Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1340 – 1354

doi:10.3906/elk-1809-38

Robust compressed domain watermarking algorithm for video protection and authentication in noisy channels

Naveen CHEGGOJU*, **Vishal SATPUTE**
Image Processing and Computer Vision Lab, Department of Electrical and Computer Engineering,
Visvesvaraya National Institute of Technology, Nagpur, Maharastra, India

Received: 06.09.2018 • **Accepted/Published Online:** 06.01.2019 • **Final Version:** 22.03.2019

Abstract: This paper introduces a robust and noise-resilient compressed domain video watermarking technique for data authentication and copyright protection. In recent years, watermarking has emerged as an essential technique to be equipped with data transmission. The main challenge pertaining to transmission is to protect the watermark from noise introduced by the channel. Here, we address this issue by watermark replication and by using the independent pass coding (INPAC) algorithm for compression. A replicated watermark is embedded into the video by the proposed blind video watermarking algorithm and then the watermarked video is compressed by the INPAC algorithm. The compressed video is transmitted through binary symmetric channels and tested under various bit error rates to evaluate the proposed algorithm in noisy environments. The results obtained suggest that the proposed algorithm is capable of handling noise efficiently and the overhead due to key embedding is observed to be only about 3%.

Key words: Watermarking, compressed domain, independent pass coding, noise-resilient, key generation

1. Introduction

The modern digital world is progressing in both positive and negative aspects. Positive aspects of development include progress in technologies, medical sciences, astronomy, etc. On the other hand, misuse of these developments may give rise to cyber-crime, copyright tampering, data misleading, etc., corresponding to the negative aspects. Technologies such as high-speed computer networks, the Internet (World Wide Web), etc. have seen some revolutionary changes in recent years [1–3]. These revolutionary technologies have provided an easy access to any kind of multimedia data transmitted through a channel. These data may include all kinds of information corresponding to the public and private domain. This kind of private data, consisting of images, videos, voice, or any other form, needs an appropriate digital management system to keep the data secure from unauthorized access [2, 4]. One of the most efficient techniques for providing the proprietorship on personal data is to insert a secret watermark within the data [5, 6].

In recent years, extensive research has been carried out in the field of watermarking techniques [7]. The main aim of this research is to provide copyright authenticity and prevent unauthorized third-party claims [8–10]. To achieve this goal, the study focuses on introducing robust and efficient watermarking algorithms that can work effectively even in the presence of noise (Gaussian, speckle, salt and pepper, etc.). However, these schemes may require heavy data handling capability as uncompressed multimedia data are of large size [11, 12]. Some of the watermarking schemes that target this issue are proposed in [13, 14]. To make the algorithm

*Correspondence: naveench@ieee.org

memory efficient, we can choose a compressed domain for watermarking, some of which were proposed by Lihua et al. and Mohammad et al. in [15, 16], respectively.

Watermarking in the compressed domain has some limitations to be taken into consideration, like: extra overhead due to the watermark and noise added by the channel due to transmission. If these limitations are treated in an effective way, compressed domain watermarking can solve both memory and copyright issues. We have taken these issues into consideration in our proposed work. A discrete wavelet transform (DWT)-based encoding mechanism is used to embed the watermark [17, 18] and the noise-resilient compression technique of the independent pass coding (INPAC) algorithm [19] is used for compressing the data.

Here, we propose a noise-resilient blind video watermark embedding algorithm in the compressed domain using decoupled 3D-DWT and INPAC. The rest of the paper is arranged as follows: Section-2 introduces the watermarking algorithm and INPAC algorithm, Section-3 presents the proposed method, experimental results and observations are presented in Section-4, and Section-5 concludes the proposed work.

2. Brief description of watermarking and INPAC

This section describes in brief the watermarking and the compression algorithms used in the proposed work. A watermarking algorithm based on DWT is used for key generation and INPAC is used for compressing the video. The compressed video along with the key are secured using a chaotic scrambler. A detailed explanation of these algorithms is given below.

2.1. Watermarking algorithm

The inputs to the watermarking algorithm are the transformed video and replicated binary watermark. The process of watermark replication is as shown in Figure 1. 3D-Decoupled DWT [20] is applied to the video to get the transformed video, and the binary watermark is replicated four times to enhance the robustness to noise [21]. The flow of key generation and key extraction algorithms can be seen from Figures 2a and 2b, respectively. Here, to extract the watermark at the receiver side, we need only the key generated at the time of embedding. In the proposed work, this key is inserted into the bitstream generated by INPAC and sent with the compressed bits.

