Authentication of Surveillance Video Sequences and Contents by Hiding Motion Vector Information*

Kuo-Feng Chien (簡國峯)[1] and *Wen-Hsiang Tsai* (蔡文祥)[1, 2]

[1]Dept. of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

[2]Dept. of Computer Sci. & Information Eng., Asia University, Taichung, Taiwan

ABSTRACT

A method for authentication of MPEG-4 surveillance videos by information hiding techniques is proposed. The method can verify both temporal and spatial tampering of video sequences by utilizing the motion vector information in the video frames. Spatial tampering means modifications manipulated on video frame contents, and temporal tampering means modifications on video frame sequences. Three types of temporal tampering can be handled by the proposed method, including frame replacement, insertion, and cropping. In order to detect spatial tampering, some authentication signals are embedded into the DCT coefficients of each 8×8 luminance block in each I frame of a video sequence. Authentication signals composed of two types of features, the index of the group of pictures (GOP) and the movement information of the inter-coded frames in the GOP, are used to detect the tampering. Good experimental results show the feasibility of the proposed method.

Keywords: Fidelity, integrity, authentication, surveillance video, MPEG-4, DCT coefficient.

1. INTRODUCTION

In this study, a method for authentication of surveillance video sequences and contents is proposed. With the rapid development of the environment surveillance system, the MPEG-4 compression technique is used popularly in many applications. However, digital videos are easy to be modified by many commercial video editing software packages. In addition, recent surveillance systems are usually designed for remote controls through computer networks. In other words, recorded surveillance videos are transmitted on the Internet to central servers and stored, and so they can be acquired and tampered with easily. It is so necessary to authenticate the integrity and fidelity of received video sequences and contents at receiver sites.

Numerous methods have been proposed in recent years for authenticating videos. Two typical approaches are the digital signature technique [1,2] and the digital watermarking technique [3-8]. Chen and Tsai [9] have proposed an authentication method for MPEG videos by hiding some random signals.

In this study, we propose an authentication method for verifying whether a given MPEG-4 surveillance video has been tampered with or not. Malicious operations on the video can be divided into two types: *spatial tampering* and *temporal tampering*. Spatial tampering means modifications manipulated on video frame contents, and temporal tampering means modifications on video frame sequences.

Temporal tampering can be categorized further into three types: *replacement*, *cropping*, and *insertion*. Replacement means to delete some video frames and then add an identical number of fake video frames into the original video. The resulting number of video frames is not changed and the difference of the video size between the original and the fake videos will be too tiny to be detected visually. Cropping means deleting some video frames from the original video sequence. For example, a malicious user might eliminate his criminal evidence by cropping some relevant video frames in the original video. The third type of tampering, insertion, means to add fake video frames into the original video sequence. For example, a malicious user might insert some fake frames to pin his crime on someone else who is innocent.

The main task of the proposed authentication system is not only to detect whether a surveillance video has been tampered with, but also to recognize the tampering type and mark further the regions which have been tampered with. In order to detect spatial tampering, some *authentication signals* are embedded into the DCT coefficients of each 8×8 luminance block in each I frame of a

* This work is supported partially by the NSC Project Advanced Technologies and Applications for Next Generation Information Networks (II) with Project No. NSC93-2752-E-009-006-PAE and partially by the Ministry of Economic Affairs under Project No. MOEA 94-EC-17-A-02-S1-032 in Technology Development Program for Academia.

video sequence. Authentication signals are designed to be composed of two types of features in each group of pictures (*GOP*) of the video. One is the index of the GOP and the other the movement information of the inter-coded frames in the GOP. Furthermore, we also utilize the index of the GOP stored in the authentication signals to detect temporal tampering.

In the remainder of this paper, the proposed method for embedding authentication signals is described in Section 2, and the proposed authentication process for video sequences and contents is described in Section 3. In Section 4, several experimental results of applying the proposed method will be shown. Finally, some discussions and a summary will be made in the last section.

2. EMBEDDING OF AUTHENTICATION SIGNALS IN SURVEILLANCE VIDEOS

In this section, the proposed authentication signal embedding method will be described.

2.1. Embedding of Authentication signals

In this study, we treat each GOP in the video as a unit for the authentication process. Each GOP consists of a leading I frame and six following P frames. The proposed process of authentication signal embedding is divided into two phases. In the first phase, we record the index of the GOP and analyze the motion vectors in each P frame to acquire the movement information of the GOP. According to the degree of movement, the P frames in the GOP can be categorized into two types: motion P frame and still P frame.

