Chapter 4
Authentication of Surveillance Video Sequences and Contents by Hiding Motion Vector Information

4.1 Introduction

With the rapid development of the environment surveillance system, the MPEG-4 compression technique is used popularly in many applications. It raises a problem that digital videos are easier to be modified by lots of video editing software than analog ones. In addition, the recent surveillance system is usually designed for remote control through the network. In other words, the recorded surveillance videos are transmitted on the Internet to the central server and stored; therefore, these videos can be acquired and tampered with easily. Hence, it is necessary to authenticate the integrity and fidelity of received video sequences and contents at the receiver site. In this chapter, a method for authentication of surveillance video sequences and contents is proposed.

In Section 4.1.1, some related problem definitions are given, and in Section 4.1.2 the idea of the proposed method is presented. In Section 4.2, the proposed method for embedding authentication signals is described, and the proposed authentication process for video sequences and contents is described in Section 4.3. In Section 4.4, several experimental results of applying the proposed method will be shown. Finally, some discussions and a summary will be made in the last section of this chapter.

4.1.1 Problem Definition

In this study, the first task of the proposed authentication system for the surveillance video is to verify whether a given video has been tampered with or not. Malicious operations on the video can be divided into two types: *spatial tampering* and *temporal tampering*. The spatial tampering means modifications manipulated on video frame contents, and the temporal one means modifications manipulated on video frame sequences.

Temporal tampering can be further divided 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. Then the number of video frames will not be changed and the difference of the size between the former and the latter video will be too tiny to be detected visually. An illustration of frame replacement is shown in Figure 4.1. Cropping means deleting some video frames from the original video sequence. For example, a malicious user may want to eliminate his criminal evidence by cropping some video frames in the original video. An illustration of frame cropping is shown in Figure 4.2.

The third type of tampering, insertion, means to add fake video frames into the original video. For example, a malicious user might insert some fake frames to pin his crime on someone else who is innocent. An illustration of frame insertion is shown in Figure 4.3. The main task of the proposed authentication system is not only to detect whether a surveillance video has been tampered with or not, but also to recognize the tampering type and mark further the regions tampered with.

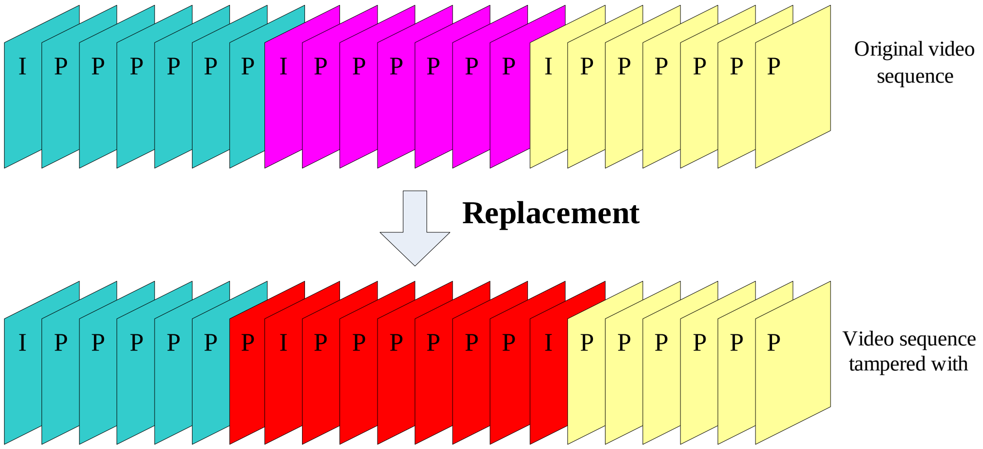

I P P P P P P I P P P P P P I P P P P P P
Original video sequence
Replacement
I P P P P P P I P P P P P P I P P P P P P
Video sequence tampered with

Figure 4.1 Illustration of replacement.

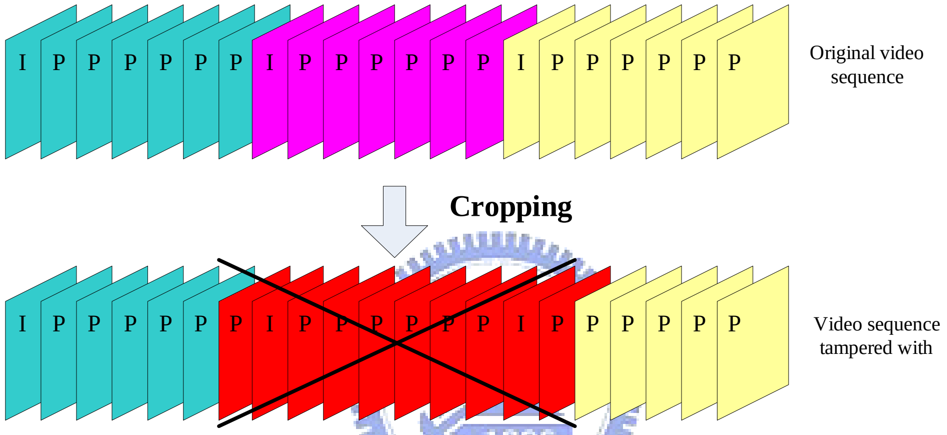

I P P P P P P I P P P P P P I P P P P P P
Original video sequence
Cropping
I P P P P P P I P P P P P P I P P P P P P
Video sequence tampered with

Figure 4.2 Illustration of cropping.

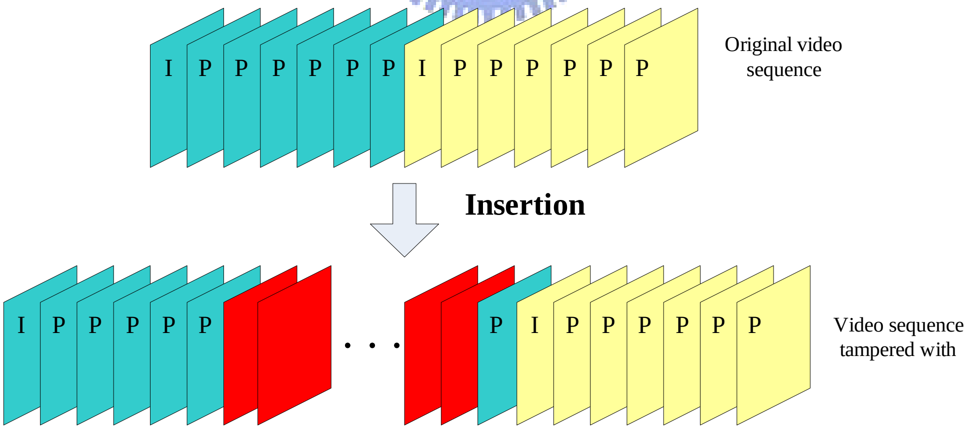


Figure 4.3 Illustration of insertion.

4.1.2 Proposed Idea of Authentication

In order to detect spatial tampering, some signals, called *authentication signals*, are embedded into the DCT coefficients of each 8×8 luminance block in each I frame of a video sequence. Authentication signals are composed of two types of features in each *GOP* (group of pictures) 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.

4.2 Embedding of Authentication Signals in Surveillance Videos

In this section, the proposed authentication signal embedding method is described. In Section 4.2.1, the principle is described in detail, and the process of generating authentication signals by analyzing motion vectors in P frames is described in Section 4.2.2. In Section 4.2.3, the process of embedding authentication signals in I frames is presented.