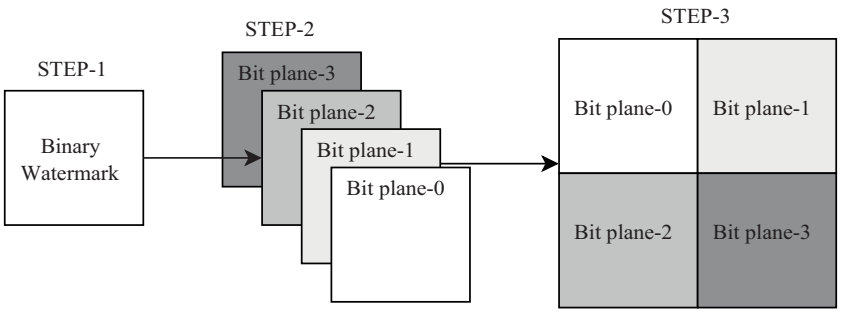
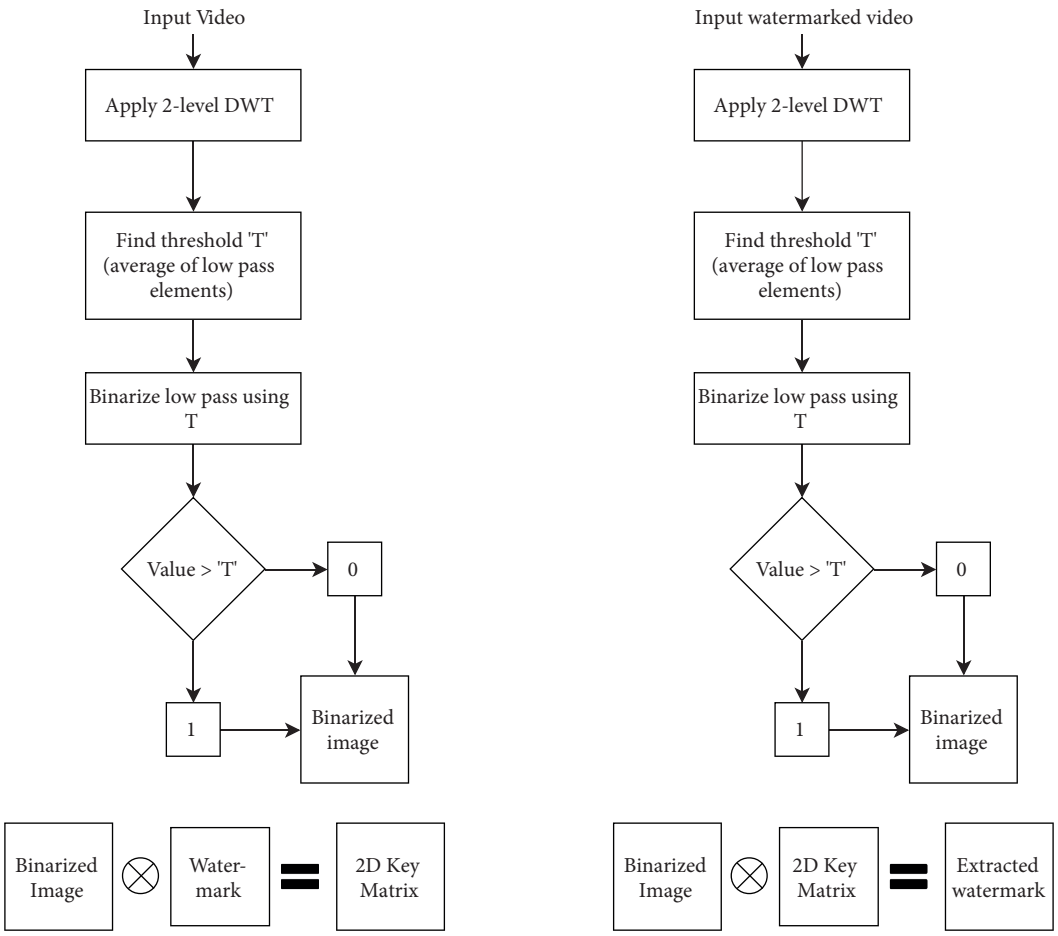


Figure 1. Watermark replication.

2.2. Independent pass coding (INPAC) algorithm

The INPAC algorithm is used for reliable transmission of the compressed bit stream through noisy channels [19]. It is a modified version of the embedded zero wavelet (EZW) algorithm that is basically applied to



(a) Key generation (b) Key extraction

Figure 2. Watermark key generation and extraction.

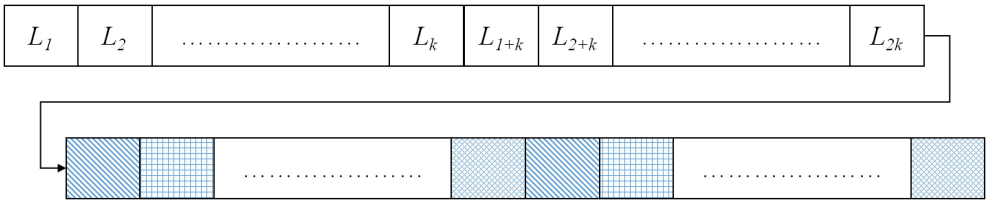


Figure 3. Data arrangement in INPAC.

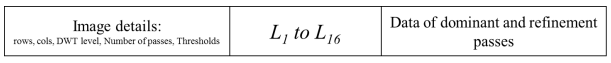


Figure 4. Header of INPAC.

DWT transformed images. Here, we have used the 3D-decoupled DWT transform discussed in [22]. The data arrangement and the header generation of INPAC for each frame of video is as shown in Figures 3 and 4. In these figures, 'k' represents the number of passes, L_1 to L_k represent lengths of data used to represent the

position of significant coefficients in each pass, and L_{1+k} to L_{2k} represent lengths of data used to represent the magnitude of significant coefficients in each pass. This kind of arrangement completely removes the dependency of the receiver algorithm on the previous acquired data. Hence, the main advantage of this arrangement is eliminating the highly noise-affected bits from the string. This method is proven successful against JPEG-2000 in single quality layer scenario [19], which grabbed our interest in choosing this algorithm for compression. The proposed algorithm is intended to reduce the effect of noise in the compressed domain using DWT. As all these requirements can be fulfilled by INPAC, it has been adopted in the proposed work. A detailed explanation regarding the reduction of noise effect is given in Figure 5. In Figure 5, the video string of different frames having different numbers of passes (indicated inside the box) for each frame is depicted. Here, one can observe the noise added to the strings of each pass in different frames. If this noise is left unnoticed, the probability of this noise entering the next clean string of a frame is very high. Hence, to avoid this, each pass of the frame is individually coded so that the noise in one pass will not affect another. This helps in successfully stopping the noise from entering the clean string, but adds some extra number of bits for storage, which act as the control marker for each pass.