In the second phase, we gather the data mentioned in the first phase together to form the authentication signals which are then embedded into each I frame.

2.2. Process of Generating Authentication Signals by Analyzed Motion Vectors in P Frames

Besides the index of the GOP, we also use the movement information of the six P frames in the GOP to generate the authentication signals. For each inter-coded macroblock in each P frame of the i-th GOP, we denote the motion vector of the corresponding macroblock as (mv_x, mv_y). If the value of mv_x or mv_y is greater than T_1, where T_1 is a pre-defined threshold, we calculate the total number $N(f_j)$ of such motion vectors in each P frame f_j. By comparing $N(f_j)$ with a pre-defined threshold, we can judge whether f_j is a motion P frame or a still one. Then we can form a binary string to represent the movement information of the six P frames in the i-th GOP, where bit 0 represents a still P frame and bit 1 represents a motion P frame. The details are described as an algorithm in the following.

```
Algorithm 1: authentication signal generation for
        each GOP of a video sequence.
Input: the $i$-th GOP $G_i$ in a video.
Output: authentication signals $S_b$ to be embedded.
Steps:
  1. For each inter-coded macroblock $m$ in each P
     frame $f_j$ of $G_i$, calculate the total number $N(f_j)$
     of selected motion vectors ($mv_x$, $mv_y$)
     according to the following rule:
        if $mv_x > T_1$ or $mv_y > T_1$,
            then set $N(f_j) = N(f_j) + 1$;
            otherwise, let $N(f_j)$ unchanged;          (1)
     where $T_1$ is a pre-defined threshold.
  2. Calculate the average $G_{avr}$ of the six $N(f_j)$ and
     get a result as follows:
        $G_{avr} = \frac{1}{N_p}\sum_{j=1}^{N_p} N(f_j)$          (2)
     where $N_p$ is the number of P frames in $G_i$.
  3. For each P frame $f_j$ of $G_i$, determine the
     movement of $f_j$ as a binary bit $M(f_j)$ according
     to the following rule:
     (1) when $G_{avr} > T_{avr}$:
           if $N(f_j) > T_2$, then set $M(f_j) = 1$;
           if $N(f_j) < T_2$, then set $M(f_j) = 0$;          (3)
     (2) when $G_{avr} < T_{avr}$:
           set all the $M(f_j) = 0$;                    (4)
     where $T_{avr}$ and $T_2$ are two pre-defined
     thresholds.
  4. Concatenate the values of $M(f_j)$ of all the P
     frames $f_j$ of $G_i$ to form a binary string $S$.
  5. Transform the index of $G_i$ into binary form
     and combine it with $S$ to form a new binary
     string $S_b$ as the desired authentication signals.
```

In the above algorithm, we use the threshold T_{avr} to judge the movement of the i-th GOP. Sometimes a still P frame may probably be analyzed as a motion one because the changes of the light and shadows may cause huge motion vectors in the P frames. Therefore, the main function of the threshold T_{avr} is to reduce erroneous judgments coming from abnormal motion vectors.

2.3. Process of Embedding Authentication Signals in I Frames

After the first phase of the proposed process of authentication signal embedding is completed, we obtain the authentication signals of each GOP of the video. Before beginning the second phase, the authentication signals in binary form are duplicated into several copies, where the total number of bits of these copies is smaller than the total number of 8×8 blocks in an I frame.

The main purpose of this duplication process is to extract authentication signals precisely in the subsequent authentication process in order to reduce the probability of misrepresentation. Furthermore, we utilize four pairs of DCT coefficients selected randomly with a user-specified key from eight pre-defined ones in the middle frequency band to embed the four bits of the authentication signals in an 8×8 luminance block. The details are described as an algorithm in the following.

***Algorithm* 2**: authentication signals embedding for I frames.

Input: an I frame F in the quantized DCT domain, authentication signals S_b in binary form, and a user-specified key K.

Output: a protected I frame F'.