4.2.1 Principle 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 embedding authentication signals 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 to be embedded into each I frame. An illustration of the proposed process for embedding authentication signals is shown in Figure 4.4.

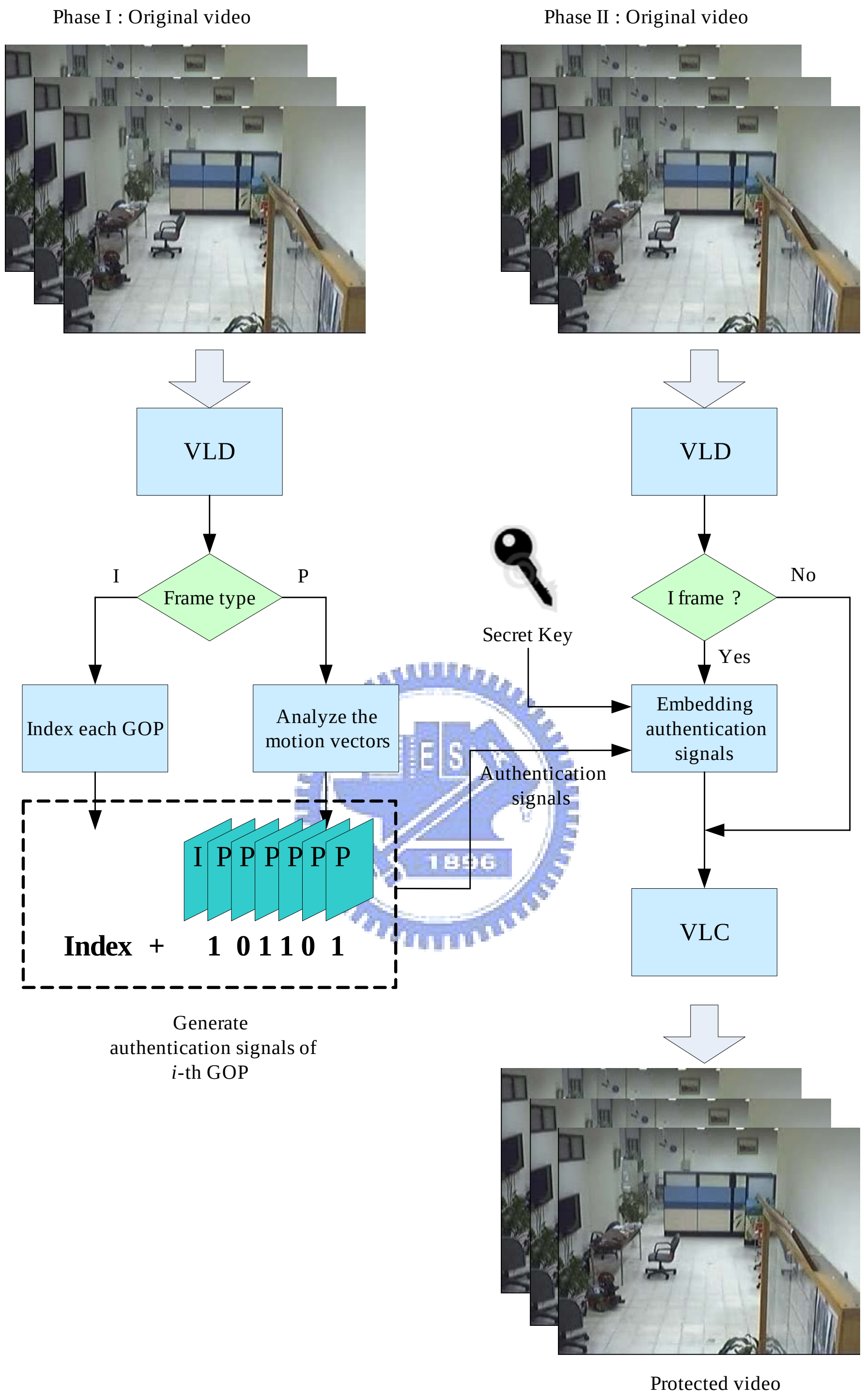


Figure 4.4 Illustration of the proposed authentication signal embedding method

4.2.2 Process for Generating Authentication Signals by Analyzing Motion Vectors in P Frames

Besides the index of the GOP, we also need 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 P frame. Then we can form a binary string to represent the movement information of the six P frames in the *i*-th GOP, where the bit 0 represents a still P frame and the bit 1 represents a motion P frame. A flowchart of the process for analyzing the motion vectors in P frames is shown in Figure 4.5 and a corresponding detailed algorithm is described in the following.

***Algorithm* 4.1**: Authentication signal generation process 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:

$$\begin{cases} if\ mv_x > T_1\ or\ mv_y > T_1, then\ set\ N(f_j) = N(f_j) + 1 \\ otherwise,\ set\ N(f_j)\ unchanged \end{cases} \tag{4.1}$$

   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:

$$G_{avr} = \frac{1}{N_p}\sum_{j=1}^{N_p} N(f_j) \tag{4.2}$$

   where N_p is the number of P frames in G_i.

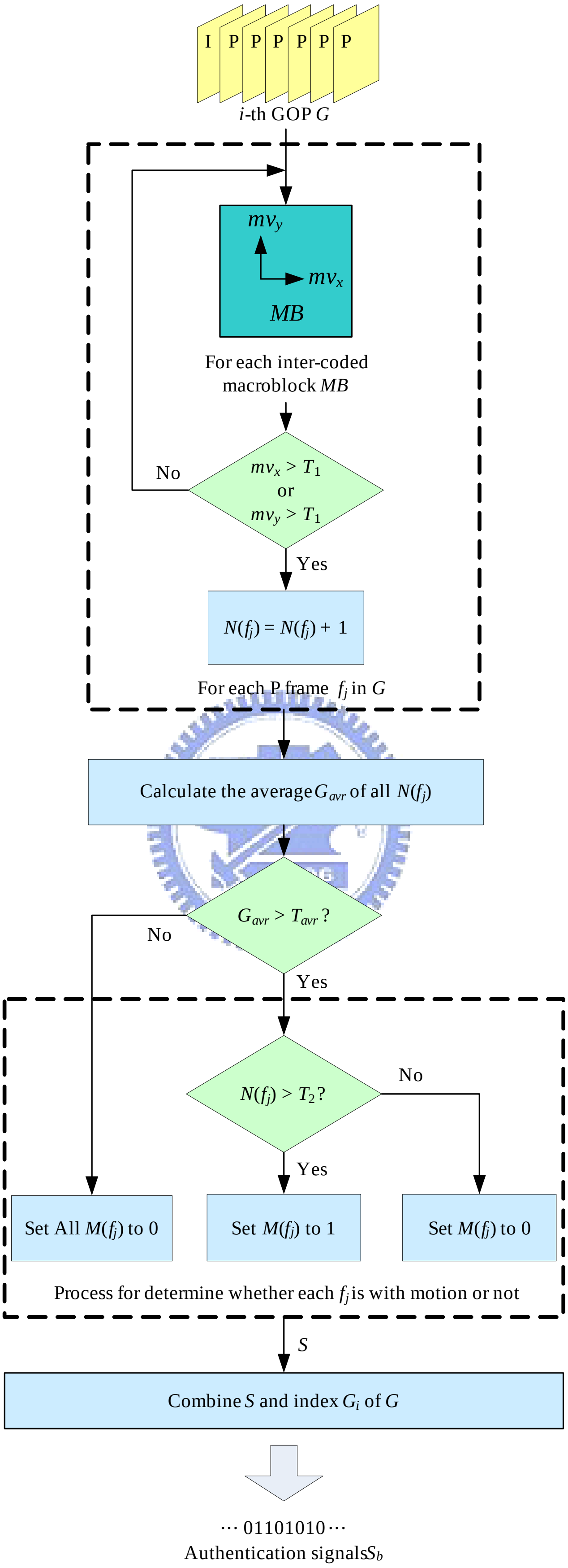

I P P P P P P P
i-th GOP G
mv_y
mv_x
MB
For each inter-coded macroblock MB
$mv_x > T_1$ or $mv_y > T_1$
No
Yes
$N(f_j) = N(f_j) + 1$
For each P frame f_j in G
Calculate the average G_{avr} of all $N(f_j)$
$G_{avr} > T_{avr}$?
No
Yes
$N(f_j) > T_2$?
No
Yes
Set All $M(f_j)$ to 0
Set $M(f_j)$ to 1
Set $M(f_j)$ to 0
Process for determine whether each f_j is with motion or not
S
Combine S and index G_i of G
⋯ 01101010 ⋯
Authentication signals S_b

