefficients present 4 × 4 intra luma blocks of HEVC.

The re-compression with different Qp values, noise attacks and frame dropping does not change the contents of the frame i.e. the objects in the frame remains the same. To examine the effectiveness of the proposed scheme in differentiating between malicious tampering and incidental attacks, malicious tampering of watermarked frame is carried out. For this, an object is deliberately removed from the watermarked I-frame of Race Horses



(a) Original watermarked frame

(b) Tampered frame

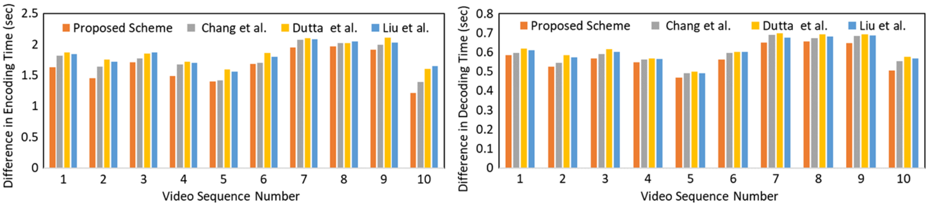
1	1	1	0	0	1	0	0	0	0	1	1	1
1	1	1	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0

(c) Extracted AIM of Tampered Frame (TP=0.201)

1	1	1	0	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0

(d) Processed AIM of Tampered Frame (TP=0.115)(STP=0.571)

Fig. 10 Performance for malicious tamper detection for Race Horses sequence



(a) Watermark embedding (b) Watermark extraction and verification

Fig. 11 Results of average time overhead for watermark embedding, watermark extraction and verification

video sequence. Figure 10a shows the watermarked I-frame and Fig. 10b shows deliberately tampered watermarked I-frame of Race Horses video sequence. In Fig. 10b tampering is done by eliminating the horse face from top left corner of the frame. The Fig. Fig. 10c shows AIM of the watermarked Race Horse frame with a TP value of 0.201. The Fig. 10d shows tampered area of 3 × 3 gliding window in which number of tampered elements are more than 5 which is localized by processing AIM with the 3 × 3 c median filter. Also in the processed AIM the value of STP is 0.571 which is more than 0.553 i.e. the value of T_m calculated using (6) and (7). Hence, this proves that the frame is maliciously tampered.

The computational complexity of the proposed scheme is very low because the algorithms used for authentication code generation, candidate block selection, the embedding of generated authentication code, extraction and verification utilize simple arithmetic operations. The implementation of the proposed scheme is performed by modifying the HM reference software. Therefore, the computational complexity of the proposed scheme is evaluated by calculating processing time overhead of modified HM reference software with respect to the original HM reference software. Figure 11 shows the average processing time overheads in the modified HM reference software for both encoder and decoder. In [1] and [10] first the intra prediction modes of the neighbouring blocks are extracted to select the transformed coefficients for watermarked embedding to prevent intra drift error propagation, but in the proposed scheme all the coefficients used by neighbouring blocks are directly excluded without extracting intra prediction modes. In [3] extra time overhead is due to calculation of pseudo motion vector to prevent error drift. Hence, it can be observed from above analysis and Fig. 11 that the proposed scheme has low computational complexity as compared to the scheme in [1, 3] and [10]. The cost overhead of the proposed scheme is no more than 2 Sec. delay in the encoder and is close to 0 Sec. in decoder. So, it can be concluded that the proposed scheme is efficient in terms of computation complexity.

4 Conclusion

In this paper, a semi-fragile authentication scheme for HEVC encoded video sequences is proposed. The scheme is based on DST coefficients of 4 × 4 intra luma block. The magnitudes of these coefficients are utilized for spatial analysis to achieve desired robustness against various attacks. The other important feature of the proposed scheme is that it is fragile to frame tempering i.e. it can detect object removal and insertion. The imperceptibility and minimal increase in bit-rate are achieved by preserving the sign of transform coefficients during watermark embedding. To evaluate the performance of the proposed scheme, experiments are carried out on various video sequences with different textures and resolutions. The experimental results show that the proposed scheme is more efficient than state-of-art schemes in terms of robustness against re-compression, frame dropping, noise attacks, imperceptible degradation in video quality, computational complexity and minimal increase in bit rate. The proposed scheme can be extended for depth videos by exploiting the new features of the 3D HEVC standard.

Acknowledgements This work is supported by Maulana Azad National Fellowship (MANF) under UGC with application number MANF-2014-15-SIK-PUN-32682.