Steps:

1. Denote S_b as $S_b = s_1s_2s_3\ldots s_L$, where L is the length of S_b, and duplicate it k times to form a new binary string S_b'.
2. For each 8×8 luminance block B of F, combine the input key K and the position P of B to form a seed for random number generation.
3. Use the result of random number generation to randomly select four pairs of DCT coefficients ($C_1[i]$, $C_2[i]$) from the eight pre-defined ones to embed four bits b_i of S_b' by changing the relation between $C_1[i]$ and $C_2[i]$ according to the following rule:

   (1) when $b_i = 0$:

   if $C_1[i] < C_2[i]$, swap $C_1[i]$ and $C_2[i]$;
   if $C_1[i] = C_2[i]$, swap $C_1[i]=C_2[i] + T_3$; (5)

   where T_3 is a pre-defined threshold;

   (2) when $b_i = 1$:

   if $C_1[i] > C_2[i]$, swap $C_1[i]$ and $C_2[i]$;
   if $C_1[i] = C_2[i]$, swap $C_2[i]=C_1[i] + T_3$; (6)

   where T_3 is a pre-defined threshold.

In the above algorithm, the threshold T_3 mentioned in Step 3 is a tradeoff between the robustness and the resulting video quality. The higher the threshold T_3 is, the authentication signals embedded in the video are more robust to survive MPEG recompression; however, the resulting video quality suffers more degradation as well.

3. AUTHENTICATION OF VIDEO SEQUENCES AND CONTENTS

In this section, the proposed method for authentication of video sequences and contents is described.

3.1. Idea of Authentication

The process of authentication of video sequences and contents also can be divided into two phases. In the first phase, we use a voting scheme to extract the embedded authentication signals in each I frame. In the mean time, we analyze the motion vectors in the six P frames to acquire the movement information of each GOP in the input video.

In the second phase, we utilize the index of the GOP to verify the sequence order of the video for detecting temporal tampering and recognize the tampering types. Moreover, we compare the extracted movement information with the analyzed one in the first phase to judge whether the GOP has been tampered with or not. If so, the extracted authentication signals will be utilized to verify the spatial integrity of the I frame and further mark the unauthentic macroblocks.

3.2. Process for Extracting Authentication Signals Using Voting

In this study, the method utilized to extract authentication signals is a voting scheme. In the previously-described process of authentication signal embedding, we embed several copies of signals into each I frame. Because sometimes the MPEG recompression causes slight changes to the DCT coefficients, some of the embedded signals may be changed. Therefore, we can extract the authentication signals more precisely by the voting scheme if the area which is not tampered with spatially in the I frame is large enough.

In the proposed voting process for I frames, we give two scores to two possible values, 0 and 1, of each bit in each copy of the extracted authentication signals. The value with the higher score will be regarded as the correct value of the corresponding bit. Therefore, after all copies of the authentication signals are extracted, we can get the correct one according to the voting result.

***Algorithm* 3**: authentication signal extraction.

Input: a protected I frame F' and a user-specified key K.

Output: authentication signals S.

Steps:

1. For each 8×8 luminance block B, combine the input key K and the position P of B to form a seed for random number generation.
2. Use the result of random number generation to randomly select four pairs ($\mathrm{C}_I[k]$, $\mathrm{C}_2[k]$) of DCT coefficients from the eight pre-defined ones and extract four bits b_k of each copy of signals S' according to the following rule:

   if $C_1[k] > C_2[k]$, then set $b_k = 1$;
   if $C_1[k] < C_2[k]$, then set $b_k = 0$; (7)

   where $0 \leq k \leq 3$.
3. For each copy of the extracted authentication signals S', transform it into a binary string $S' = b_1b_2b_3\ldots b_L$ where L is the length of signals. Assign each bit of S' two voting scores $V_0[m]$ and $V_1[m]$, where $1 \leq m \leq L$. Calculate the score of each bit of the authentication signals according to the follow rule:

```
    if $b_m = 0$, then set $V_0[m] = V_0[m] + 1$;
    if $b_m = 1$, then set $V_1[m] = V_1[m] + 1$.     (8)
4. Denote the binary form of the correct authentication signals $S$ as $S = s_1s_2s_3\ldots s_L$. Reconstruct $S$ by comparing the two scores of each bit of $S$ according to the following rule:
    if $V_0[m] > V_1[m]$, then set $s_m = 0$;
    if $V_1[m] > V_0[m]$, then set $s_m = 1$;     (9)
   where $1 \leq m \leq L$.
```

During the process of extracting authentication signals mentioned above, we also analyze each GOP of the input video concurrently and get the movement information from the statistics of the motion vectors in each P frame. The detailed algorithm is similar to the one described in Section 2.2. Moreover, we store the extracted authentication signals and the movement information of statistics into a temporary report R for the subsequent authentication process.