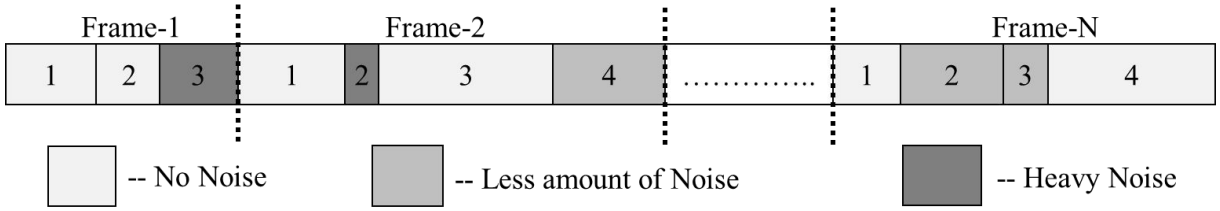


Figure 5. Detailed explanation regarding the reduction of noise effect.

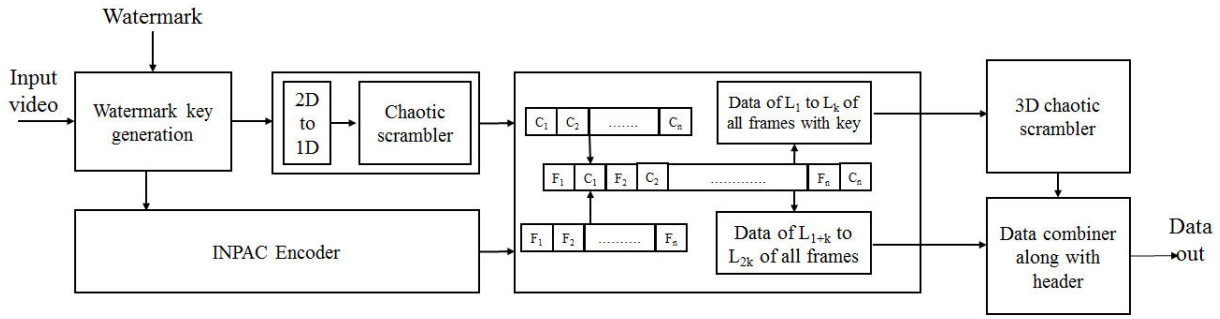


Figure 6. Algorithmic flow of the watermark embedding procedure.

3. Proposed method: INPAC-based video watermarking

The algorithmic flow of the proposed noise-resilient compressed domain video watermarking algorithm is described in Figures 6 and 7. The main goal of this paper is to reduce the effect of channel noise on the embedded watermark and to increase security from third party intrusion. To achieve this, three modules have been used in this work: a watermark key generation module, a noise-resilient compression module (INPAC), and a data scrambling module (chaotic scrambler). These modules are integrated together to achieve a robust and noise-resilient compressed domain watermarking algorithm. The importance of these modules in the proposed algorithm is discussed here briefly.

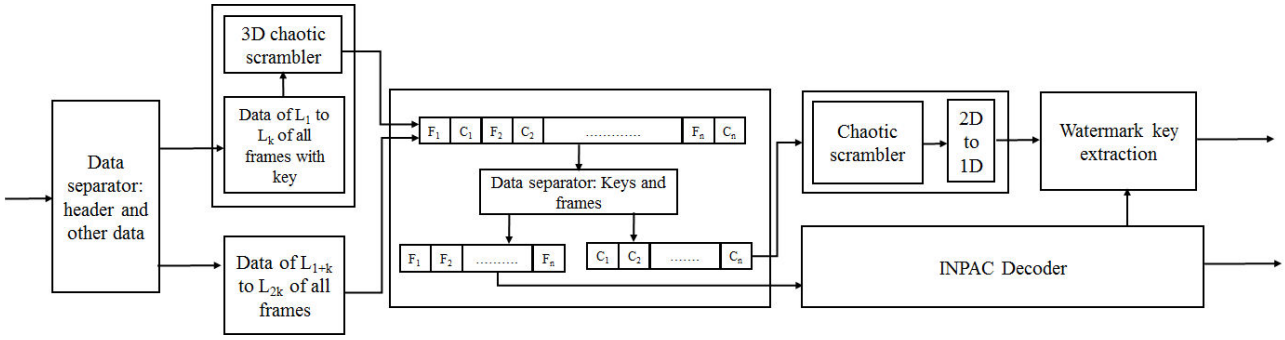


Figure 7. Algorithmic flow of the watermark extraction procedure.

1. Discrete wavelet transform (DWT) is used in the key generation module, as this can be directly given as an input to the INPAC encoder for compression. This reduces the computation time because the same transformed video is used for both key generation and compression. This module consists of the watermark replication process, which plays an important role in making the algorithm noise-resilient.

2. Independent pass coding (INPAC) is used for compression due to its robustness to noise and the capability of accepting the wavelet domain as the input for compression. As MP4 works in a different domain than that of the DWT, INPAC is chosen over it.

3. The chaotic scrambler is used in the algorithm in two stages to provide increased security for the intrusions. A chaotic map is used due to its simplicity and high sensitivity to the change in the inputs.

3.1. Watermark embedding procedure

Figure 6 describes the watermark embedding procedure, where an input video is given to the watermark key generation module along with a binary replicated watermark as the input. The process of binary watermark replication is as shown in Figure 1. The process of replication is done to ensure good reconstruction of the watermark after it is affected by noise. As the watermark extraction follows the maximum rule, it would be safe to store the watermark bits in redundancy. This helps in preserving the watermark with good quality along with the help of the INPAC compression algorithm. This module performs DWT and key generation operations. This transformed video is given as input to the INPAC encoder as shown in Figure 6. The INPAC encoder generates the compressed string for each pass of all the frames and obtained strings are arranged as shown in Figure 3. This ensures that the noise introduced due to the channel is stopped from propagating into the unaffected bits as illustrated in Figure 5. The key generated from the watermark key generation module is given to the chaotic scrambler for key scrambling to ensure security. The scrambled key along with the INPAC compressed string of the dominant pass are combined together to generate a single string, which looks like the compressed string of INPAC. It acts as a mask to the keys generated from the video, which makes it difficult for an intruder to separate the key and the compressed video bits. To ensure further safety, the key along with the dominant pass string (L_1 to L_k) is sent to a chaotic 3D scrambler, in which the string and keys are combined together. The scrambled data along with the refinement string (L_{k+1} to L_{2k}) of INPAC are formed as an INPAC string along with its header as shown in Figure 4. In Figure 6, F_i indicates the frame number and c_i indicates the scrambled key of the ith frame (where i = 1 to number of frames), and L_1 to L_{2k} indicates the

INPAC compressed string of each frame, where k indicates the pass number in each frame. These data along with the header are transmitted through the channel.