Figure 4.5 Flowchart of process for generating authentication signals

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 $G_{avr} > T_{avr}$:

$$\begin{cases} if\ \ N(f_j) > T_2, then\ set\ M(f_j) = 1 \\ if\ \ N(f_j) < T_2, then\ set\ M(f_j) = 0 \end{cases} \tag{4.3}$$

   (2) When $G_{avr} < T_{avr}$:

$$\text{Set all the } M(f_i) = 0 \tag{4.4}$$

   where T_{avr} and T_2 are pre-defined thresholds.

4. For each P frame f_j of G_i, combine each $M(f_j)$ of f_j to form a binary string S.
5. Transfer the index of G_i into the 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 changes of the light and shadows may cause huge motion vectors in P frames. Therefore, the main task of the threshold T_{avr} is to avoid erroneous judgment by abnormal motion vectors.

4.2.3 Process for Embedding Authentication Signals in I Frames

After the first phase of the proposed process of embedding authentication signals, we obtain the authentication signals of each GOP of the video. Before beginning the second phase, the authentication signals in binary form were 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. Then we utilize four pairs of DCT coefficients selected randomly from eight pre-defined ones mentioned in Chapter 3 to embed the four bits of the authentication signals in an 8×8 luminance block with a secret key. A flowchart of the process for embedding authentication signals is illustrated in Figure 4.6 and a detailed algorithm is described in the following.

***Algorithm* 4.2**: Authentication signals embedding process for I frames.

Input: an I frame F, authentication signals of binary form S_b, and a secret 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 secret key K and the position P of B to form a seed for a random number generation.
3. 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 $b_i = 0$:

   $$\begin{cases} if\ \ C_1[i] < C_2[i], then\ swap\ C_1[i]\ and\ C_2[i] \\ if\ \ C_1[i] = C_2[i], then\ set\ C_1[i] = C_2[i] + T_3 \end{cases} \tag{4.5}$$

   where T_3 is a pre-defined threshold.

   (2) When $b_i = 1$:

   $$\begin{cases} if\ \ C_1[i] > C_2[i], then\ swap\ C_1[i]\ and\ C_2[i] \\ if\ \ C_1[i] = C_2[i], then\ set\ C_2[i] = C_1[i] + T_3 \end{cases} \tag{4.6}$$

   where T_3 is a pre-defined threshold.

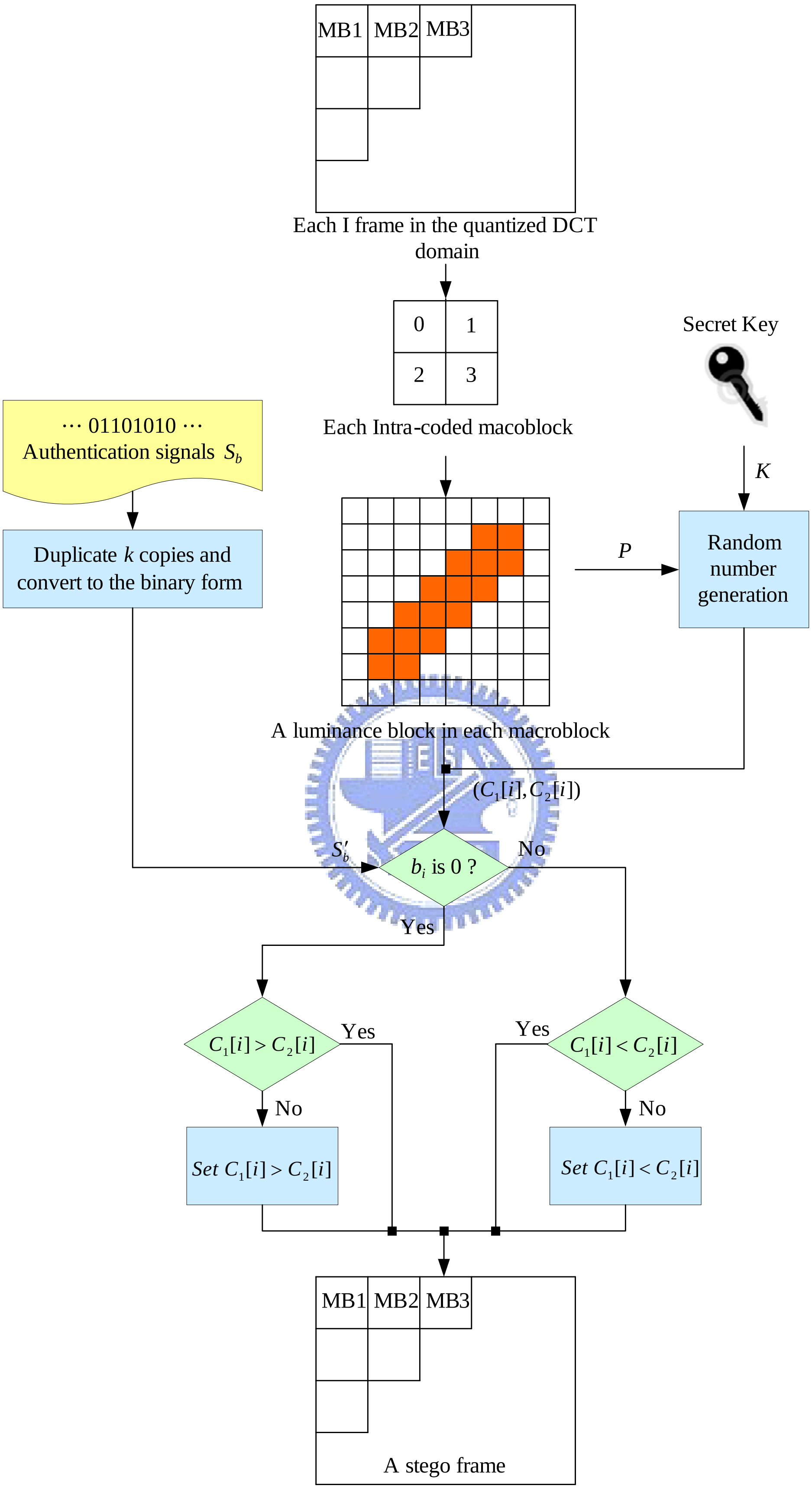

MB1
MB2
MB3
Each I frame in the quantized DCT domain
0
1
2
3
Each Intra-coded macoblock
Secret Key
$\cdots$ 01101010 $\cdots$
Authentication signals S_b
K
Duplicate k copies and convert to the binary form
P
Random number generation
A luminance block in each macroblock
$(C_1[i], C_2[i])$
S_b'
b_i is 0 ?
No
Yes
$C_1[i] > C_2[i]$
Yes
Yes
$C_1[i] < C_2[i]$
No
No
Set $C_1[i] > C_2[i]$
Set $C_1[i] < C_2[i]$
MB1
MB2
MB3
A stego frame

Figure 4.6 Flowchart of the process for embedding authentication signals.

