DDCA: A Distortion Drift-Based Cost Assignment Method for Adaptive Video Steganography in the Transform Domain

Yi Chen, Hongxia Wang, Kim-Kwang Raymond Choo, *Senior Member, IEEE,* Peisong He, Zoran Salcic, *Life Senior Member, IEEE,* Dali Kaafar, Xuyun Zhang

Abstract—Cost assignment plays a key role in coding performance and security of video steganography. Existing cost assignment methods (for adaptive video steganography) are designed for specific transform coefficients rather than all transform coefficients. In addition, existing video steganographic frameworks do not allow Syndrome-Trellis Codes (STCs) to modify all transform coefficients in both intra-coded and inter-coded frames at the same time. To address these limitations, in this paper, we first propose a novel video steganographic framework. Then, we give a theoretical analysis of distortion drift in both intra- and inter-coding procedures. Based on the analysis, we design a Distortion Drift-based Cost Assignment method, hereafter referred to as DDCA. DDCA considers the inner-block, inter-block and inter-frame distortion costs in order to improve the coding performance and the security of stego videos when the embedding payload is fixed. We conducted extensive experiments using two video datasets to evaluate the proposed video steganographic framework and DDCA, in terms of the coding performance and the security. Our experiments show that the proposed framework outperforms three recent state-of-the-art methods, for example the coding performance and the security of stego videos can benefit from DDCA by making full use of all nonzero transform coefficients.

Index Terms—Adaptive video steganography, distortion drift, minimal distortion, Syndrome-Trellis Codes (STCs), H.264.

1 Introduction

MODERN steganography is the art and science of covert communication that slightly changes the digital media, such as image, video, audio, and text, to hide secret messages and circumvent steganalysis efforts [1]. In recent years, successful steganography schemes have been mostly based on the minimum distortion embedding framework [2], in which each cover element is assigned a distortion cost according to the relationship between the cover element and its adjacent cover elements. In general, this embedding framework embeds secret messages into the textured and noisy regions for maximizing its effectiveness against steganalysis. The assigned distortion cost for each cover element measures the impact of changing it; thus, defining an additive distortion cost function as the sum of costs for all cover elements. Syndrome-Trellis-Codes (STCs) [2] can perform well in minimizing the additive distortion cost, and hence efforts have been dedicated to designing additive cost functions for expressing the real modification effects. Since the work by Filler et al. [2], a number of additive distortion cost functions, such as HUGO (Highly Undetectable steGO) [3], WOW (Wavelet Obtained Weights) [4], SMD (Synchronizing Modification Direction) [5], UNIWARD (UNIversal WAvelet Relative Distortion) [6], J-MSUNIWARD [7], have been proposed for grayscale image steganography and JPEG (Joint Photographic Experts Group) image steganography.

As video coding and communications technologies advance and the increasing popularity of mobile devices and applications (including video applications), digital videos have become more commonplace. Digital videos have rich video entities and high embedding capacity, which can be leveraged for steganography.

Generally speaking, video steganography can be classified into spatial domain steganography and compressed domain steganography [8, 9]. Most steganographic schemes in spatial domain leverage modification methods adopted in image steganography [10]. That is to say, only raw pixels of videos are changed for steganography. In contrast, compressed domain steganography has various kinds of video cover elements, such as intra prediction mode [11, 12], inter prediction mode [13–15], motion vector [15–19], quantization parameter [20, 21] and Quantized Discrete Cosine Transform (QDCT) coefficient [22–28], for steganography. Correspondingly, a number of steganalysis methods have been proposed in the literature [29–33]. Unlike compressed domain steganography, spatial domain steganography is not capable of correctly extracting the embedded massages because the quantization operation is lossy.

In compression domain steganography, cost assignment

- *Y. Chen is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China, also with the Department of Electrical, Computer and Software Engineering, The University of Auckland, Auckland 1010, New Zealand (e-mail: yichen.research@gmail.com)*
- *H. Wang and P. He are with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China (e-mail: {hxwang,gokeyhps}@scu.edu.cn)*
- *K.-K. R. Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA. He is also with the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA. (e-mail: raymond.choo@fulbrightmail.org)*
- *Z. Salcic is with the Department of Electrical, Computer and Software Engineering, The University of Auckland, Auckland 1010, New Zealand (e-mail: z.salcic@auckland.ac.nz)*
- *D. Kaafar and X. Zhang are with the Department of Computing, Faculty of Science and Engineering, Macquarie University, NSW 2109, Australia (e-mail:{dali.kaafar,xuyun.zhang}@mq.edu.au)*
- *Corresponding authors: Peisong He and Xuyun Zhang*

plays a very important role in coding performance and security of stego videos. Different cost assignment schemes have been designed for different embedding domains [11, 14, 17, 23, 26–28]. However, in transform domain, currently existing video steganographic frameworks, such as [23, 28], are based on video coding, such as [34–37]. In these video coding frameworks, videos are encoded block by block. Thus, such video steganographic frameworks are designed to only allow STCs to change transform coefficients block by block (or frame by frame). In other words, STCs does not obtain full freedom from these frameworks to change all transform coefficients of a whole video for steganography. This limits the full use of STCs, thus further limiting the coding performance and the security of stego videos. In addition, existing cost assignment methods [23, 26, 27] based on these video steganographic frameworks are designed for specific transform coefficients rather than all transform coefficients. For instance, the authors of [23, 27] respectively propose cost assignment schemes for particular transform coefficients in I frames, and the cost assignment scheme in [26] is designed for that in P frames. Namely, all transform coefficients are not fully considered and exploited for steganography. Totally, the limitations of existing video steganographic frameworks and the cost assignment methods based on these frameworks limit the coding performance and the security of stego videos.

Therefore, to provide STCs full freedom to select the transform coefficients with less distortion costs in an entire video for steganography, we propose a novel video steganographic framework in this paper (see Section 2). The proposed video steganographic framework contains two encoders. One is exploited to calculate distortion costs and obtain cover elements for steganography and the other is used to update the corresponding cover elements for obtaining stego bitstreams. Thereby, the proposed video steganographic framework solves the limitation of existing video steganographic frameworks. In addition, to design a cost assignment method for all transform coefficients, we first give a theoretical analysis of distortion drift in both intra- and inter-coding procedures, including inner-block distortion drift, inter-block distortion drift, and inter-frame distortion drift, in Section 3. Based on the theoretical analysis, we design a novel cost assignment method, hereafter referred to as DDCA, from the distortion drift point of view for all transform coefficients in Section 4. The main idea of DDCA is that, the more significant distortion changing one transform coefficient by adding or subtracting 1 leads to, the larger cost changing this coefficient for steganography has. In other words, in DDCA the cost assignment of each transform coefficient depends on the impact level of the distortion drift. To prevent bit-rate from significantly increasing and to obtain better coding performance and security of stego videos, we apply DDCA in all nonzero transform coefficients, i.e, all nonzero QDCT AC coefficients. Experiments are conducted to demonstrate the effectiveness of the proposed video steganographic framework and DDCA to improve the coding performance and the security of stego videos. Specifically, the findings show that DDCA outperforms the cost assignments methods presented in [23, 26, 27] in terms of both the coding performance and the security. Furthermore, the results demonstrate that fully utilizing all nonzero transform coefficients improves the coding performance and the security of stego videos when given the embedding payloads.

The main contributions of this paper are summarized as follows:

1) We propose a novel video steganographic framework, which provides full freedom for STCs in selecting cover elements from all nonzero transform coefficients for both intra-coded and inter-coded frames, while also achieving improved coding performance and security.
2) Based on the theoretical analysis of drift distortion in both intra- and inter-coding procedure, DDCA method is designed by comprehensively considering inner-block, inter-block and inter-frame distortion costs. Such a design more accurately reflects the real modification distortion.
3) By considering various video contents and coding parameter settings, we perform extensive experiments to demonstrate the utility of the proposed steganographic framework.