3.2. Watermark extraction procedure

Figure 7 describes the watermark extraction procedure of the proposed algorithm. The received data are unpacked from the header and separated into two parts as shown in Figure 7. These two parts comprise data from the dominant pass of INPAC along with the scrambled key and the data from the refinement pass. After descrambling the data from the dominant pass, keys for each frame are extracted by using the length markers L_1 to L_{2k} of INPAC. The separated dominant pass string and the refinement pass string are combined together and sent to the INPAC decoder for video reconstruction as illustrated in Figure 7. Keys extracted are sent to the chaotic scrambler for rearrangement as the original and then converted to a 1-D string for watermark extraction. Reconstructed video through INPAC and the descrambled key are given as inputs to the watermark key extraction module, which extracts the watermark from the reconstructed video. The extracted watermark is the replicated watermark. Using the maximum replication rule, binary data are selected from the replicated watermark to get the watermark of required size. This rule helps in reducing the errors caused to the data by the channel noise. The noise-resilient property of the proposed algorithm is evaluated in Section 4.

4. Experimental results and observations

The proposed noise-resilient blind video watermarking algorithm has been tested under various spatial domain attacks and compressed domain attacks in a binary symmetric channel (BSC). To evaluate the performance, a large database of various videos has been used, selected with different scenes, textures, and scenarios. As the complete database cannot be presented in this paper, results are shown for a sample database of 15 videos with different textures, scenarios, and resolution, which are enlisted in Table 1. Standard MATLAB images "testpat1.png" and "logo.tif" are used as the test watermark images for presenting the results. The watermark is resized according to requirements of the algorithm before embedding. To present the robustness of the proposed algorithm in both the spatial domain and compressed domain, this section has been divided into two subsections for the respective domains. Mathematical parameters of compression ratio (CR), normalized correlation coefficient (NC), and structural similarity (SSIM) index are used to evaluate the performance of the proposed algorithm. The mathematical formulae for CR, SSIM [23], and NC are given by Eqs. (1), (2), and (3), respectively.

$$CR = \frac{size\ of\ compressed\ data}{size\ of\ original\ raw\ image} \tag{1}$$

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

$$NC = \frac{1}{PQ}\sum_{k1=1}^{P}\sum_{k2=1}^{Q} \overline{W(k1,k2) \otimes W'(k1,k2)} \tag{3}$$

Here,

P × Q: size of the image/frame;

x and y: window sizes used for calculating SSIM metric;

Table 1. Details of video database.

Video	Dimension	No. of frames/s	Total no. of frames
Sample 1	120 × 160	15	120
Sample 2	120 × 160	10	92
Sample 3	240 × 320	15	114
Sample 4	240 × 320	10	420
Sample 5	480 × 640	12	662
Sample 6	480 × 640	30	541
Sample 7	480 × 640	10	2424
Sample 8	480 × 640	10	670
Sample 9	480 × 856	29	760
Sample 10	480 × 856	29	334
Sample 11	576 × 704	30	421
Sample 12	576 × 768	25	1452
Sample 13	720 × 1280	15	300
Sample 14	720 × 1280	29	400
Sample 15	720 × 1280	15	102

R: maximum value of the image format (here R = 255);

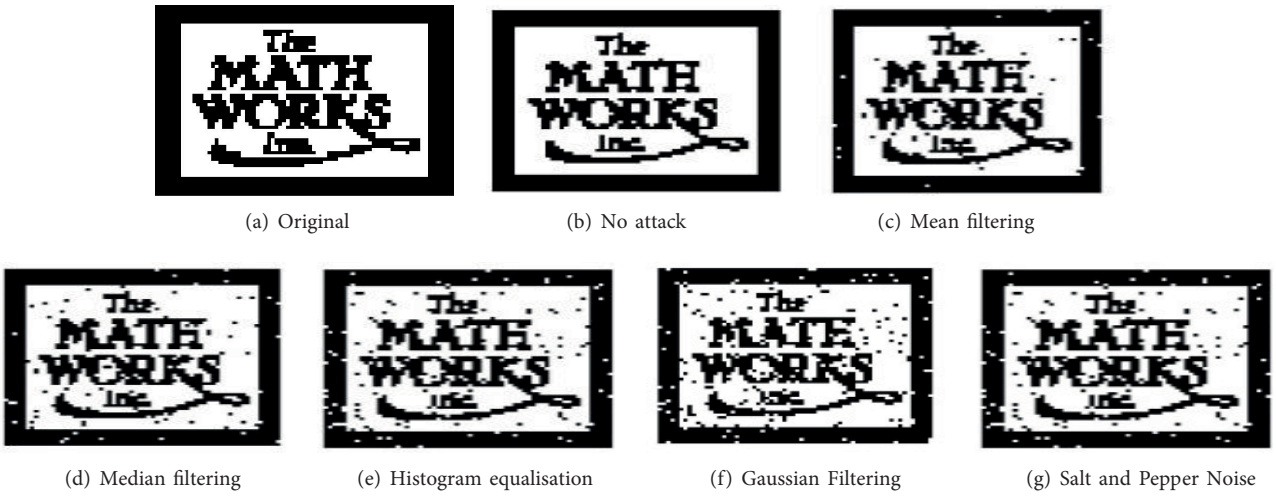
μ_x and μ_y: mean of x and y, respectively;

σ_x^2 and σ_y^2: variance of x and y, respectively; σ_{xy}: covariance of x and y;

$C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$: two variables to stabilize the division with weak denominator;

L: dynamic range of the pixel values (typically L = 255);

W and W': original and reconstructed watermark, respectively; and k_1=0.01, k_2=0.03 [23].