3.3. Detection and Verification of Temporal Tampering for Suspected Videos

According to the output report R received from the process of extracting authentication signals proposed in Section 3.2, we can utilize the extracted index of each GOP, denoted as G_i', to verify the correctness of a video sequence. We denote the current index of the GOP as G_i and compare it with G_i' to detect possible temporal tampering of the input video.

In the proposed method, we classify the temporal tampering into two types: one being replacement, and the other cropping or insertion. We can not only recognize both types of tampering but also detect how many segments of the video have suffered the replacement tampering and approximately report the starting and ending positions for each modified segment.

***Algorithm* 4**: detecting temporal tampering.

Input: a GOP sequence G of a suspected video and a report R with extracted authentication signals.

Output: a report R' of the temporal tampering detection result.

Steps:

```
1. Let flag bit $b$ denote the occurrence of tampering, and initialize it to be 0.
2. Denote $G_i$ as the index of each GOP of $G$ and compare the extracted index $G_i'$ in $R$ with $G_i$.
3. If $G_i \neq G_i'$, then perform the following steps.
   3.1 If $b$ equals 0, then set $b$ to 1 and record the frame index $n_s$ of the I frame in $G_i$.
   3.2 If $b$ equals 1, then do nothing.
4. If $G_i = G_i'$, then perform the following steps:
   4.1 If $b$ equals 1, then record the frame index $n_f$ of the I frame in $G_i$; recognize the tampering type as replacement and store it into $R'$; store $n_s$ and $n_f$ into $R'$ as the starting and ending positions of the replacement tampering; and then set $b$ to 0.
   4.2 If $b$ equals 0, then do nothing.
5. Repeat Steps 1 through 4 for each GOP until reaching the end of the file of $R$.
6. If $b$ equals 1, then recognize the tampering type as cropping or insertion and store it into $R'$; otherwise, do nothing.
```

In the above algorithm, the flag bit b represents whether the video sequences are tampered with or not. If the extracted index G_i' is not equal to the current index G_i, we set b to 1 because the GOP with index G_i has been tampered with. During the authentication process, if the extracted index G_i' equals the current index G_k, where k is greater than i, then an approximate segment of replacement tampering is detected and b is reset to 0 for detecting the next modified segment.

3.4. Detection and Verification of Spatial Tampering for Suspected Videos

Most frames of surveillance videos are still background without moving objects, and so such frames might be used by malicious users to replace frames including contents of criminal behaviors. They even might cut some regions from background frames and paste them onto regions in other frames where criminal activities occur. In the proposed method, if the area of not being tampered with in an I frame is large enough, we can utilize the voting scheme to extract the correct authentication signals. Moreover, the extracted signals can be used to verify the spatial integrity and fidelity of the corresponding I frame.

As mentioned before, we get a report R after the process of extracting authentication signals. Each entry of R contains three types of data, the extracted index G_i', the extracted movement information M_i' and the current movement information M_i, of each GOP. We calculate the number N_d of different bits between M_i' and M_i. If N_d is greater than zero, it represents that the GOP with index G_i' is recognized as a suspected GOP. The details are described as an algorithm in the following.

***Algorithm* 5**: detection of spatial tampering.

Input: a suspected GOP G_s with I frame F', extracted authentication signals S, and a user-specified key K.

Output: an authenticated I frame F''.

Steps:

```
1. For each macroblock $MB$ of $F'$, denote the number of suspected 8×8 blocks in $MB$ as $N_{8\times8}$.
2. For each 8×8 luminance block $B_i$ of $MB$,
```

where $i = 0, 1, 2, 3$, combine the input key K and the position P of B_i to form a seed for random number generation.

3. Use the result of random number generation to randomly select four pairs of DCT coefficients ($C_1[i]$, $C_2[i]$) from the eight pre-defined ones to extract four bits b_i of embedded signals according to the following rule:

$$\text{if } C_1[i] > C_2[\mathrm{i}], \text{ then set } b_i = 0;$$
$$\text{if } C_1[i] < C_2[\mathrm{i}], \text{ then set } b_i = 1; \quad (10)$$

where $i = 0$ through 3.

4. Compare b_i with the corresponding bit s_i in S to determine whether B_i is tampered with or not according to following rules:

$$\text{if } b_i \neq s_i, \text{ then set } N_b = N_b + 1;$$
$$\text{otherwise, set } N_b \text{ unchanged;} \quad (11)$$

where N_b is the number of suspected bits in B_i.