The rest of this paper is organized as follows. In Section 2, we present our proposed video steganographic framework. In Section 3, we present the analysis of distortion drift, which then informs the design of cost assignment method in Section 4. In Section 5, we describe our evaluation setups and findings. Finally, we conclude this paper in Section 6.

2 Proposed Steganographic Framework

Existing video steganographic frameworks, such as [23, 28], are hard to apply STCs in all transform coefficients of a whole video because they are designed for a single coding block or a single frame. Therefore, this limits their efficiency.

2.1 Limitation of Existing Steganographic Frameworks

In transform domain of video coding framework, transform coefficients in a whole video cannot provide full freedom to STCs for making modifications in one run, but multiple runs, which depends on the number of video frames or coding blocks. It is because the video coding framework limits the use of STCs. The main limitation stems from the fact that videos are encoded block by block, such as a macroblock for H.264/AVC. For each block, once it finishes coding its all coding parameters, like QDCT coefficients, motion vectors, prediction modes and so on, are written into the file of video bitstream. Based on this fact, the existing video steganographic frameworks [23, 28] are designed to allow STCs to be applied in each coding block or each frame. This limits the full use of STCs, thus further limiting the coding performance and the security of stego videos.

2.2 Proposed Video Steganographic Framework

To make full use of STCs in a whole video rather than just in a single coding block or a single frame in one run, we propose a novel video steganographic framework shown in Fig. 1.

Our proposed framework is composed of four stages labelled ①②③④ shown in Fig. 1. In stage ①, the data-hider uses a video encoder to encode one video and then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

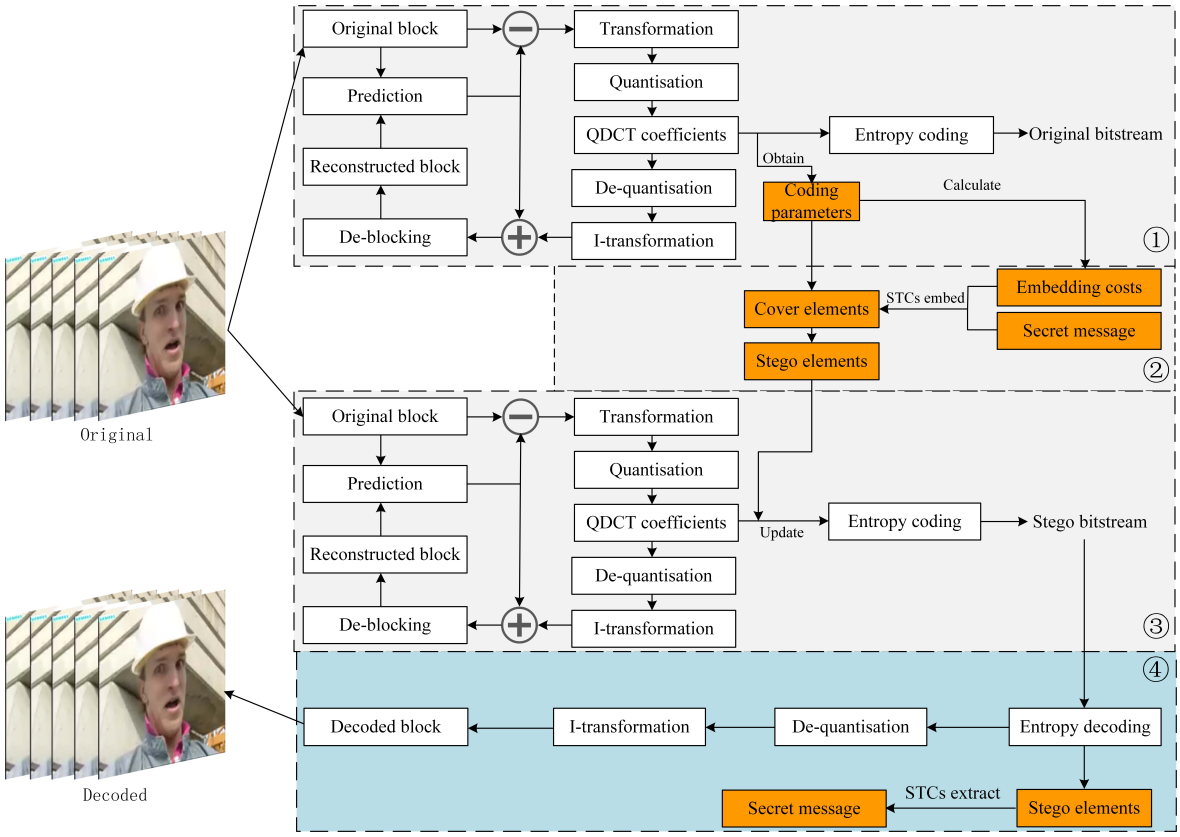


Fig. 1. New video steganographic framework based on the video coding framework.

obtains all coding parameters. The data-hider selects the parts of coding parameters, such as all nonzero transform coefficients, as cover elements and exploits the parts of coding parameters, such as motion vector, reconstructed pixels, and residuals, to calculate the embedding costs of the cover elements for video steganography. In stage ②, the data-hider makes use of the STCs embedding algorithm to combine the embedding costs to embed the secret message into the cover elements for obtaining the stego elements. In stage ③, the data-hider utilizes another video encoder, which is set to the same parameters as the video encoder used in stage ①, to encode the same video and replaces the corresponding cover elements with the stego elements in stage ②. Finally, the secret messages are embedded into the video bitstream. In stage ④, a video decoder corresponding to that of stage ① or stage ③ is used to decode the stego bitstream for obtaining the stego elements and the decoded video. By STCs extracting, the messages can be extracted from the stego elements. Noted that the proposed video steganographic framework can keep format compatibility of the video codec, which meets the encoding framework like stage ① or stage ③.

3 Analysis of Distortion Drift

Distortion drift [22, 38, 39] is the main reason of reducing the coding performance and the security of stego videos. To better design DDCA from the distortion drift point of view, we first introduce the procedure of video coding and then analyze the distortion drift in this section. Throughout this paper, matrices and sets are written in boldface. For example, a pixel block is denoted by $\boldsymbol{P}_b = (P_b(i,j))^{n_1 \times n_2}$, $P_b(i,j) \in \{0, 1, \cdots, 255\}$, $1 \le i \le n_1$, $1 \le j \le n_2$.

3.1 Procedure of Video Coding

As important parts of video compression standards, intra-frame prediction and inter-frame prediction are employed to reduce spatial redundancy and temporal redundancy in video frames, respectively. According to the 2019 Global Media Formats Report, H.264 remains the top video codec [40]. Therefore, we take H.264/AVC compression standard [34, 35] for an example in this paper and address the analysis of the distortion drift in this section. H.264/AVC provides two sizes of luminance block: 4×4 and 16×16 for the intra-frame prediction. The size 4×4 is used for complex areas and the size 16×16 is used for smooth areas in video frames. In addition, H.264/AVC provides seven sizes of luminance block: 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4 for the inter-frame prediction. Likewise, the smaller sizes are used for complex areas and the larger sizes are used for smooth areas in video frames.

According to video compression standards [34, 35, 41], on the encoding side the relationship between a pixel block $\boldsymbol{P}_b$, its predicted pixel block $\boldsymbol{P}_p$ and its corresponding residual block $\boldsymbol{R}_b$ is denoted by

$$\boldsymbol{P}_b = \boldsymbol{P}_p + \boldsymbol{R}_b \tag{1}$$

where $\boldsymbol{P}_p$ is determined by the intra-frame prediction or the inter-frame prediction exploited on the reference pixel block $\boldsymbol{P}_r$. After the prediction, the block size of 4×4 is the basic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

operation unit of the transformation and the quantization. Herein, let $\boldsymbol{P}_b$, $\boldsymbol{P}_p$, $\boldsymbol{R}_b$ and $\boldsymbol{P}_r$ be the size of 4×4. In the following, the two-dimensional integer DCT transformation is applied to $\boldsymbol{R}_b$ as follows:

$$\boldsymbol{R}_b^{DCT} = \boldsymbol{C}_f \cdot \boldsymbol{R}_b \cdot \boldsymbol{C}_f^T \tag{2}$$

where

$$\boldsymbol{C}_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

and $\boldsymbol{C}_f^T$ is the transpose matrix of $\boldsymbol{C}_f$. And then the quantization operation applied to $\boldsymbol{R}_b^{DCT}$ can be formulated as

$$\boldsymbol{R}_b^{QDCT} = \boldsymbol{R}_b^{DCT}.\times \boldsymbol{E}_f./\boldsymbol{Q}_{step}. \tag{3}$$

where

$$\boldsymbol{E}_f = \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}, a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}$$

and Q_{step} denotes the quantizer step size determined by Quantization Parameter (QP). Finally, the QDCT coefficients $\boldsymbol{R}_b^{QDCT}$ are entropy encoded to obtain the video bitstream.