References

1. Chang PC, Chung KL, Chen JJ, Lin CH, Lin TJ (2014) A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames. J Vis Commun Image Represent 25(2):239–253
2. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2008) Digital watermarking and steganography, 2nd edn. The Morgan Kaufmann Series in Multimedia Information and Systems. Elsevier, San Francisco
3. Dutta T, Gupta HP (2016) A robust watermarking framework for High Efficiency Video Coding (HEVC) –encoded video with blind extraction process. J Vis Commun Image Represent 38:29–44
4. Elrowayati AA, Abdullah M, Manaf AA, Alfagi AS (2016) Robust HEVC video watermarking scheme based on repetition-BCH syndrome code. Int J Softw Eng Appl 10(1):263–270
5. Farfoura ME, Horng SJ, Guo JM, Al–Haj A (2015) Low complexity semi-fragile watermarking scheme for H.264/AVC authentication. Multimed Tools Appl 75(13):7465–7493
6. Furht B, Marques O (2003) Handbook of video databases: design and applications, 1st edn. CRC Press, Boca Raton
7. Hevc reference software [online], available at: https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags
8. Kim DW, Choi YG, Kim HS, Yoo JS, Choi HJ, Seo YH (2010) The problems in digital watermarking into intra-frames of H.264/AVC. Image Vis Comput 28(8):1220–1228
9. Li L, Dong Z, Lu J, Dai J, Huang Q, Chang CC, Wu T (2015) An H.264/AVC hdtv watermarking algorithm robust to camcorder recording. J Vis Commun Image Represent 26:1–8
10. Liu Y, Liu S, Zhao H, Liu S (2018) A new data hiding method for H.265/HEVC video streams without intra-frame distortion drift. Multimed Tools Appl 1–28. https://doi.org/10.1007/s11042-018-6320-y
11. Mansouri A, Aznaveh AM, Torkamani–Azar F, Kurugollu F (2010) A low complexity video watermarking in H.264 compressed domain. IEEE Trans Inf Forensics Secur 5(4):649–657
12. Salomon D (2007) Data compression, 4th edn. Springer, London
13. Saxena A, Fernandes FC (2013) DCT/DST-based transform coding for intra prediction in image/video coding. IEEE Trans Image Process 22(10):3974–3981
14. Shanableh T (2018) Altering split decisions of coding units for message embedding in HEVC. Multimed Tools Appl 77(7):8939–8953
15. Sheikh HR, Bovik AC (2006) Image information and visual quality. IEEE Trans Image Process 15(2):430–444
16. Sullivan GJ, Ohm JR, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circ Syst Video Technol 22(12):1649–1668
17. Swati S, Hayat K, Shahid Z (2014) A watermarking scheme for High Efficiency Video Coding (HEVC). PLoS ONE 9(8):e105–613
18. Sze V, Budagavi M, Sullivan GJ (2014) High efficiency video coding (HEVC), 1st edn. Springer International Publishing, Switzerland
19. Tew Y, Wong K, Phan RCW, Ngan KN (2016) Multi-layer authentication scheme for HEVC video based on embedded statistics. J Vis Commun Image Represent 40:502–515
20. Tew Y, Wong K, Phan RCW, Ngan KN (2018) Separable authentication in encrypted hevc video. Multimed Tools Appl 1–20. https://doi.org/10.1007/s11042–018–5611–7

21. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
22. Wang JJ, Wang RD, Xu DW, Li W (2015) An information hiding algorithm for HEVC based on angle differences of intra prediction mode. JSW 10(2):213–221
23. Wiegand T, Sullivan G, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circ Syst Video Technol 13(7):560–576
24. Wien M (2015) High efficiency video coding, 1st edn. Springer, Berlin
25. Xu D, Wang R, Wang J (2011) A novel watermarking scheme for H.264/AVC video authentication. Signal Process Image Commun 26(6):267–279

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Gagandeep Kaur received the M.C.A degree from Thapar University, Patiala, India, in 2013. She is currently pursuing the Ph.D. degree from Thapar Institute of Engineering and Technology, Patiala, India. Her research interests include video processing and watermarking.

Singara Singh Kasana is working as Associate Professor in Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India. He has eighteen years of teaching and research experience. He received his PhD degree in image compression from Thapar University. His research interests include image processing, machine learning and information security. He has published more than thirty research papers in reputed International Journals and Conferences.

M. K. Sharma is working as Associate Professor in School of Mathematics, Thapar Institute of Engineering and Technology, Patiala, India. He has twenty years of teaching and research experience. He received his PhD degree in Astrophysics from IIT, Roorkee. His research interests include theoretical astrophysics, operations research and image processing. He has published more than fifty research papers in reputed International Journals and Conferences.