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 the MPEG recompression; however, the resulting video quality suffers more degradation as well.

4.3 Authentication of Video Sequences and Contents

In this section, the proposed authentication method of video sequences and contents will be described. An illustration of this method is shown in Figure 4.7. In Section 4.3.1, the idea for authentication is stated. In Section 4.3.2, the process of extracting authentication signals is presented. Next, the process for detection and verification of temporal tampering is described in Section 4.3.3. Finally, the process for detection and verification of spatial tampering is described in Section 4.3.4.

4.3.1 Idea for Authentication

The process for authentication of video sequences and contents also can be divided into two phases. In the first phase, we use a voting technique 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 the 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.

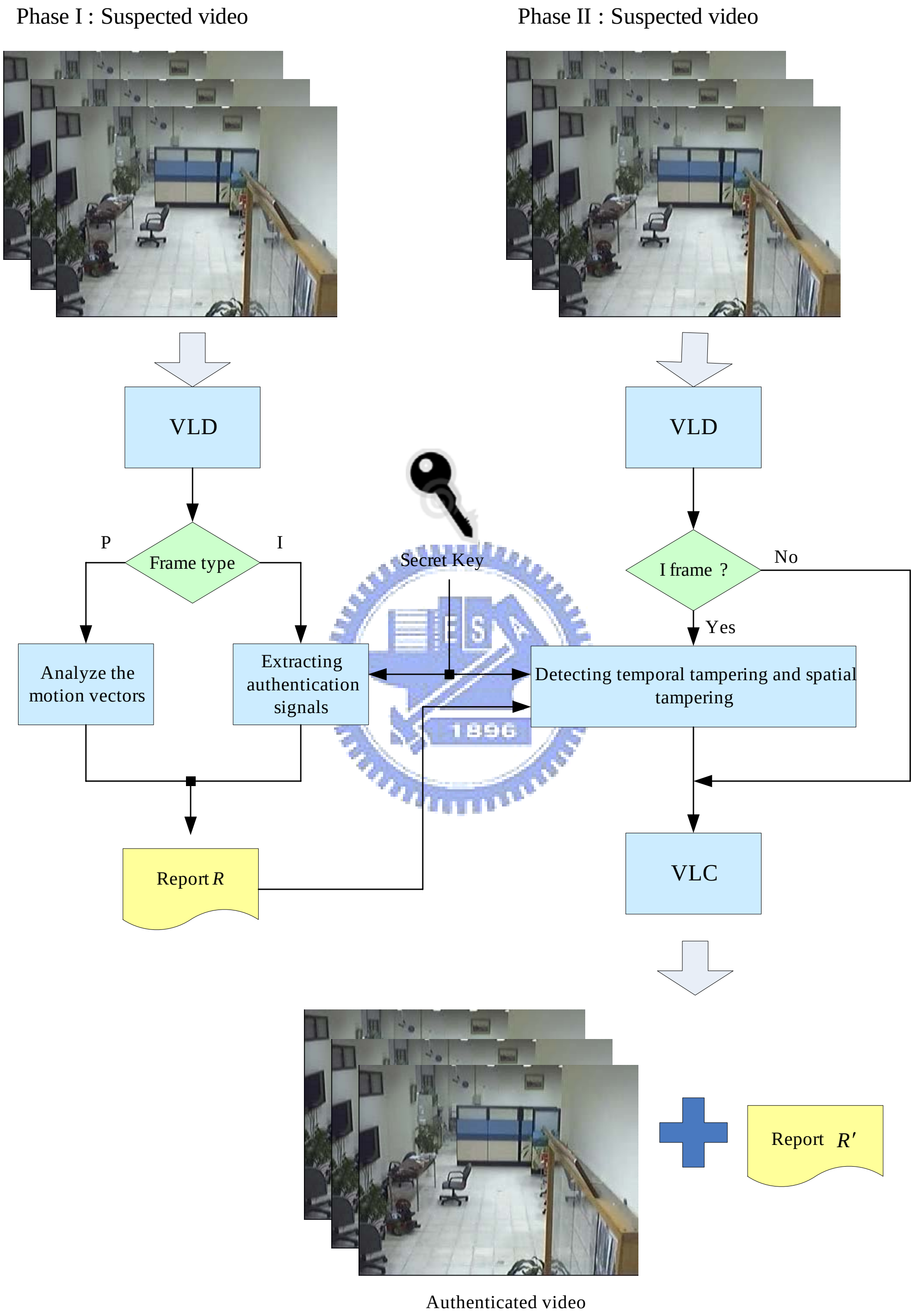

Phase I : Suspected video
Phase II : Suspected video
VLD
VLD
Frame type
P
I
Secret Key
I frame ?
No
Yes
Analyze the motion vectors
Extracting authentication signals
Detecting temporal tampering and spatial tampering
Report R
VLC
Report R′
Authenticated video

Figure 4.7 Illustration of the proposed video contents and sequences authentication method.

4.3.2 Process for Extracting Authentication Signals Using Voting Technique

In this study, the method utilized to extract authentication signals is called a voting technique. In the previous embedding process of authentication signals, we embedded 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 technique 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. A flowchart of the process for authentication signal extraction is shown in Figure 4.8 and a corresponding detailed algorithm is described in the following.

***Algorithm* 4.3**: Authentication signal extraction process.

Input: a protected I frame F' and a secret key *K*.

Output: the correct authentication signals *S*.

Steps:

1. For each 8×8 luminance block *B*, combine the input secret key *K* and the position *P* of *B* to form a seed for a random number generation.
2. Use the result of the random number generation to randomly select four pairs ($C_1[k]$, $C_2[k]$) of DCT coefficients from eight pre-defined ones and extract