(a) Original (b) No attack (c) Mean filtering

(d) Median filtering (e) Histogram equalisation (f) Gaussian Filtering (g) Salt and Pepper Noise

Figure 8. Reconstructed watermark with various attacks.

Table 2. Spatial domain attacks: NC and SSIM of watermark.

Video	Parameters	Spatial domain attacks				
		Mean filtering	Median filtering	Histogram equalization	Gaussian noise	Salt & pepper noise
Sample 1	NC	0.9930	0.9870	0.9690	0.9560	0.9470
	SSIM	0.9940	0.9890	0.9690	0.9570	0.9480
Sample 2	NC	0.9930	0.9850	0.8960	0.9860	0.9820
	SSIM	0.9930	0.9860	0.9290	0.9870	0.9860
Sample 3	NC	0.9970	0.9970	0.9301	0.9776	0.9750
	SSIM	0.9980	0.9980	0.9258	0.9842	0.9830
Sample 4	NC	0.9980	0.9980	0.9420	0.9890	0.9870
	SSIM	0.9990	0.9990	0.9420	0.9930	0.9920
Sample 5	NC	0.9980	0.9970	0.9530	0.9840	0.9970
	SSIM	0.9980	0.9980	0.9670	0.9880	0.9860
Sample 6	NC	0.9955	0.9947	0.9700	0.9901	0.9878
	SSIM	0.9998	0.9949	0.9812	0.9998	0.9890
Sample 7	NC	0.9986	0.9993	0.9750	0.9930	0.9993
	SSIM	0.9990	0.9994	0.9823	0.9930	0.9994
Sample 8	NC	0.9990	0.9990	0.9420	0.9860	0.9830
	SSIM	0.9990	0.9990	0.9610	0.9900	0.9880
Sample 9	NC	0.9987	0.9991	0.7839	0.9955	0.9960
	SSIM	0.9991	0.9995	0.8437	0.9969	0.9969
Sample 10	NC	0.9989	0.9940	0.9332	0.9959	0.9941
	SSIM	0.9994	0.9996	0.9527	0.9933	0.9962
Sample 11	NC	0.9970	0.9970	0.9480	0.9920	0.9910
	SSIM	0.9980	0.9980	0.9630	0.9950	0.9950
Sample 12	NC	0.9960	0.9971	0.9819	0.9699	0.9652
	SSIM	0.9972	0.9980	0.9918	0.9784	0.9750
Sample 13	NC	0.9929	0.9957	0.8246	0.9333	0.9320
	SSIM	0.9946	0.9967	0.8689	0.9493	0.9483
Sample 14	NC	0.9981	0.9982	0.8458	0.9351	0.9245
	SSIM	0.9921	0.9988	0.9254	0.9492	0.9425
Sample 15	NC	0.9981	0.9983	0.7232	0.9890	0.9878
	SSIM	0.9985	0.9986	0.8398	0.9917	0.9908

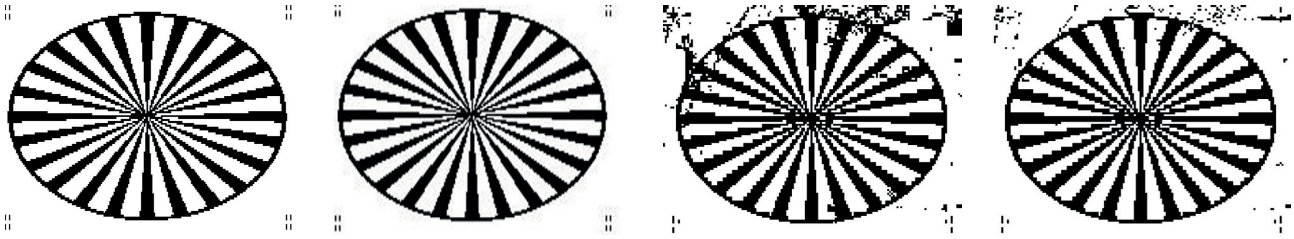
4.1. Results of spatial domain attacks

The watermarking algorithm has been tested with various attacks, mean filtering, median filtering, histogram equalization, Gaussian filtering, and salt and pepper noise, in the spatial domain. Results obtained for these attacks are tabulated in Table 2 and the extracted watermarks for sample 1 are presented in Figures 8a to 8g. Values presented in Table 2 are approaching '1', which shows that the watermarking algorithm is robust against the spatial domain attacks. The extracted watermarks presented in Figures 8c–8g clearly show that they are

Table 3. Video parameters: extra overhead calculations.

Video	Compression ratio		Extra overhead (%)
	Without watermark	With watermark	
Sample 1	0.2455	0.2143	3.12
Sample 2	0.1504	0.1192	3.12
Sample 3	0.3274	0.2962	3.12
Sample 4	0.1199	0.0887	3.12
Sample 5	0.2022	0.1710	2.12
Sample 6	0.0709	0.0397	3.12
Sample 7	0.2826	0.2356	4.70
Sample 8	0.1401	0.1083	3.18
Sample 9	0.1279	0.0974	3.05
Sample 10	0.1330	0.1018	3.12
Sample 11	0.0836	0.0186	6.50
Sample 12	0.0653	0.0340	3.13
Sample 13	0.0647	0.0234	4.13
Sample 14	0.1972	0.1665	3.07
Sample 15	0.1979	0.1667	3.12

perceptually good for providing authentication. From Table 2, it is clear that for mean and median filtering attacks, the values of NC and SSIM obtained are clearly approaching '1' for all the videos. For the histogram equalization attack, NC and SSIM values of samples 9, 13, 14, and 15 are less when compared to other videos; this is because of the high dynamic range in the intensity levels of the video frame. Due to this high dynamic range, histogram equalization has introduced more error into the video frame, which is bringing down the values of NC and SSIM. From the visual representation given in Figure 8, it can be clearly observed that the error introduced in the histogram equalization output is more when compared to mean and median filtering outputs.