On the decoding side, once the video bitstream is received it is first entropy decoded to obtain the QDCT coefficients $\boldsymbol{R}_b^{QDCT}$. In the following, the inverse quantization operation applied to $\boldsymbol{R}_b^{QDCT}$ is formulated as follows:

$$\boldsymbol{R}_{b'}^{DCT} = \boldsymbol{R}_b^{QDCT}. \times Q_{step}. \times \boldsymbol{E}_f. \times 64 \tag{4}$$

In general, the quantization operation is lossy, thus leading to $\boldsymbol{R}_{b'}^{DCT}$ different from $\boldsymbol{R}_b^{DCT}$. After that, $\boldsymbol{R}_{b'}^{DCT}$ is transformed by the inverse DCT transformation as follows:

$$\begin{aligned} \boldsymbol{R}_{b'} &= Round[(\boldsymbol{C}_i^T \cdot \boldsymbol{R}_{b'}^{DCT} \cdot \boldsymbol{C}_i)./64] \\ &= Round[\boldsymbol{C}_i^T (\boldsymbol{R}_b^{QDCT}. \times Q_{step}. \times \boldsymbol{E}_f) \cdot \boldsymbol{C}_i] \end{aligned} \tag{5}$$

where

$$\boldsymbol{C}_i^T = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

and $\boldsymbol{C}_i$ is the transpose matrix of $\boldsymbol{C}_i^T$. Therefore, the decoded (reconstructed) pixel block $\boldsymbol{P}_{b'}$ can be denoted as

$$\boldsymbol{P}_{b'} = \boldsymbol{P}_p + \boldsymbol{R}_{b'} \tag{6}$$

Clearly, $\boldsymbol{P}_{b'} \neq \boldsymbol{P}_b$. Thus, this distortion can be formulated as

$$\boldsymbol{\rho}_o = \boldsymbol{P}_b - \boldsymbol{P}_{b'} = \boldsymbol{R}_{b'} - \boldsymbol{R}_b \tag{7}$$

It should be noted that $\boldsymbol{\rho}_o$ is due to the video coding standard (the quantization operation is lossy); thus, it cannot be avoided.

3.2 Propagation of Distortion Drift

3.2.1 Inner-Block Distortion Drift

According to Equations (4-6), when changing one of QDCT coefficients of $\boldsymbol{R}_b^{QDCT}$ for steganography, this makes all pixel values corresponding to $\boldsymbol{R}_b^{QDCT}$, i.e., $\boldsymbol{P}_{b'}$, modified. Without loss of generality, let the modification be

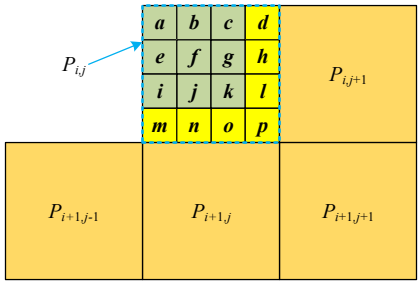


Fig. 2. $\boldsymbol{P}_{i,j}$, $\boldsymbol{P}_{i+1,j-1}$, $\boldsymbol{P}_{i+1,j}$, $\boldsymbol{P}_{i,j+1}$ and $\boldsymbol{P}_{i+1,j+1}$ are pixel blocks with the size of 4×4. The pixels d, h, l, m, n, o and p of $\boldsymbol{P}_{i,j}$ are exploited to predict and reconstruct $\boldsymbol{P}_{i,j+1}, \boldsymbol{P}_{i+1,j-1}, \boldsymbol{P}_{i+1,j}$ and $\boldsymbol{P}_{i+1,j+1}$.

$$\triangle = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and thus

$$\boldsymbol{R}_{b'}^{QDCT} = \boldsymbol{R}_b^{QDCT} + \triangle \tag{8}$$

where $\triangle(i,j) = 0(0 \leq i,j \leq 3)$ denotes making no modification on the (i^{th}, j^{th}) position of $\boldsymbol{R}_b^{QDCT}$. Therefore, the modified residual block $\boldsymbol{R}_{b''}$ with steganography (corresponding to the pixel block $\boldsymbol{P}_{i,j}$ shown in Fig. 2) is expressed as

$$\boldsymbol{R}_{b''} = Round[\boldsymbol{C}_i^T \cdot [(\boldsymbol{R}_b^{QDCT} + \triangle). \times Q_{step}. \times \boldsymbol{E}_f] \cdot \boldsymbol{C}_i] \tag{9}$$

Furthermore, the reconstructed pixel block $\boldsymbol{P}_{b''}$ with steganography can be denoted by

$$\boldsymbol{P}_{b''} = \boldsymbol{P}_p + \boldsymbol{R}_{b''} \tag{10}$$

Clearly, $\boldsymbol{P}_{b'} \neq \boldsymbol{P}_{b''}$. That is to say, the pixels of $\boldsymbol{P}_{b'}$ are changed due to the modification of steganography. More specifically, all the pixels a-p of $\boldsymbol{P}_{i,j}$ are changed (shown as Fig. 2). This is called as inner-block distortion drift, which is caused by steganography, the inverse tow-dimensional DCT transformation, and the quantization. Note that, here, the distortion of $\boldsymbol{P}_{i,j}$ is caused by the modifications of the residual $\boldsymbol{R}_{b'}$ (in Equation (6)) because of steganography.

3.2.2 Inter-Block Distortion Drift

Furthermore, when d, h, l, m, n, o, p of $\boldsymbol{P}_{i,j}$ are exploited as the reference pixels of its neighboring blocks $\boldsymbol{P}_{i-1,j-1}$, $\boldsymbol{P}_{i+1,j}$, $\boldsymbol{P}_{i,j+1}$ and $\boldsymbol{P}_{i+1,j+1}$, the distortions of d, h, l, m, n, o, p of $\boldsymbol{P}_{i,j}$ caused by steganography will spread to the blocks $\boldsymbol{P}_{i-1,j-1}$, $\boldsymbol{P}_{i+1,j}$, $\boldsymbol{P}_{i,j+1}$ and $\boldsymbol{P}_{i+1,j+1}$ by the intra-frame prediction (shown as Fig. 2). This is called as inter-block distortion drift. Herein, Fig. 2 shows an example for the intra-frame 4×4 luma block prediction modes. For the intra-frame 16×16 luma prediction modes, the same conclusion can be made. Note that, here, the change of $\boldsymbol{P}_{i,j}$ results in the inter-block distortion drift by the intra-frame prediction. For the predicted pixel blocks $\boldsymbol{P}_{i-1,j-1}$, $\boldsymbol{P}_{i+1,j}$, $\boldsymbol{P}_{i,j+1}$ and $\boldsymbol{P}_{i+1,j+1}$, Equation (6) can be rewritten as

$$\boldsymbol{P}_{b^3} = \boldsymbol{P}_{p'} + \boldsymbol{R}_{b'} \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

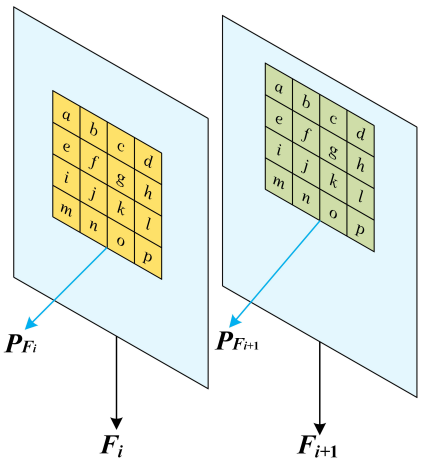


Fig. 3. The predicted frame $\boldsymbol{F}_{i+1}$ and the reference frame $\boldsymbol{F}_i$. $\boldsymbol{P}_{\boldsymbol{F}_i}$ is the reference block of the predicted block $\boldsymbol{P}_{\boldsymbol{F}_{i+1}}$ by inter-frame prediction.