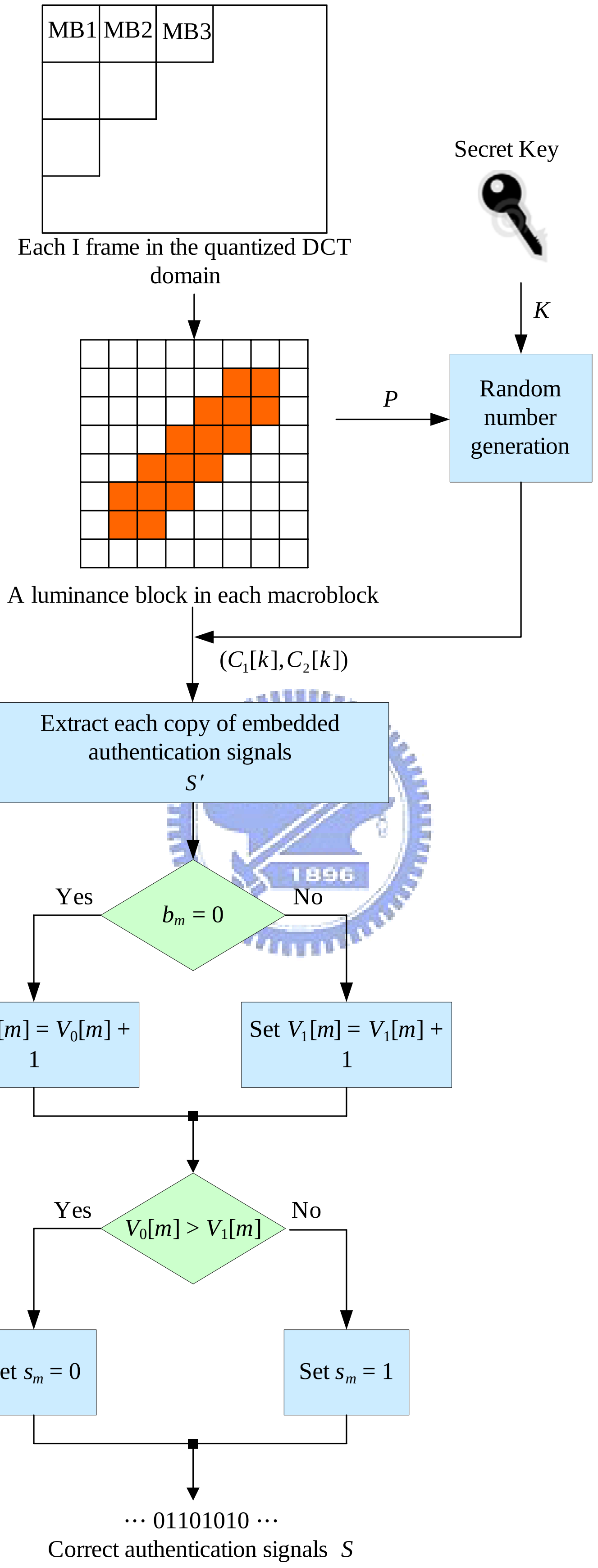

MB1
MB2
MB3
Each I frame in the quantized DCT domain
Secret Key
K
P
Random number generation
A luminance block in each macroblock
$(C_1[k], C_2[k])$
Extract each copy of embedded authentication signals S'
Yes
$b_m = 0$
No
Set $V_0[m] = V_0[m] + 1$
Set $V_1[m] = V_1[m] + 1$
Yes
$V_0[m] > V_1[m]$
No
Set $s_m = 0$
Set $s_m = 1$
⋯ 01101010 ⋯
Correct authentication signals S

Figure 4.8 Illustration of extracting authentication signals by the voting technique.

four bits b_k of each copy of signals S' according to the following rule:

$$\begin{cases} if\ C_1[k] > C_2[k], then\ set\ b_k = 0; \\ if\ C_1[k] > C_2[k], then\ set\ b_k = 1. \end{cases} \tag{4.7}$$

where $k = 0$ through 3.

3. For each copy of the extracted authentication signals S', denote the binary form of S' as $S' = b_1b_2b_3\ldots b_L$, where L is the length of signals. Assign each bit of S' two voting scores $V_0[m]$ and $V_1[m]$, where $1 \le m \le L$. Calculate the score of each bit of authentication signals according to the follow rule.

$$\begin{cases} if\ b_m = 0, then\ set\ V_0[m] = V_0[m] + 1; \\ if\ b_m = 1, then\ set\ V_1[m] = V_1[m] + 1. \end{cases} \tag{4.8}$$

4. Denote the binary form of the correct authentication signals S as $S = s_1s_2s_3\ldots s_L$. Reconstruct S by comparing two scores of each bit of S according to the following rule:

$$\begin{cases} if\ V_0[m] > V_1[m], then\ set\ s_m = 0; \\ if\ V_1[m] > V_0[m], then\ set\ s_m = 1. \end{cases} \tag{4.9}$$

where $1 \le m \le L$.

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 frames. The detailed algorithm is similar to the one described in Section 4.2.2. Moreover, we store the extracted authentication signals and the movement information of statistics into a temporary report R for the subsequent authentication process.

4.3.3 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 4.3.2, 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 GOP as G_i and compare it with G_i' to detect the temporal tampering of the input video.

In the proposed method, we classify the temporal tampering into two main types: one is replacement, and the other is 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 tampered segment. A corresponding detailed algorithm is described in the following.

***Algorithm* 4.4**: The process for detecting the temporal tampering.

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

Output: a report R' of detection results

Steps:

1 Declare a flag bit b for the occurrence of tampering, and initialize b to 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, 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 $G_i$ and
              perform the following steps:
            4.1.1 Recognize the tampering type as replacement and store it into $R'$.
            4.1.2 Store $ns$ and $nf$ 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 Step 1 to Step 4 for each GOP until reaching the end-of-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 current index G_i, we set b to 1 because the GOP with index G_i has been tampered with. As the proceeding of the authentication process, if the extracted index G_i' equals the current index G_k, where k is greater than i, an approximate tampered segment of replacement tampering is detected and b is reset to 0 for detecting the next tampered segment.

Sometimes, we can detect the tampered segment of insertion or cropping tampering by means of the method similar to the one mentioned above when the number of tampered frames is a multiple of the number of frames in a GOP. But the probability of this condition is extremely small.

4.3.4 Detection and Verification of Spatial Tampering for Suspected Videos

Because most frames of surveillance videos are still background without moving objects, this kind of frame might be used to replace frames including criminal behaviors by malicious users. They even might cut some regions from background frames and paste them to 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 technique 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 will 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 will 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 the index G_i' is recognized as a suspected GOP. A flowchart of the process for the spatial authentication is shown in Figure 4.9 and a corresponding detailed algorithm is described in the following.

***Algorithm* 4.5**: The process for detecting spatial tampering.

Input: a suspected GOP G_s with I frame F', extracted authentication signals S and a secret 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 through 3, combine the input secret key K and the position P of B_i to form a seed for a random number generation.

3. According to the result of the random number generation, 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:

$$\begin{cases} if \;\; C_1[i] > C_2[i], \; then \;\; set \;\; b_i = 0; \\ if \;\; C_1[i] < C_2[i], \; then \;\; set \;\; b_i = 1. \end{cases} \tag{4.10}$$

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:

$$\begin{cases} if \; b_i \neq s_i, then \; set \; N_b = N_b + 1; \\ otherwise, set \; N_b \; unchanged. \end{cases} \tag{4.11}$$

where N_b is the number of suspected bit in B_i.

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

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

   (1) Rule 1:

$$\begin{cases} if \;\; B_0 \; and \; B_3 \; is \; suspected \\ or \; B_1 \; and \; B_2 \; is \; suspected \end{cases} \tag{4.12}$$

   then set MB is unauthentic and mark MB as a tampered region.

   (2) Rule 2:

$$\begin{cases} if \;\; MB_{left} \; and \; MB_{top} \; is \; unauthentic \\ or \; MB_{left} \; is \; unauthentic \; and \; N_{8\times8} > 0 \\ or \; MB_{top} \; is \; unauthentic \; and \; N_{8\times8} > 0 \\ or \; MB_{left-top} \; is \; unauthentic \; and \; N_{8\times8} > 0 \\ or \; MB_{top-right} \; is \; unauthentic \; and \; N_{8\times8} > 0 \end{cases} \tag{4.13}$$

   then set MB as unauthentic and mark MB as a tampered region.