(a) Original (b) No noise (c) BER = 10^{-3} (d) BER = 10^{-4}

Figure 9. (a) Original image; (b), (c), (d) reconstructed watermark.

4.2. Compressed domain attacks

To present the efficacy of INPAC with the watermarking algorithm, compression ratio and extra overhead due to watermark are calculated and presented in Table 3, which shows that the extra overhead on average is only approximately equal to 3.5%. Extra overhead is the amount of extra information bits stored in the compressed

Table 4. Compressed domain attacks: NC and SSIM of watermark.

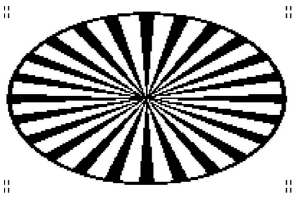
Video	Parameters	Compression attack		
		No noise	$BER = 10^{-3}$	$BER = 10^{-4}$
Sample 1	NC	0.9157	0.9154	0.9137
	SSIM	0.9956	0.9956	0.9956
Sample 2	NC	0.9426 -	0.9655	0.9258
	SSIM	0.9958	0.9978	0.9941
Sample 3	NC	0.9361	0.9318	0.9350
	SSIM	0.9972	0.9970	0.9972
Sample 4	NC	0.9940	0.9878	0.9987
	SSIM	0.9987	0.8279	0.9649
Sample 5	NC	0.9876	0.9974	0.9975
	SSIM	0.9876	0.8940	0.9377
Sample 6	NC	0.9655	0.9938	0.9970
	SSIM	0.9987	0.8606	0.9375
Sample 7	NC	0.9489	0.8980	0.9342
	SSIM	0.9980	0.9950	0.9973
Sample 8	NC	0.8711	0.9976	0.9984
	SSIM	0.9910	0.9367	0.9560
Sample 9	NC	0.9479	0.9980	0.9988
	SSIM	0.9979	0.9545	0.9678
Sample 10	NC	0.9692	0.8650	0.9542
	SSIM	0.9989	0.9918	0.9981
Sample 11	NC	0.9606	0.8872	0.9221
	SSIM	0.9987	0.9941	0.9965
Sample 12	NC	0.9647	0.8562	0.9582
	SSIM	0.9989	0.9925	0.9986
Sample 13	NC	0.9684	0.6315	0.9126
	SSIM	0.9990	0.9730	0.9964
Sample 14	NC	0.9485	0.8999	0.9158
	SSIM	0.9856	0.9968	0.9985
Sample 15	NC	0.9688	0.9199	0.9537
	SSIM	0.9989	0.9953	0.9981

string of the video due to the addition of the watermark. This makes INPAC suitable for compressed domain watermarking. The watermarked string obtained from the proposed algorithm is then passed through a BSC and tested for various amounts of noise attacks as presented in Table 4. The "no noise" attack refers to the compression attack with no additional channel noise introduced into the string. "BER of 10^{-3} (1 in 1000) and 10^{-4} (1 in 10000)" refers to the number of bit changes made to the obtained watermarked string. It can be observed from Table 4 that the SSIM and NC values of the reconstructed watermark are not affected much, which indicates the robustness of the proposed algorithm in noisy conditions. The extracted watermark images for "testpat1.png" are presented in Figures 9a–9d and those for "logo.tif" are presented in Figures 10a–10d.

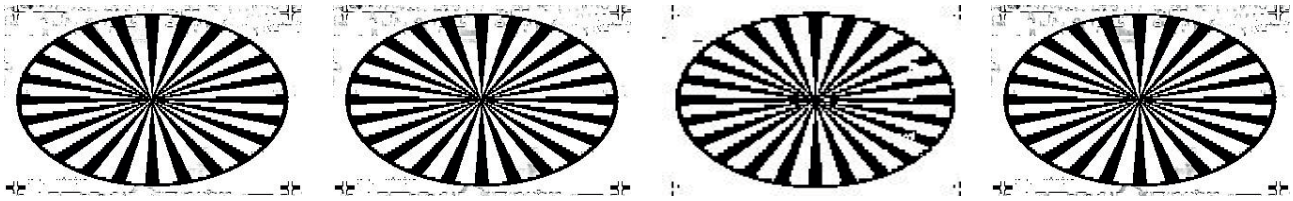
(a) Original (b) No noise (c) BER = 10^{-3} (d) BER = 10^{-4}

Figure 10. (a) Original image; (b), (c), (d) reconstructed watermark.

These figures clearly show that the extracted watermark is perceptually excellent for providing authentication even at higher noise levels. From Table 4, it is clear that in no-noise conditions, the values of NC and SSIM are approaching '1', whereas for the erroneous scenario they are ≈ 0.9, which is still very good for authentication. Results of some videos like sample 13 are not as anticipated because of the background noise present in the video. Hence, the combination of the watermarking algorithm with INPAC is most suitable for compressed domain watermarking for providing efficient authentication.



(a) Original



(b) Noattack (c) Frameaveraging (d) Framedropping (e) Frameswapping

Figure 11. Reconstructed watermark with various temporal attacks.

4.3. Temporal domain attacks

Robustness of the proposed algorithm to temporal attacks has been tested and is presented in this section. From Table 5, it is clearly evident that the value of NC and SSIM for all types of temporal attacks is approaching '1'. For frame dropping, the dropped frames are replaced with a blank frame, and the max rule is followed at the time of reconstruction. Frame swapping attack is the combination of frame swapping and frame duplication, but still the proposed algorithm is robust to handle the attack. For the averaging attack, 10 consecutive frames are averaged before extracting the watermark, and in the case of frame dropping, 20% of the total frames are dropped from the video before extraction of the watermark. To present the visual quality, averages of reconstructed watermarks for different attacks are presented in Figures 11a–11e. It can be clearly seen that the visual quality of the watermark is satisfactory for providing authentication.