Equation (11) shows that the inter-block distortion drift is caused due to the modification of $\boldsymbol{P}_p$, as the inner-block distortion drift changes the reference pixel block from $\boldsymbol{P}_p$ to $\boldsymbol{P}_{p'}$ of $\boldsymbol{P}_{i,j}$. In other words, the inner-block distortion drift induces the inter-block distortion drift.

3.2.3 *Inter-Frame Distortion Drift*

Similarly, when $\boldsymbol{P}_p$ (i.e., $\boldsymbol{P}_{i,j}$, also called as $\boldsymbol{P}_{\boldsymbol{F}_i}$ of the video frame $\boldsymbol{F}_i$ shown in Fig. 3) is exploited as the reference block of the block $\boldsymbol{P}_{\boldsymbol{F}_{i+1}}$ of the video frame $\boldsymbol{F}_{i+1}$, the distortions of the pixels a-p of $\boldsymbol{P}_{\boldsymbol{F}_i}$ induced by steganography will propagate to the pixels a-p of the predicted block $\boldsymbol{P}_{\boldsymbol{F}_{i+1}}$ because of the inter-frame prediction. This is called as inter-frame distortion drift. Fig. 3 just gives an example of the inter-frame prediction with the size of 4×4 and the same conclusion for the inter-frame prediction with other sizes can be drawn .

Equation (11) also indicates that the inter-frame distortion drift is induced by the inner-block distortion drift. In short, the distortion drift is classified into the inner-block, the inter-block and the inter-frame distortion drifts and they are caused by steganography.

4 Proposed Cost Assignment Method

The design of the proposed cost assignment is informed by the analysis presented in Section 3. Specifically, the proposed cost assignment method (DDCA) comprises the inner-block distortion cost $\eta[R_b^{QDCT}(m,n)]$, the inter-block distortion cost $\phi[R_b^{QDCT}(m,n)]$, and the inter-frame distortion cost $\varphi[R_b^{QDCT}(m,n)]$. It is formulated as:

$$\rho[R_b^{QDCT}(m,n)] = \eta[R_b^{QDCT}(m,n)] + \phi[R_b^{QDCT}(m,n)] + \varphi[R_b^{QDCT}(m,n)] \tag{12}$$

In the above equation, $R_b^{QDCT}(m,n)$ denotes a QDCT coefficient.

4.1 Inner-block Distortion Cost

As analyzed in Section 3, due to the inverse two-dimensional DCT transformation and the quantization operation, the modification on $\boldsymbol{R}_b^{QDCT}$ caused by steganography will make $\boldsymbol{R}_{b'}$ distortion, i.e., the inner-block distortion drift, (results in $\boldsymbol{R}_b$ more distortion). Therefore, the inner-block distortion cost can be defined as follows:

$$\eta^-[R_b^{QDCT}(m,n)] = \sum_{i=0}^{3}\sum_{j=0}^{3} \frac{|R^-_{b''}(i,j) - R_{b'}(i,j)|}{P_{b'}(i,j)+1} \tag{13}$$

$$\eta^+[R_b^{QDCT}(m,n)] = \sum_{i=0}^{3}\sum_{j=0}^{3} \frac{|R^+_{b''}(i,j) - R_{b'}(i,j)|}{P_{b'}(i,j)+1} \tag{14}$$

where $|\cdot|$ denotes the absolute value function and $\eta^-[R_b^{QDCT}(m,n)]$ and $\eta^+[R_b^{QDCT}(m,n)]$ respectively correspond to $R_b^{QDCT}(m,n)-1$ and $R_b^{QDCT}(m,n)+1$ ($0 \le m,n \le 3$). In addition, $R^-_{b''}(i,j)$, $R^+_{b''}(i,j)$ and $R_{b'}(i,j)$ are the residual values corresponding to the different embedding fashions, i.e., $R_b^{QDCT}(m,n)-1$, $R_b^{QDCT}(m,n)+1$ and unchanged. $P_{b'}(i,j)$ is the reconstructed pixel without steganography.

4.2 Inter-block Distortion Cost

The intra-frame prediction makes the distortion caused by steganography in the encoded blocks propagate to their adjacent blocks. For example, as shown in Fig. 2, steganography on QDCT coefficients in $\boldsymbol{P}_{i,j}$ changes the pixels a-p, thus distorting the pixels of $\boldsymbol{P}_{i,j+1}$, $\boldsymbol{P}_{i+1,j-1}$, $\boldsymbol{P}_{i+1,j}$ and $\boldsymbol{P}_{i+1,j+1}$ by the intra-frame prediction. Actually, this is because steganography changes the part $\boldsymbol{R}_{b'}$ of Equation (6). Thereby, the inter-block distortion cost function $\phi[R_b^{QDCT}(m,n)]$ can be defined as

$$\phi^-[R_b^{QDCT}(m,n)] = \sum_{k\in\boldsymbol{V}}\sum_{h\in\boldsymbol{U}} \frac{|R^-_{b''}(k) - R_{b'}(k)|}{P_{b'}(h)+1} \tag{15}$$

$$\phi^+[R_b^{QDCT}(m,n)] = \sum_{k\in\boldsymbol{V}}\sum_{h\in\boldsymbol{U}} \frac{|R^+_{b''}(k) - R_{b'}(k)|}{P_{b'}(h)+1} \tag{16}$$

where $\boldsymbol{V}$ and $\boldsymbol{U}$ denote the pixel sets of the reference pixels and the predicted pixels, respectively. $P_{b'}(h)$ represents the reconstructed pixel without steganography. Without loss of generality, Fig. 4 gives an example when $\boldsymbol{P}_{i,j+1}$ is predicted by $\boldsymbol{P}_{i,j}$ exploiting the intra-frame 4×4 prediction mode 1, i.e., horizontal prediction. For this case, $\boldsymbol{V} = \{(0,3),(1,3),(2,3),(3,3)\}$ in $\boldsymbol{P}_{i,j}$ and $\boldsymbol{U} = \{(m,n)|0 \le m,n \le 3\}$ in $\boldsymbol{P}_{i,j+1}$. Furthermore, when $k = (m,3)(0 \le m \le 3)$, $h \in \{(m,n)|0 \le n \le 3\}$. For instance, when

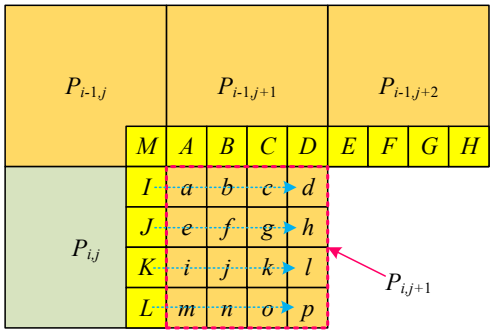


Fig. 4. I, J, K and L (respectively corresponding to d, h, l and p in Fig. 2) of $\boldsymbol{P}_{i,j}$ are exploited as the reference pixels of $\boldsymbol{P}_{i,j+1}$ by the intra-frame 4×4 prediction mode 1, i.e., horizontal prediction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