5. If N_b is greater than 2, set B_i as a suspected block and increment $N_{8\times8}$ by 1.

6. After processing each 8×8 luminance block, employ the following two rules to verify each macroblock MB.

(1) Rule 1:

if B_0 and B_3 are suspected or if B_1 and B_2 are suspected, then set MB as unauthentic and mark MB as a region tampered with. (12)

(2) Rule 2:

if MB_{left} and MB_{top} are unauthentic, or
if MB_{left} is unauthentic and $N_{8\times8} > 0$, or
if MB_{left} is unauthentic and $N_{8\times8} > 0$, or
if $MB_{left\text{-}top}$ is unauthentic and $N_{8\times8} > 0$ or
if $MB_{top\text{-}right}$ is unauthentic and $N_{8\times8} > 0$,
then set MB as unauthentic and mark MB as a region tampered with. (13)

4. EXPERIMENTAL RESULTS

In our experiments, an MPEG-4 video with frame size 320×240 was used as the input. Four frames of the original video are shown in Figure 1. Four corresponding frames of the protected video after performing the proposed authentication signal embedding process are shown in Figure 2. Figure 3 shows imperceptive results of spatial tampering on the four frames by a modern video editing software, which crops part of the previous background image to cover the walking person in the video. Figure 4 is the authentication result of these modified frames, in which the gray areas represent the attacked areas.

5. CONCLUSIONS

In this paper, we have proposed a method for authentication of surveillance video sequences and contents by embedding authentication signals in each I frame of an input video. Authentication signals composed of the index of each GOP and the movement information of P frames in each GOP are embedded into the quantized frequency domain of each I frame according to a secret key. In order to extract authentication signals precisely from each I frame, we use a voting scheme to ensure that we can still extract correct signals when most regions of an image frame are not tampered with. The extracted authentication signals can detect not only temporal tampering on video sequences but also spatial tampering on image frames. Besides checking whether a surveillance video has been tampered with or not, the proposed authentication system also can be employed to recognize the tampering types and mark the modified regions in the surveillance video as well.

REFERENCES

[1] M. Schneider and S. F. Chang, "A Robust Content Based Digital Signature for Image Authentication," *Proc. of IEEE Intl. Conf. on Image Processing*, Lausanne, Switzerland, vol. 3, pp. 227-230, Sept. 1996.

[2] C. H. Tzeng and W. H. Tsai, "A new technique for authentication of image/video for multimedia applications," *Proc. of ACM Multimedia 2001 Workshops*, Ottawa, pp.23-26, Oct. 2001.

[3] D. He et al., "An Object Based Watermarking Solution for MPEG4 Video Authentication," *Proc. of IEEE Intl. Symp. on Acoustics, Speech, & Signal Processing*, Hong Kong, vol. 3, pp. 537-540, Apr. 2003.

[4] F. Bartolini et al., "Image Authentication Techniques for Surveillance Applications", *Proceedings of the IEEE*, vol. 89, iss. 10, pp. 1403-1418, Oct. 2001.

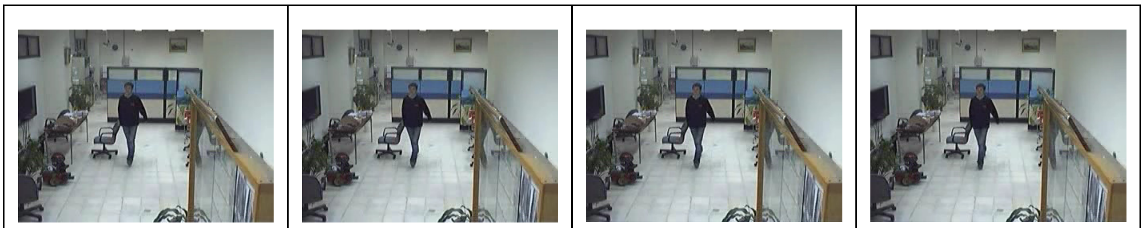
[5] P. Yin and H. H. Yu, "A Semi-fragile Watermarking System for MPEG Video Authentication," *Proc. of IEEE Intl. Conf. on Acoustics*, *Speech*, *& Signal Processing*, Orlando, Florida, USA, vol. 4, pp. 3461-3464, May 2002.

[6] R. Du and J. Fridrich, "Lossless Authentication of MPEG-2 Video," *Proc. of IEEE Intl. Conf. on Image Processing*, New York, USA, vol. 2, pp. 893-896, Sept. 2002.