Table 5. Temporal domain attacks: NC and SSIM of watermark.

Video	Parameters	Frame averaging	Frame dropping	Frame swapping
Sample 1	NC	0.9339	0.9281	0.9320
	SSIM	0.9971	0.9969	0.9969
Sample 2	NC	0.9068	0.8390	0.8788
	SSIM	0.9949	0.9860	0.9926
Sample 4	NC	0.9663	0.9502	0.9845
	SSIM	0.9988	0.9970	0.9969
Sample 5	NC	0.9641	0.9422	0.9560
	SSIM	0.9988	0.9969	0.9982
Sample 6	NC	0.9669	0.9534	0.9643
	SSIM	0.9990	0.9971	0.9989
Sample 9	NC	0.9664	0.9653	0.9650
	SSIM	0.9987	0.9856	0.9987
Sample 10	NC	0.9650	0.9623	0.9650
	SSIM	0.9988	0.9986	0.9988
Sample 11	NC	0.9263	0.9564	0.9623
	SSIM	0.9987	0.9983	0.9987
Sample 12	NC	0.9606	0.9329	0.9605
	SSIM	0.9987	0.9964	0.9987
Sample 13	NC	0.9476	0.9451	0.9476
	SSIM	0.9979	0.9978	0.9979
Average	NC	0.9057	0.9375	0.9516
	SSIM	0.9982	0.9964	0.9976

4.4. Comparisons

The proposed work has been compared with the existing literature in two categories: spatial domain attacks and temporal domain attacks. In the spatial domain, attack comparisons have been done by considering the parameters of Gaussian noise attack at $\mu = 0$ and $\sigma = 0.1, 0.3, 0.5$; salt and pepper noise attack at $d = 0.01, 0.02, 0.04, 0.08$; blurring attack at $\sigma = 0.1, 0.2, 0.3$; brightening attack by 50 and 80; and darkening attack by 50 and 80. The comparative results with [24–26] are presented in Table 6. From the table, one can observe that the proposed algorithm has a negligible change in NC values under various attack conditions. The other algorithms have considerable changes in the values of NC under various attack conditions, which shows the robustness of the proposed algorithm. The values obtained using the existing algorithms consider only the above mentioned attacks, whereas the values presented for the proposed algorithm are obtained after both compression attack using INPAC and the above mentioned attacks as well. Therefore, the NC values obtained represent the values obtained after two attacks applied simultaneously.

For temporal attack comparisons, frame averaging and frame dropping are compared with [27]. Average NC values for frame averaging (10 frames) and frame dropping (at a rate of 20%) of [27] are 0.8060 and 0.9376, respectively. The average values for the same parameters of the proposed algorithm from Table 5 are 0.9507 and 0.9375, respectively. This shows that the proposed algorithm has good robustness against both frame

Table 6. Comparison of the spatial domain attacks.

Attack	Parameter	Lusson et al. [24]	Su et al. [25]	Liu et al. [26]	Proposed
Gaussian	0.1	0.9664	0.9767	0.9664	0.9651
	0.3	0.9334	0.9003	0.9171	0.9650
	0.5	0.9064	0.8679	0.8735	0.9558
Salt and pepper	0.01	0.9986	0.9952	0.999	0.9651
	0.02	0.9946	0.9889	0.9981	0.9651
	0.04	0.9913	0.9847	0.9959	0.9651
	0.08	0.9898	0.96	0.9893	0.9650
Blurring	0.1	0.9998	1	0.9997	0.9652
	0.2	0.4436	0.9877	0.9514	0.9652
	0.3	0.3279	0.9393	0.8735	0.9652
Brighten	50	0	0.9208	0.9904	0.9651
	80	0	0.804	0.9806	0.9651
Darken	50	0	0.9999	0.9655	0.8968
	80	0	0.9921	0.8813	0.7769

Table 7. Comparison of the temporal domain attacks.

	Frame averaging (no. of averaged frames)				Frame dropping (%)		
	2	**5**	**7**	**10**	**5**	**10**	**20**
[27]	0.996	0.956	0.885	0.810	0.997	0.976	0.962
Proposed	0.968	0.968	0.968	0.968	0.963	0.963	0.962

averaging and frame dropping when compared to the existing algorithms. Comparative values for a single video are presented in Table 7. To compare the robustness, the best values from [27] are taken against the proposed algorithm.

5. Conclusion

In this paper, a noise-resilient blind video watermarking method is proposed that uses the INPAC algorithm and DWT. DWT is used to transform the video and INPAC is used for noise-resilient compression. A new concept of key replication is also introduced to increase the noise resiliency. The key matrix generated from the proposed watermarking algorithm is embedded in the compressed domain and transmitted along with the video frames, which acts as a mask to the original key. This helps in hiding the key from intruders and makes it difficult to differentiate the video data and the key. For additional security, chaotic scrambling is applied in two stages, one on the key matrix and the other on the final output of the encoder block. The final string is tested in noisy conditions of the BSC and the results show the error handling capability of the proposed algorithm. The key string transmitted along with the INPAC compressed data contributes only approximately equal to 3.5% of the extra data. Also, as the algorithm is in the compressed domain, it proves to be both memory-efficient and computationally effective. Comparison with the existing algorithms shows that the proposed method has maintained its robustness at all levels of attacks.

References

[1] Bhattacharya S, Chattopadhyay T, Pal A. A survey on different video watermarking techniques and comparative analysis with reference to h.264/avc. In: IEEE 2006 Tenth International Symposium on Consumer Electronics; St. Petersburg, Russia; 2006. pp. 1-6.

[2] Chang X, Wang W, Zhao J, Zhang L. A survey of digital video watermarking. In: IEEE 2011 Seventh International Conference on Natural Computation; Shanghai, China; 2011. pp. 61-65.