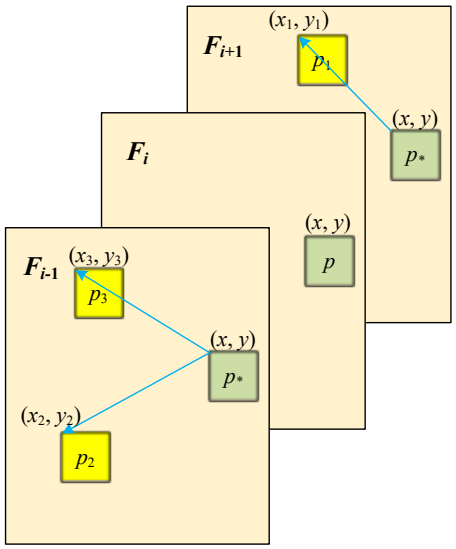


Fig. 5. Inter-frame prediction. p denotes the reference pixel at the position (x, y) in the frame $\boldsymbol{F}_i$. p_j $(1 \leq j \leq 3)$ denotes the predicted pixel at the position (x_j, y_j) and $p*$ has the same position with p in the frame $\boldsymbol{F}_{i-1}$ or $\boldsymbol{F}_{i+1}$

$k = (0, 3)$, $h \in \{(0, n)|0 \leq n \leq 3\}$ and $P_{b'}(h) \in \{a, b, c, d\}$ shown in Fig. 4. Note that, this example just gives the case of horizontal prediction. In fact, $\boldsymbol{V}$ must also contain the positions of m, n, o and p in $\boldsymbol{P}_{i,j}$ (shown as Fig. 2) when m, n, o and p are used to predict the blocks $\boldsymbol{P}_{i+1,j-1}, \boldsymbol{P}_{i+1,j}$ and $\boldsymbol{P}_{i+1,j+1}$. This depends on whether these pixels are as reference pixels or not. Correspondingly, $\boldsymbol{U}$ should contain more positions corresponding to $\boldsymbol{V}$. For other intra-frame 4×4 prediction modes, the same way can be used to calculate the inter-block distortion costs. Likewise, the inter-block distortion cost exploiting the intra-frame 16×16 prediction also can be calculated in the same way. When the pixels of the current block are not exploited as the reference pixels for the intra-frame prediction, $\phi^-[R_b^{QDCT}(m, n)] = \phi^+[R_b^{QDCT}(m, n)] = 0$.

4.3 Inter-frame Distortion Cost

The inter-frame prediction makes the distortion caused by steganography in the reference frames propagate to their predicted frames. For example, $\boldsymbol{F_i}$ is exploited as a reference frame for $\boldsymbol{F}_{i-1}$ and $\boldsymbol{F}_{i+1}$ (shown as Fig. 5). In Fig. 5, the pixels p_1, p_2 and p_3 are predicted by the pixel p. As analyzed in Section 3, because of the modification of QDCT coefficient(s) of a block, the distortion caused by steganography on p propagates to p_1, p_2 and p_3. This distortion in essence is induced by its residual when reconstructing p. Therefore, the inter-frame distortion cost can be defined as follows:

$$\varphi^-[R_b^{QDCT}(m, n)] = \sum_{h=1}^{H} \frac{|R^-_{b''}(m, n) - R_{b'}(m, n)|}{P_{b'}(h) + 1} \cdot g[P_{b'}(h)] \tag{17}$$

$$\varphi^+[R_b^{QDCT}(m, n)] = \sum_{h=1}^{H} \frac{|R^+_{b''}(m, n) - R_{b'}(m, n)|}{P_{b'}(h) + 1} \cdot g[P_{b'}(h)] \tag{18}$$

where H denotes the number of the pixels predicted by the pixel p corresponding to $R_b^{QDCT}(m, n)$. For the example of Fig. 5, $H = 3$ and $P_{b'}(h) \in \{p_1, p_2, p_3\}$. In addition, the function $g(p)$ is defined as

$$g(p) = \begin{cases} 1 & p \text{ satisfies BP} \\ 0.5 & p \text{ satisfies SP} \end{cases} \tag{19}$$

where BP denotes the pixel p is predicted by the Bi-directional prediction, i.e., both the forward prediction and the backward prediction used at the same time. Correspondingly, SP denotes p is predicted just by either the forward prediction or the backward prediction. In Fig. 5, suppose the display order is F_{i-1}, F_i and F_{i+1} and the encoding order is F_i, F_{i-1} and F_{i+1}, therefore p_2 and p_3 are predicted by the backward prediction but p_1 is predicted by the forward prediction. When p is not exploited as a reference pixel for the inter-frame prediction, $\varphi^-[R_b^{QDCT}(m, n)] = \varphi^+[R_b^{QDCT}(m, n)] = 0$.

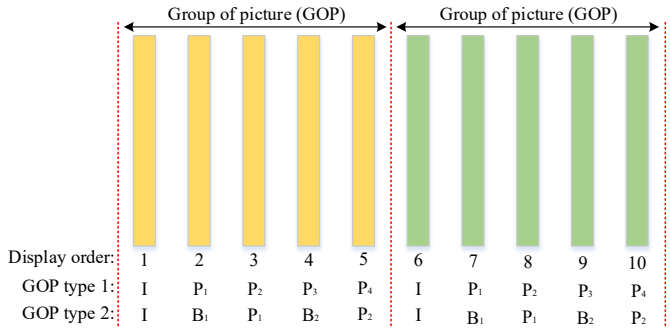


Fig. 6. An example to show the display order and GOP types of a video sequence containing two GOPs.

4.4 Final Distortion Cost

As discussed in Sections 4.1 to 4.3, the inner-block distortion, the inter-block distortion, and the intra-frame distortion collectively determine the final distortion cost. The inner-block distortion causes the inter-block distortion and the inter-frame distortion by the intra-frame prediction and the inter-frame prediction, respectively. Moreover, the display order of video frames also plays an important role in the final distortion cost. Fig. 6 is given to explain this.

According to video compression standards, such as H.264/AVC [34, 35] and H.265/HEVC [41], the encoded frames are used as reference frames for the to-be-encoded frames in the same GOP. Without loss of generality, the first one GOP shown in Fig. 6 is exploited to address and its display order is from 1 to 5. When its GOP type is IPPPP, I is the reference frame of P_1 and then I and P_1 can be used as the reference frames of P_2. Similarly, P_3 can be predicted according to I, P_1 and P_2 and then P_4 can be predicted by I, P_1, P_2 and P_3. When its GOP type is IBPBP, I is used as the reference frame of P_1 and then I and P_1 can be leveraged as the reference frames of B_1. In the following, P_2 can be predicted by I, B_1 and P_1. Finally, B_2 can be predicted by I, B_1, P_1 and P_2. Based on these analyses, assume that the video frame which is first displayed has more serious distortion caused by steganography. Therefore, the final distortion cost can be formulated by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