[7] D. A. Winne et al., "Spatial Digital Watermark for MPEG-2 Video Authentication and Tamper Detection," *Proc. of IEEE Intl. Conf. on Acoustics*, *Speech*, *& Signal Processing*, Orlando, Florida, USA, vol. 4, pp. 3457-3460, May 2002.

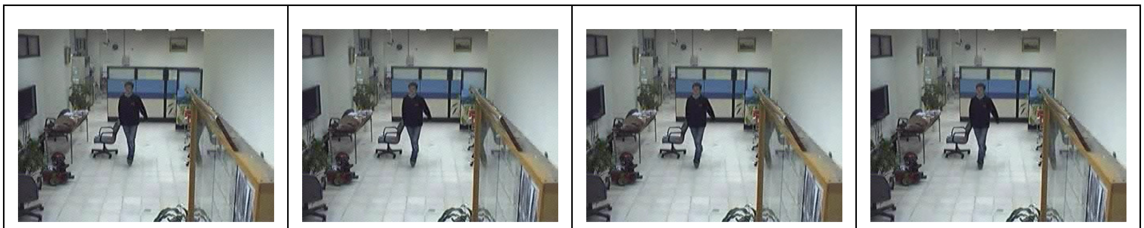
[8] S. Lee, D. Jang, and C. D. Yoo, "AN SVD-Based watermarking method for image content authentication with improved security," *Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, Philadelphia, USA, vol. 2, pp. 525-528, Mar 2005.

[9] H. Y. Chen and W. H. Tsai, "Verification of MPEG Video Contents by Random Signal Hiding," *Proc. of IPPR Conference on Computer Vision, Graphics, and Image Processing*, Kinmen, Taiwan, pp. 692-701, Aug. 16-18, 2003.



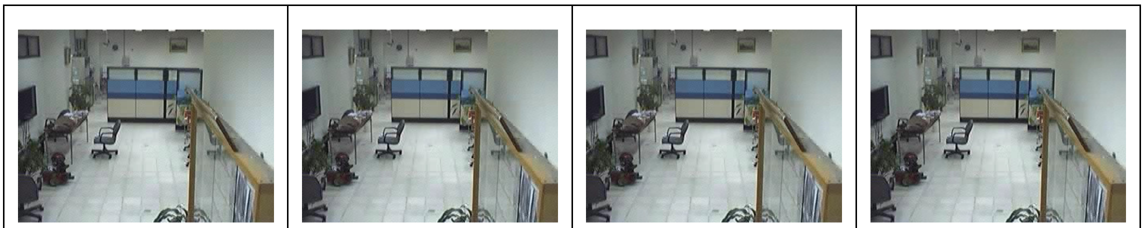
(a) (b) (c) (d)

Figure 1 Four frames of an original video. (a) First frame (I frame). (b) Second frame (P frame). (c) third frame (P frame). (d) 4-th frame (P frame)



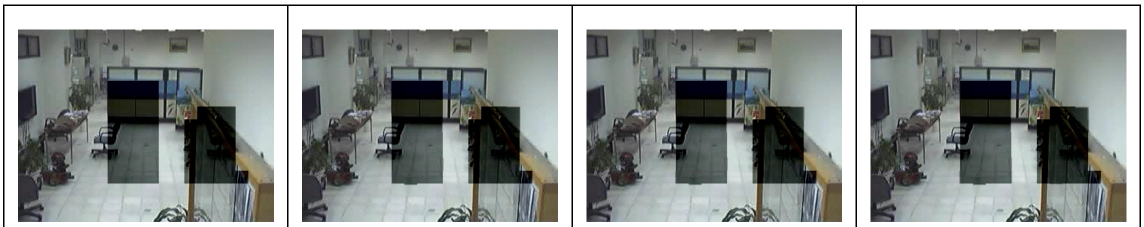
(a) (b) (c) (d)

Figure 2 Four frames of the protected video. (a) First frame (I frame). (b) Second frame (P frame). (c) third frame (P frame). (d) 4-th frame (P frame)



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

Figure 3 Four frames of a modified video of Figure 1. (a) First frame (I frame). (b) Second frame (P frame). (c) third frame (P frame). (d) 4-th frame (P frame)



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

Figure 4 Four frames of the authenticated video of Figure 3. (a) First frame (I frame). (b) Second frame (P frame). (c) third frame (P frame). (d) 4-th frame (P frame)