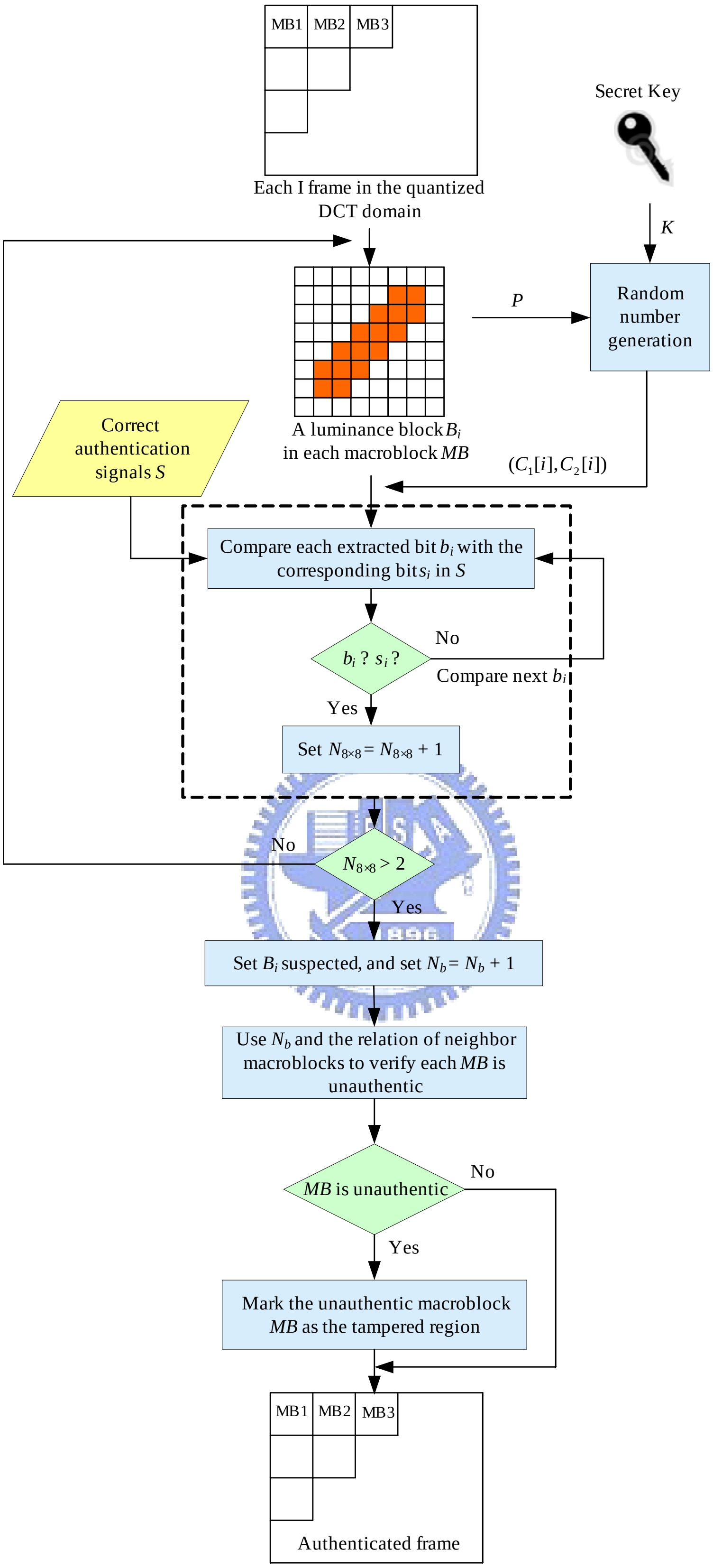
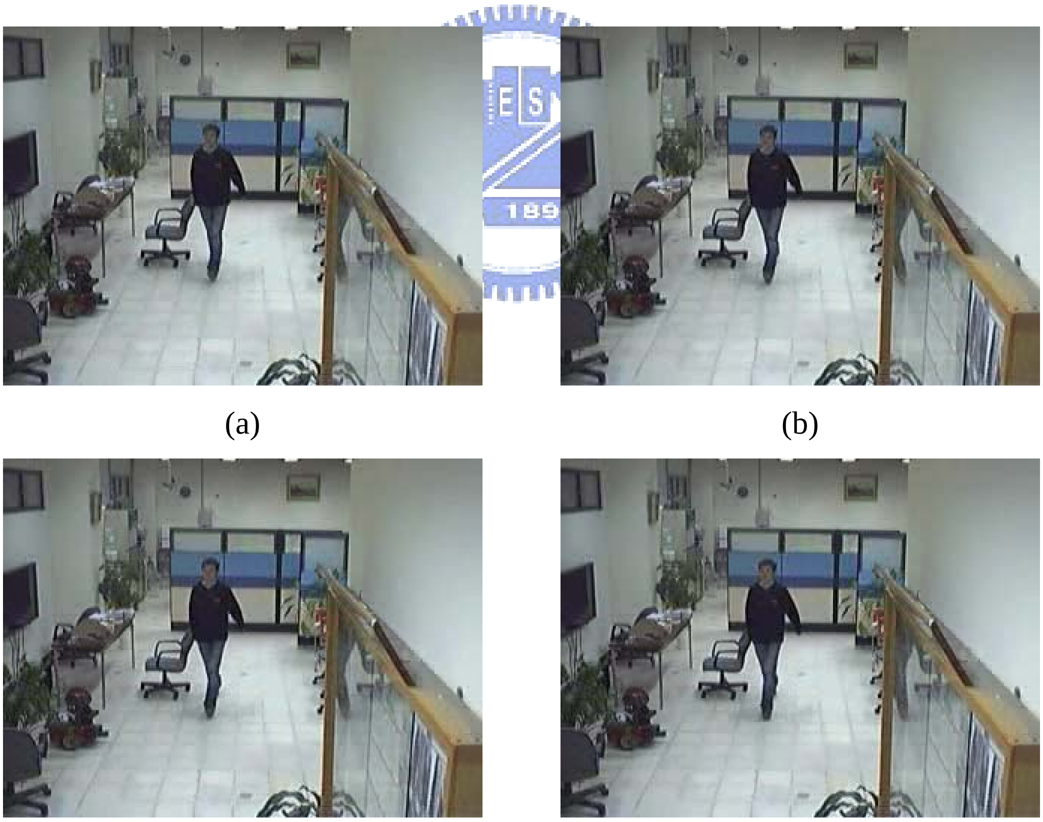

MB1
MB2
MB3
Each I frame in the quantized DCT domain
Secret Key
K
P
Random number generation
A luminance block B_i in each macroblock MB
Correct authentication signals S
$(C_1[i], C_2[i])$
Compare each extracted bit b_i with the corresponding bit s_i in S
b_i ? s_i ?
No
Compare next b_i
Yes
Set $N_{8\times8} = N_{8\times8} + 1$
No
$N_{8\times8} > 2$
Yes
Set B_i suspected, and set $N_b = N_b + 1$
Use N_b and the relation of neighbor macroblocks to verify each MB is unauthentic
MB is unauthentic
No
Yes
Mark the unauthentic macroblock MB as the tampered region
MB1
MB2
MB3
Authenticated frame

Figure 4.9 Illustration of detecting and verifying spatial tampering of suspected videos.