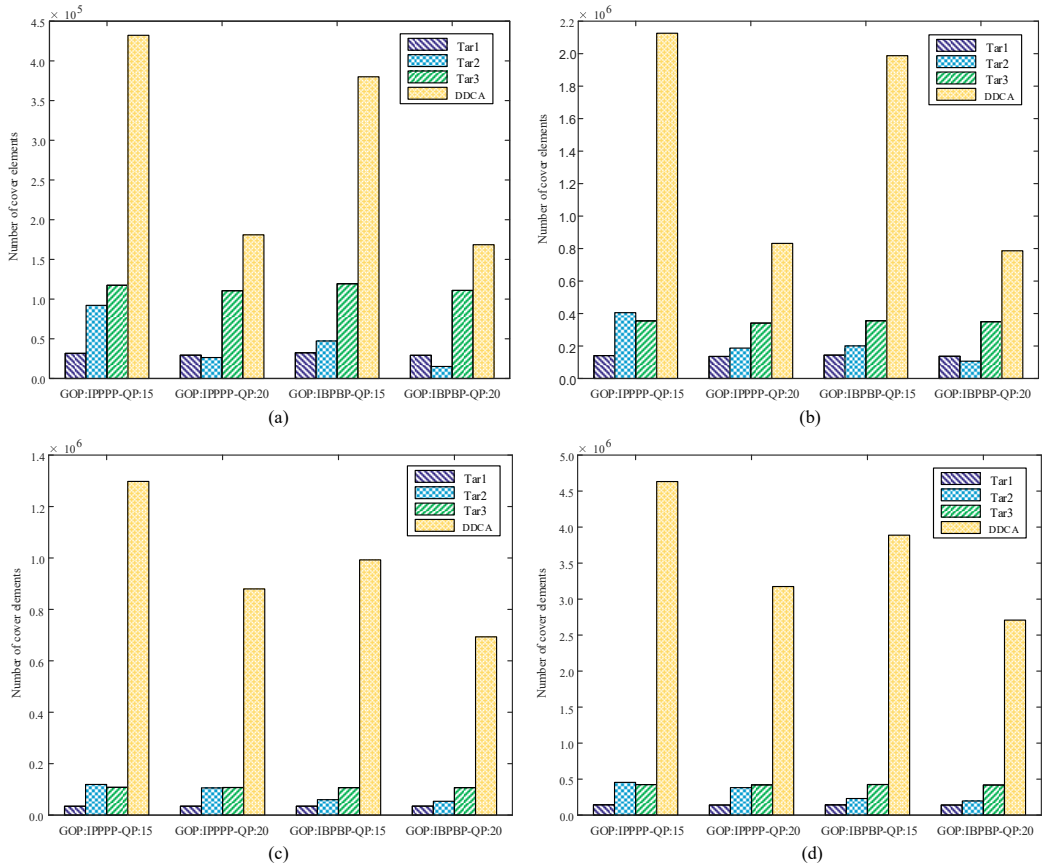


Fig. 7. Comparisons of the number of cover elements using Tar1 [23], Tar2 [26], Tar3 [27] and DDCA on video sequences Foreman and Mobile. (a) Foreman (resolution: 176×144). (b) Mobile (resolution: 176×144). (c) Foreman (resolution: 352×288). (d) Mobile (resolution: 352×288).

TABLE 1
Fixed embedding payload (bits)
for the four steganographic methods on
databases DB1 and DB2 with different GOPs and QPs.

Database	GOP	QP	Embedding-Payload (bits)
DB1	IPPPP	15	6000
		20	4000
	IBPBP	15	6000
		20	4000
DB2	IPPPP	15	18000
		20	16000
	IBPBP	15	18000
		20	16000

$$\rho^{-}[R_b^{QDCT}(m,n)] = \alpha\eta^{-}[R_b^{QDCT}(m,n)]\psi(d)+ \beta\phi^{-}[R_b^{QDCT}(m,n)] + \gamma\varphi^{-}[R_b^{QDCT}(m,n)] \quad (20)$$

$$\rho^{+}[R_b^{QDCT}(m,n)] = \alpha\eta^{+}[R_b^{QDCT}(m,n)]\psi(d)+ \beta\phi^{+}[R_b^{QDCT}(m,n)] + \gamma\varphi^{+}[R_b^{QDCT}(m,n)] \quad (21)$$

where α, β and γ are three scaling factors and $\alpha+\beta+\gamma = 1$. In addition, the function $\psi(d)$ is defined as

$$\psi(d) = \frac{1}{mod(d, period) + 1} \quad (22)$$

In the above equation, d is the display order of the current frame and $period$ is the intra-period. For DDCA, the nonzero coefficients (e.g. ± 1) cannot be changed to 0, thus the costs of these coefficients changed to 0 are $+\infty$.

5 Experimental Results and Analysis

In this section, we will describe our experimental setup, the scaling factors, and the number of cover elements. We will also present a comparative summary of the performance of DDCA and those of three other state-of-the-art steganographic schemes [23, 26, 27].

5.1 Experimental Setup

5.1.1 Video Databases

Two video databases are collected from the Internet, and each video contains 100 frames. The first video database (DB1) consists of 70 standard test video sequences with YUV 4:2:0 color space and QCIF resolution (176×144). The other video database (DB2) consists of 30 standard test video sequences with YUV 4:2:0 color space and CIF resolution (352×288). All video sequences in the two databases are stored without being compressed.

5.1.2 Coding Performance

Based on our proposed framework (shown in Fig. 1), DDCA is implemented on a well-known H.264/AVC codec named Joint Model (JM) 19.0 [42]. To implement our proposed method, the double-layered STCs [2] with the constraint

TABLE 2
Average PSNRs (dB) of DDCA with different embedding rates on DB1 and DB2.

Dataset	GOP	QP	Original	Embedding Rate (bpnzAC)							
				0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
DB1	IPPPP	15	47.4786	47.1040	46.5670	46.0027	45.4110	44.6789	43.9332	43.1414	42.2488
		20	43.1416	42.8244	42.3732	42.7869	41.1210	40.3305	39.2297	38.1422	37.0562
	IBPBP	15	47.4874	47.1841	46.6924	46.1313	45.5505	44.8630	44.1438	43.3140	42.5090
		20	43.1927	42.9000	42.4413	41.8390	41.1490	40.3597	39.4536	38.3704	37.5893
DB2	IPPPP	15	47.8861	47.3490	46.8481	46.3574	45.8489	45.3348	44.7815	44.1827	43.3978
		20	42.9272	42.7305	42.4893	42.2046	41.8531	41.4097	40.9313	40.3109	39.6199
	IBPBP	15	47.8620	47.3421	46.8507	46.3987	45.9137	45.4319	44.8510	44.1610	43.4556
		20	42.6841	42.4958	42.2589	41.9799	41.6192	41.2145	40.6816	39.9495	39.2325

TABLE 3
Average SSIMs of DDCA with different embedding rates on DB1 and DB2.

Database	GOP	QP	Original	Embedding Rate (bpnzAC)							
				0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
DB1	IPPPP	15	0.9931	0.9925	0.9920	0.9914	0.9908	0.9899	0.9890	0.9879	0.9867
		20	0.9832	0.9823	0.9818	0.9810	0.9800	0.9786	0.9766	0.9743	0.9742
	IBPBP	15	0.9930	0.9926	0.9921	0.9916	0.9910	0.9904	0.9895	0.9885	0.9874
		20	0.9833	0.9823	0.9818	0.9810	0.9799	0.9786	0.9768	0.9745	0.9723
DB2	IPPPP	15	0.9928	0.9919	0.9911	0.9904	0.9896	0.9886	0.9876	0.9865	0.9851
		20	0.9782	0.9770	0.9762	0.9754	0.9744	0.9733	0.9719	0.9703	0.9683
	IBPBP	15	0.9927	0.9920	0.9914	0.9907	0.9900	0.9892	0.9883	0.9872	0.9861
		20	0.9768	0.9757	0.9750	0.9743	0.9735	0.9725	0.9712	0.9696	0.9676

height h set to 10 is utilized, and its embedding payloads (bits) [or embedding rate (bits embedded per non-zero AC coefficients, bpnzAC)] will be discussed in Sections 5.3 to 5.5. In addition, we set (α,β,γ)=(0.50,0.15,0.35) in Sections 5.3-5.5 and discuss it in Section 5.6. All video sequences are compressed by H.264/AVC JM 19.0 with an intra-period of 5 and Group of Picture (GOP): IPPPP and IBPBP. In addition, two different Quantization Parameters (QPs) 15 and 20 are considered at the encoder side for the two databases. In order to evaluate the coding performance, visual quality, including Peak-Signal-to-Noise-Ratio (PSNR) and Structural SIMilarity index (SSIM) [35, 36, 43, 44], and bit-rate are used. Moreover, to better measure the impact on coding performance by DDCA, Bit-rate Increase Rate (BIR) [24] is defined by

$$BIR = \frac{BitRate_{Stego} - BitRate_{Original}}{BitRate_{Original}} \times 100\% \quad (23)$$

where $BitRate_{Original}$ and $BitRate_{Stego}$ are respectively generated by H.264/AVC JM 19.0 without and with a steganographic method.