[3] Goel B, Agarwal C. An optimized un-compressed video watermarking scheme based on SVD and DWT. In: IEEE 2013 Sixth International Conference on Contemporary Computing; Noida, India; 2013. pp. 307-312.

[4] Raghavendra K, Chetan K. A blind and robust watermarking scheme with scrambled watermark for video authentication. In: IEEE 2009 International Conference on Internet Multimedia Services Architecture and Applications; Bangalore, India; 2009. pp. 1-6.

[5] Petrovic R, Yang DT. Audio watermarking in compressed domain. In: 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services; Nis, Serbia; 2009. pp. 395-401.

[6] Mansouri A, Aznaveh AM, Torkamani-Azar F, Kurugollu F. A low complexity video watermarking in h. 264 compressed domain. IEEE Transactions on Information Forensics and Security 2010; 5 (4): 649–657. doi: 10.1109/TIFS.2010.2076280

[7] Asikuzzaman M, Pickering MR. An overview of digital video watermarking. IEEE Transactions on Circuits and Systems for Video Technology 2016; 28 (9): 2131-2153. doi: 10.1109/TCSVT.2017.2712162

[8] Liu S, Chen T, Yao H, Gao W. A real-time video watermarking using adjacent luminance blocks correlation based on compressed domain. In: IEEE 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing; Harbin, China; 2008. pp. 833-836.

[9] Asikuzzaman M, Alam MJ, Lambert AJ, Pickering MR. Imperceptible and robust blind video watermarking using chrominance embedding: a set of approaches in the dt cwt domain. IEEE Transactions on Information Forensics and Security 2014; 9 (9): 1502-1517. doi: DOI: 10.1109/TIFS.2014.2338274

[10] Naveen C, Satpute VR, Kulat KD, Keskar AG. Comparative analysis of video compression mechanisms based on EZW coded 3D-DWT and AVI. In: IEEE 2014 International Symposium on Signal Processing and Information Technology; Noida, India; 2014. pp. 96-101.

[11] Tabassum T, Islam SM. A digital video watermarking technique based on identical frame extraction in 3-level dwt. In: IEEE 2012 15th International Conference on Computer and Information Technology; Chittagong, Bangladesh; 2012. pp. 101-106.

[12] Li LZA. A study on video watermark based-on discrete wavelet transform and genetic algorithm. In: IEEE 2009 First International Workshop on Education Technology and Computer Science; Wuhan, China; 2009. pp. 374-377.

[13] Al-Taweel SA, Sumari P, Kamarulhaili H. Digital video watermarking based on 3d-discrete wavelet transform domain. In: IEEE 2009 International Conference on Signal and Image Processing Applications; Kuala Lumpur, Malaysia; 2009. pp. 352-356.

[14] Tianming G, Yanjie W. DWT-based digital image watermarking algorithm. In: IEEE 2011 10th International Conference on Electronic Measurement & Instruments; Chengdu, China; 2011. pp. 163-166.

[15] Tian L, Zheng N, Xue J, Xu T. A cavlc-based blind watermarking method for h. 264/avc compressed video. In: IEEE 2008 Asia-Pacific Services Computing Conference; Yilan, Taiwan; 2008. pp. 1295-1299.

[16] Nambakhsh MS, Ahmadian A, Ghavami M, Dilmaghani RS, Karimi-Fard S. A novel blind watermarking of ECG signals on medical images using EZW algorithm. In: IEEE 2006 28th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society; New York, NY, USA; 2006. pp. 3274-3277.

[17] Shapiro JM. Embedded image coding using zerotrees of wavelet coefficients. IEEE Transactions on Signal Processing 1993; 41 (12): 3445-3462. doi: 10.1109/78.258085

[18] Said A, Pearlman WA. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on Circuits and Systems for Video Technology 1996; 6 (3): 243–250. doi: 10.1109/76.499834

[19] Cheggoju N, Satpute VR. INPAC: Independent pass coding algorithm for robust image data transmission through low snr channels. Visual Computer 2018; 34 (4): 563-573. doi: https://doi.org/10.1007/s00371-017-1361-1

[20] Naveen C, Satpute V, Kulat K, Keskar A. Video encoding techniques based on 3D-DWT. In: IEEE 2014 Students' Conference on Electrical, Electronics and Computer Science; Bhopal, India; 2014. pp. 1-6.

[21] Satpute V, Kadu S, Naveen C. Compressed domain video watermarking using EZW and chaos. In: IEEE 2006 Region 10 Conference; Singapore; 2016. pp. 3083-3086.

[22] He C, Dong J, Zheng YF, Gao Z. Optimal 3-D coefficient tree structure for 3-D wavelet video coding. IEEE Transactions on Circuits and Systems for Video Technology 2003; 13 (10): 961-972. doi: 10.1109/TCSVT.2003.816514

[23] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 2004; 13 (4): 600-612. doi: 10.1109/TIP.2003.819861

[24] Lusson F, Bailey K, Leeney M, Curran K. A novel approach to digital watermarking, exploiting colour spaces. Signal Processing 2013; 93 (5): 1268-1294. doi: https://doi.org/10.1016/j.sigpro.2012.10.018

[25] Su Q, Niu Y, Zou H, Liu X. A blind dual color images watermarking based on singular value decomposition. Applied Mathematics and Computation 2013; 219 (16): 8455-8466. doi: https://doi.org/10.1016/j.amc.2013.03.013

[26] Liu XL, Lin CC, Yuan SM. Blind dual watermarking for color images authentication and copyright protection. IEEE Transactions on Circuits and Systems for Video Technology 2018; 28 (5): 1047-1055. doi: 10.1109/TCSVT.2016.2633878

[27] Karmakar A, Phadikar A, Phadikar BS, Maity GK. A blind video watermarking scheme resistant to rotation and collusion attacks. Journal of King Saud University-Computer and Information Sciences 2016; 28 (2): 199-210. doi: https://doi.org/10.1016/j.jksuci.2014.06.019