4.4 Experimental Results

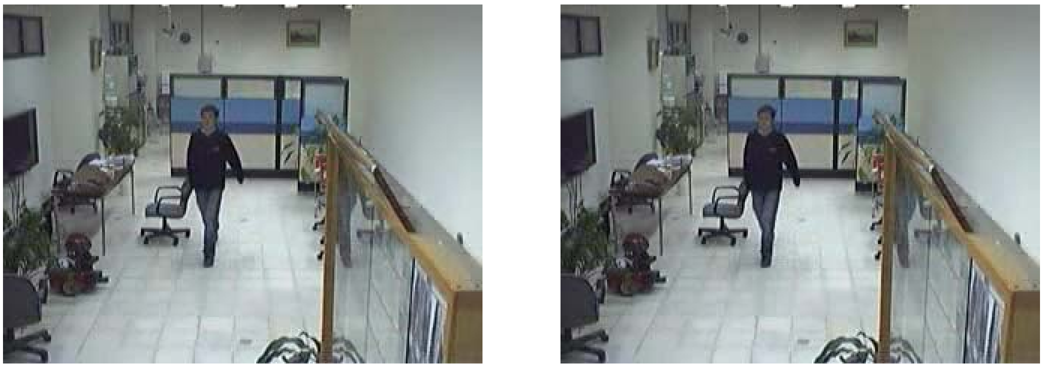
In out experiments, an MPEG-4 video with frame size 320×240 was used as the input. Six frames of the original video are shown in Figure 4.10. Six corresponding frames of the protected video after performing the proposed authentication signal embedding process are shown in Figure 4.11. Figure 4.12 shows imperceptive results of spatial tampering on the six frames by a modern video editing software, which crops a part of the previous background image to cover the walking person in the video to remove him. Figure 4.13 is the authentication result of these tampered frames, in which the gray areas represent the attacked areas.



(a) (b)

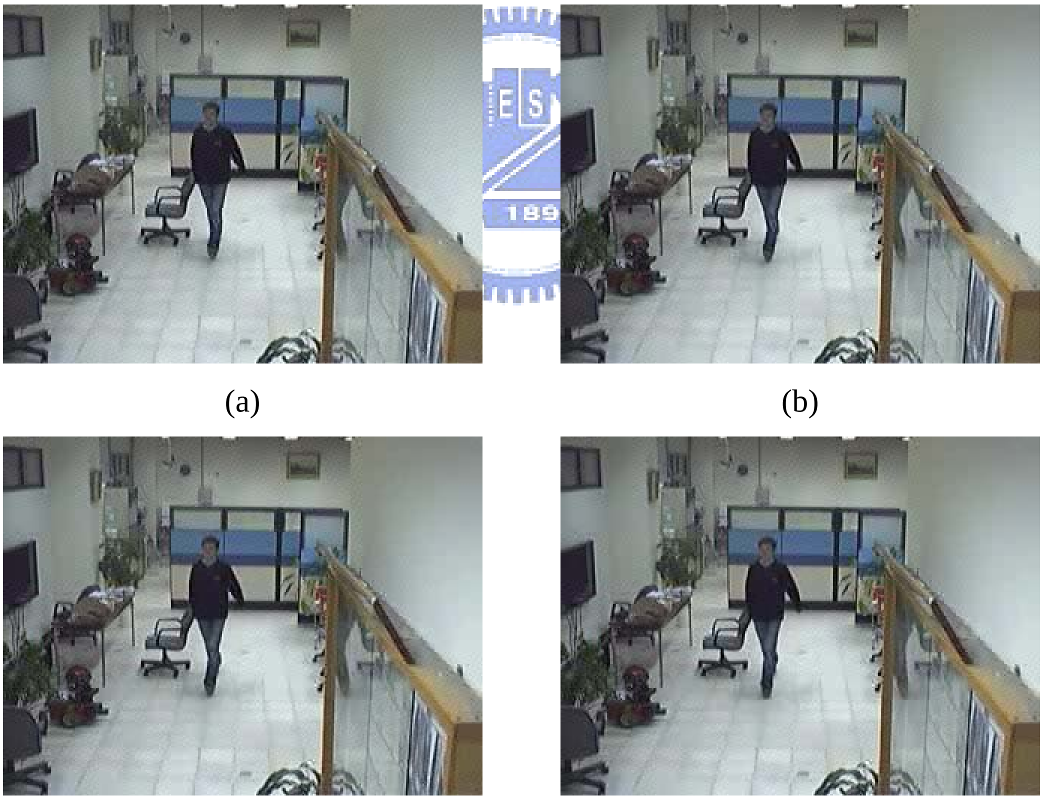
(c) (d)

Figure 4.10 Six frames of the original video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(e) (f)

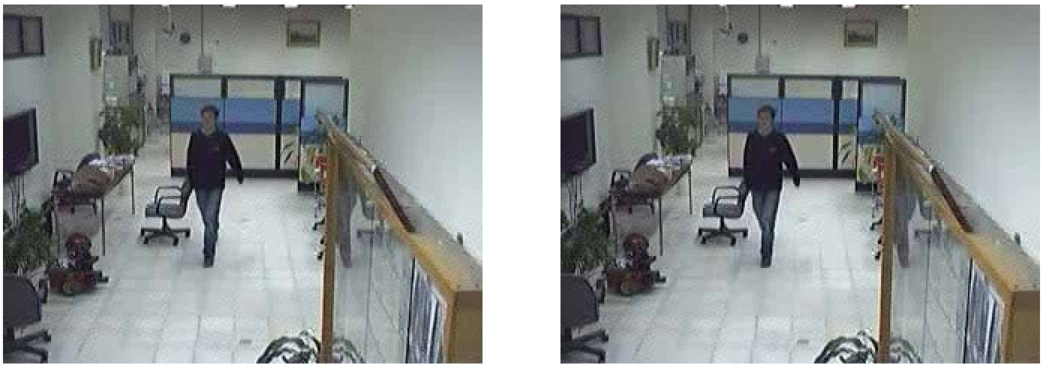
Figure 4.10 Six frames of the original video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(a) (b)

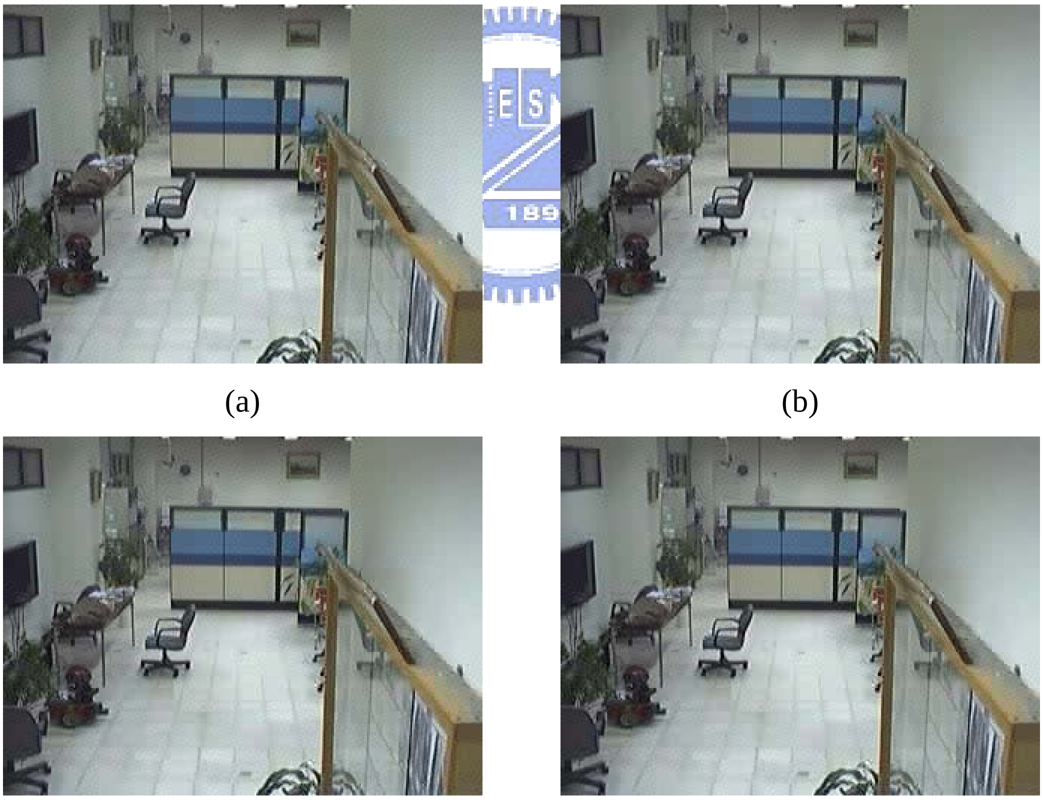
(c) (d)