5.1.3 Security

The state-of-the-art feature set (denoted as Tar4) designed by Wang et al. [29] combined with ensemble classifier [45] is used to measure the security performance of DDCA against steganalysis. The security performance is qualified by detection accuracy [30], which is the mean value of the true positive rate and the true negative rate, and the final detection accuracy is averaged over 20 iterations with different splits of each database. In the procedure of detecting, the cover and stego pairs of the video sequences in each database are randomly split into two halves, one half is for training and the rest half is for testing. What is more, to better evaluate the security, we compare DDCA with the state-of-the-art steganography methods, which contain Cao et al.'s scheme [23] (denoted as Tar1), Chen et al.'s scheme [26] (denoted as Tar2) and Xue et al.'s scheme [27] (denoted as Tar3).

5.2 Number of Cover Elements

Currently, the existing state-of-the-art steganographic methods only exploit a small portion of QDCT coefficients, including zero and nonzero QDCT coefficients. However, when compared with DDCA, these state-of-the-art steganographic methods [23, 26, 27] have much less QDCT coefficients used for steganography. More specifically, Tar1 [23] and Tar3 [27] only exploit a small portion of QDCT coefficients in I frames and in contrast Tar2 [26] only utilizes that in P frames. For DDCA, all nonzero QDCT AC coefficients, which belong to I, P and B frames, are used. To better compare the number of cover elements used for steganography, two video sequences, i.e., Foreman and Mobile, are used for experiments and each one has two kinds of different resolutions, including 176×144 and 352×288. The numbers of video cover elements of Tar1, Tar3, Tar2 and DDCA on Foreman and Mobile are shown in Fig. 7.

As shown in Fig. 7, for DDCA video texture impacts the number of cover elements, i.e. a video sequence with more complex texture will have a larger number of cover

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

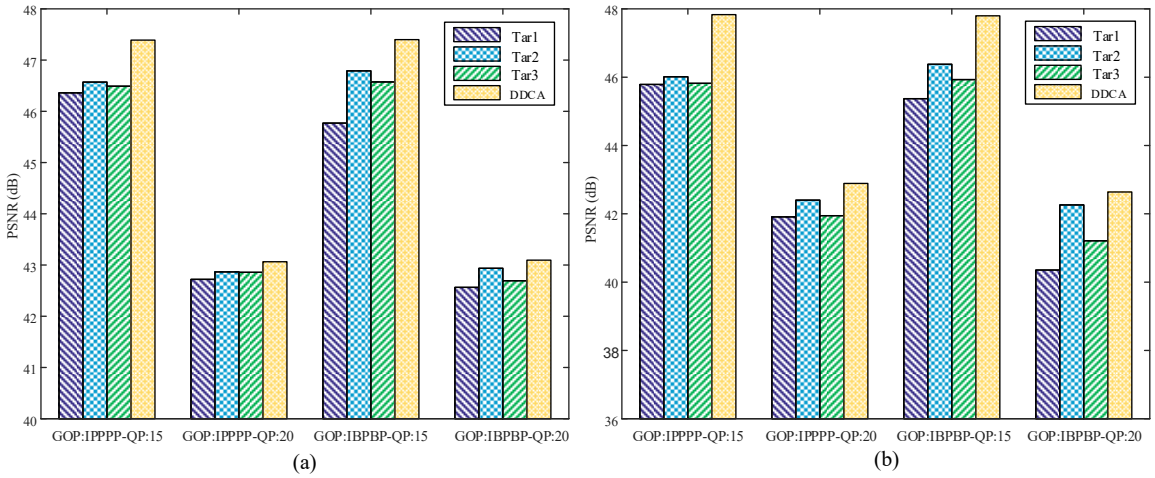


Fig. 8. Comparisons of average PSNRs of Tar1 [23], Tar2 [26], Tar3 [27] and DDCA on databases DB1 and DB2. (a) DB1. (b) DB2.

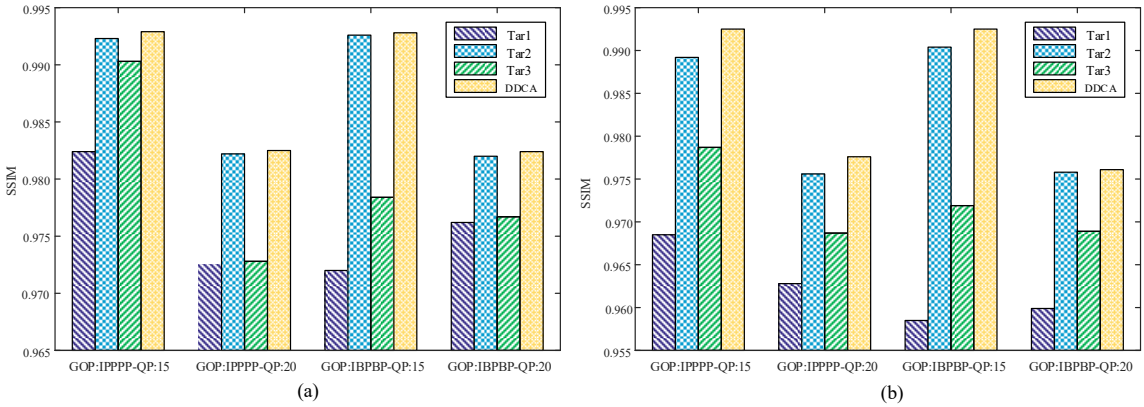


Fig. 9. Comparisons of average SSIMs of Tar1 [23], Tar2 [26], Tar3 [27] and DDCA on databases DB1 and DB2. (a) DB1. (b) DB2.

elements. For instance, DDCA has larger number of cover elements on Mobile than Foreman, which respectively correspond to Figs. 7(a) and 7(b). Comparing Figs. 7(c) with 7(d), the same conclusion can be drawn on Foreman and Mobile. When compared with Tar1, Tar2 and Tar3, obviously DDCA has more cover elements for steganography. Fig. 7(a) is used as an example for addressing this. When setting GOP:IPPPP-GP:15, DDCA has around 4.4×10^5 nonzero QDCT AC coefficients (shown in Fig. 7(a)). Under the same setting, Tar3 has more QDCT coefficients, around 1.2×10^5, than Tar1 and Tar2. Therefore, DDCA indeed has more cover elements than the three methods [23, 26, 27]. Figs. 7(b)-(d) can also be exploited to explain this conclusion.

5.3 Impact of Visual Quality

QDCT coefficient-based steganography modifies the coding parameters during video compression, thus further affecting the video coding quality (coding performance). The video coding quality can be reflected by the visual quality. In this paper, PSNR and SSIM are used to reflect the visual quality of stego videos and they are calculated by comparing the uncompressed video sequence and the decoded reconstructed video sequence before or after steganography. The visual quality for the databases DB1 and DB2 using DDCA with different embedding rates 0.05-0.40 bpnzAC is listed in Tables 2 and 3.

As observed from Tables 2 and 3, steganography indeed affects the visual quality and leads to its degradations when compared to the original PSNRs and SSIMs. With the increase of embedding rate setting at the same GOP and QP, PSNRs and SSIMs are decreasing. This is because that setting a larger embedding rate induces more modifications of nonzero QDCT coefficients. Furthermore, the more modification makes the visual quality more significantly degraded. For instance, when setting GOP:IPPPP-QP:15 on database DB1, the embedding rate increases from 0.05 to 0.40 bpnzAC and the corresponding PSNR decreases from 47.1040 to 42.2488 dB at the same time (shown as Table 2). Similarly, the same conclusion for SSIM can be drawn on database DB1 with the same settings (shown in Table 3). In fact, the same conclusion for PSNR and SSIM changes can still be drawn on databases DB1 and DB2 under different GOPs and QPs (shown in Tables 2 and 3).