Figure 4.11 Six frames of the protected video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(e) (f)

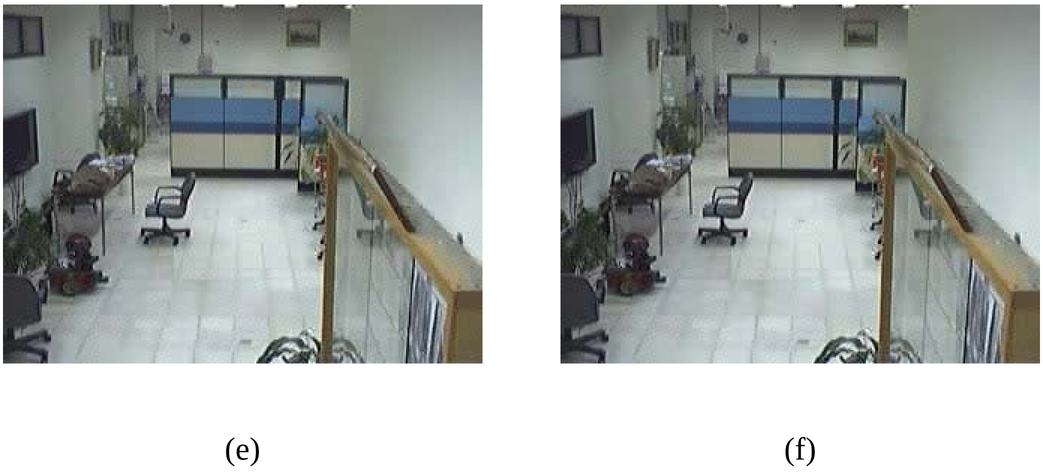
Figure 4.11 Six frames of the protected video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(a) (b)

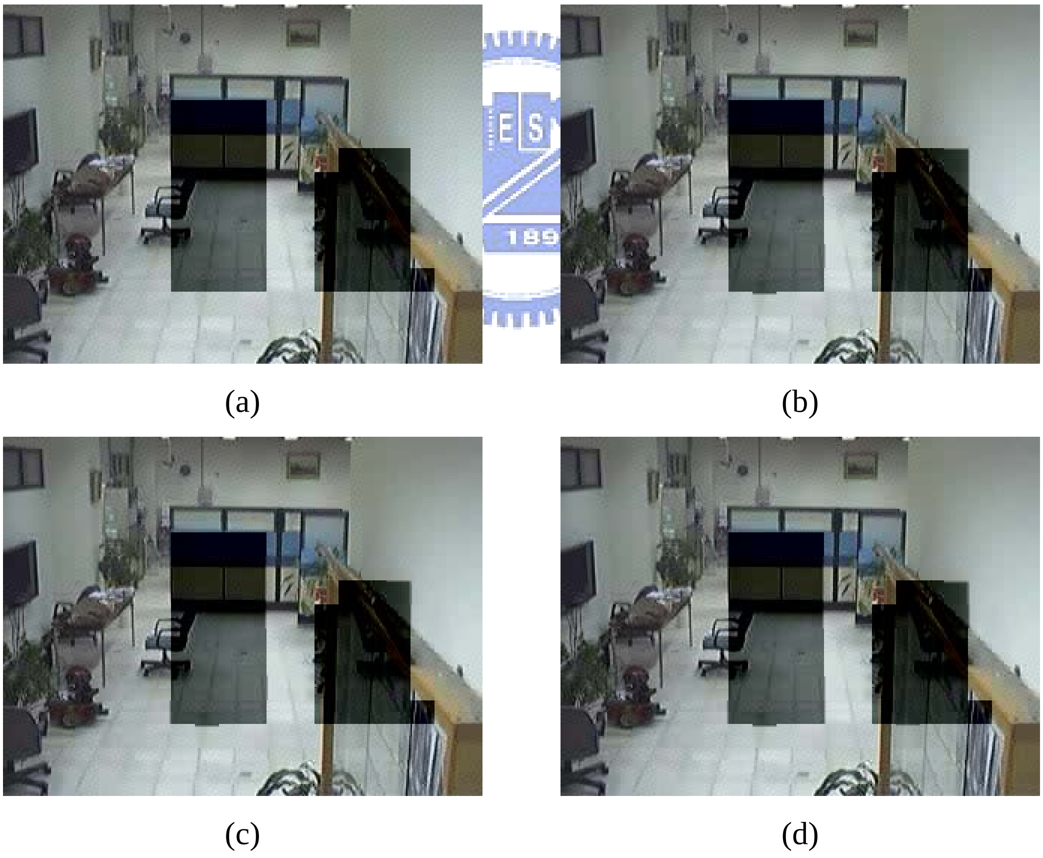
(c) (d)

Figure 4.12 Six frames of the tampered video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(e) (f)

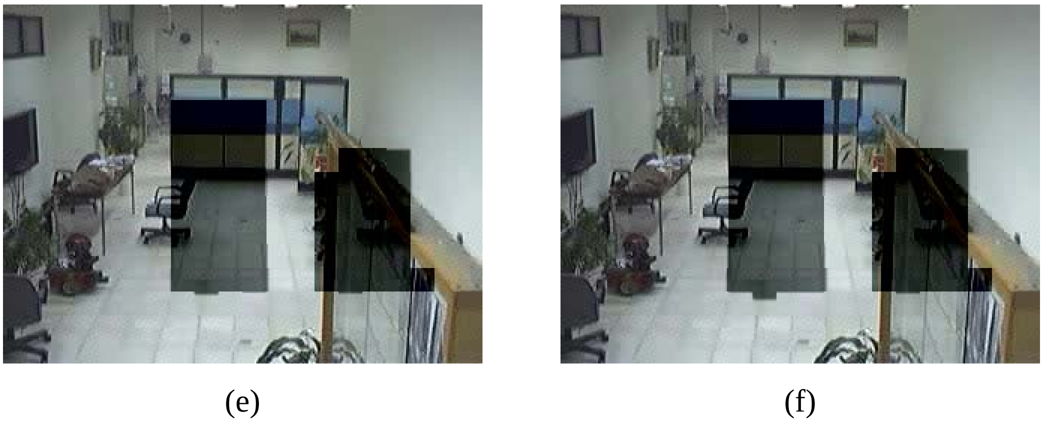
Figure 4.12 Six frames of the tampered video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(a) (b)

(c) (d)

Figure 4.13 Six frames of the authenticated video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)



(e) (f)

Figure 4.13 Six frames of the authenticated video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4-th frame (P frame) (e) The 5-th frame (P frame). (f) The 6-th frame (P frame) (continued)

4.5 Discussion and Summary

In this chapter, we have proposed a method for authentication of surveillance video sequences and contents by embedding authentication signals in each I frame. 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 the voting technique to make sure 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 the temporal tampering performed on video sequences but also the spatial tampering performed on image frames. Except for checking whether a surveillance video has been tampered with or not, the proposed authentication system can recognize the tampering types and mark the tampering regions in the surveillance video as well.