Based on the analysis in Section 5.2, Tar1, Tar2, Tar3 and DDCA have different numbers of cover elements. Therefore, setting the same embedding rate for Tar1, Tar2, Tar3 and DDCA, different embedding payloads will be embedded

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

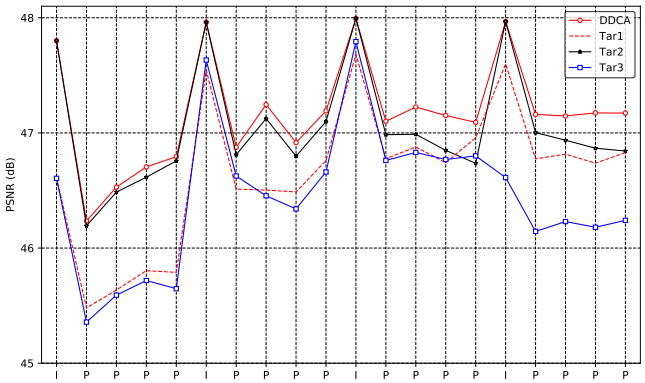


Fig. 10. The PSNR comparison of the first four GOPs of Foreman.

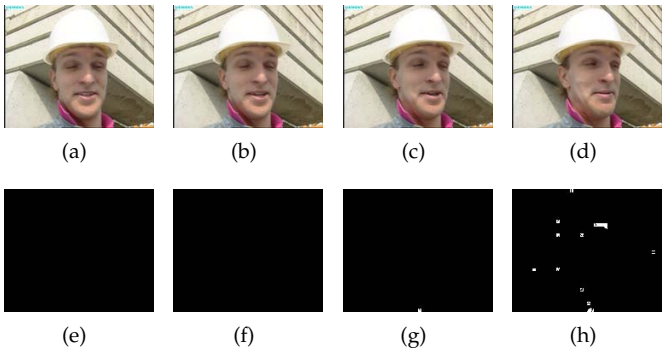


Fig. 11. The first four stego frames and their difference frames on Foreman, where (a)-(d) are the stego frames generated by DDCA and correspond to the first four frames of Fig. 10, and (e)-(h) are the difference frames generated by comparing (a)-(d) and their corresponding original compressed frames.

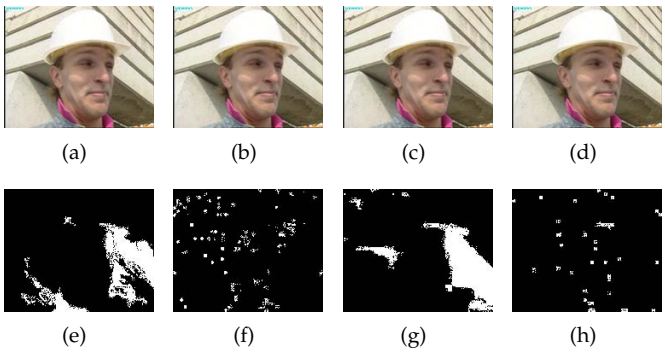


Fig. 12. The fifth stego frames and their difference frames on Foreman with Tar1 [23], Tar2 [26], Tar3 [27], and DDCA (corresponding to the fifth frames of Fig. 10). (a) Tar1. (b) Tar2. (c) Tar3. (d) DDCA. (e) Tar1. (f) Tar2. (g) Tar3. (h) DDCA.

into videos sequences. It is not fair to compare the visual quality of Tar1, Tar2, Tar3 and DDCA under this setting. To fairly compare the PSNR and SSIM of Tar1, Tar2, Tar3 and DDCA, the fixed embedding payloads, i.e., 4000, 6000, 16000 and 18000 bits, are set and shown as Table 1.

By setting different embedding payloads and same GOPs and QPs on DB1 and DB2 (shown as Table 1), Figs. 8 and 9 are obtained to compare the visual quality of Tar1, Tar2, Tar3 and DDCA. Fig. 8 shows the average PSNRs of Tar1, Tar2, Tar3 and DDCA and Fig. 9 shows the average SSIMs of that. As shown in Fig. 8, DDCA has the best PSNRs when compared with Tar1, Tar2 and Tar3 under the same setting. In addition, although Tar2 and Tar3 have very close PSNRs when setting GOP:IPPPP-QP:20 on DB1 (shown as Fig. 8(a)), on the whole Tar2 has better PSNRs when compared with Tar1 and Tar3. Obviously, Tar1 has the most significant degradation of PSNR according to Fig. 8(a). The same conclusion can be drawn from Fig. 8(b). Analogously, the same conclusion in terms of SSIM can be drawn from Fig. 9. In summary, DDCA indeed has advantages in PSNR and SSIM compared with Tar1, Tar2 and Tar3.

For DDCA, all nonzero QDCT AC coefficients, which belong to I, P and B frames, are fully used for minimizing the impact of distortion drift to steganography. That is to say, after each coefficient is assigned a distortion cost, STCs can select coefficients with less distortion impacts to change for steganography when given the embedding payload. Thereby, DDCA obtains the best visual quality in terms of PSNR and SSIM when compared to Tar1, Tar2 and Tar3. For Tar1, only the macroblocks with intra-frame 4×4 prediction modes in I frames are considered for steganography. Actually, only parts of 4×4 blocks of these macroblocks are embedded into messages and the others are normally encoded. Likewise, for Tar3 only the macroblocks with intra-frame 4×4 prediction modes in I frames are exploited to combine distortion compensation and texture complexity for optimizing distortion. However, Tar1 and Tar3 do not exploit the same type macroblocks in P and B frames. As analyzed in Section 4.4, more frames are predicted by I frames but less frames by P or B frames. Therefore, only utilizing the macroblocks of I frames leads to more distortion. For Tar2, the QDCT coefficients in high frequency areas are exploited and combined with STCs. Therefore, Tar2 outperforms Tar1 and Tar3 in terms of PSNR and SSIM. However, Tar2 still does not make use of all coefficients, thus resulting in less PSNRs and SSIMs compared with DDCA.

In order to more specifically compare the change of visual quality of Tar1, Tar2, Tar3, and DDCA, we conduct experiments on Foreman by setting GOP:IPPPP-QP:15 and the embedding payload as 6000 bits. Without loss of generality, we draw Fig. 10 to illustrate the PSNR changes of the first four GOPs on Foreman. One can observe from Fig. 10 that DDCA has the same PSNR values with Tar2 and larger PSNR values than Tar1 and Tar3 for I frames. For each P frame, DDCA has a larger PSNR value than Tar1, Tar2 and Tar3. Moreover, Figs. 11 to 12 show the distortion drift due to DDCA and compare the propagated distortions (due to Tar1, Tar2, Tar3 and DDCA) in the last stego frame of the first GOP on Foreman. There is no obvious distortion observed from Figs. 11(a)-(d) and Figs. 12(a)-(d). Figs.11(e)-(f) show their corresponding stego frames, i.e., Figs. 11(a)-(b), have no difference with the original compressed frames. In other words, the cover elements in the two frames are not modified for steganography. In addition, by observing Figs.11(g)-(h) and Fig. 12(h) the distortion caused by DDCA in Fig. 11(c) is propagated to Fig. 11(d) and Fig. 12(d), and the distortion caused by DDCA in Fig. 11(d) is spread to Fig. 12(d). Therefore, the last stego frame in the GOP contains not only distortions due to the modifications for

TABLE 4
Levels and corresponding codewords of Level codeword.

suffixLength	Level	Codeword	Level	Codeword
0	1	1	-1	01
	2	001	-2	0001
	3	00001	-3	000001
	4	0000001	-4	00000001

steganography but also the distortions propagated from the reference frame. When compared to Fig. 12(f), Fig. 12(h) has less difference (white) areas with the original compressed frame. In other words, DDCA results in the last P frame of the first GOP on Foreman with less distortion than Tar2. For Fig. 12(e) and Fig. 12(g), obviously they have much more difference areas with the original compressed than Fig. 12(f) and Fig. 12(h). That is to say, Tar1 and Tar3 make the last P frame of the first GOP on Foreman with more significant distortion. However, actually Tar1 and Tar3 do not exploit P frames for steganography. Thus, the distortion of the last P frames is caused because of the stego modification of I frame. This way, the short analysis demonstrates that designing cost assignment methods from the distortion drift point of view for video steganography is reasonable.

5.4 Impact of Bit-Rate

The video coding quality can be also reflected by the bit-rate increase. Generally speaking, increasing bit-rate will obtain better visual quality according to Rate Distortion Optimization (RDO) in video coding standards [35, 36]. That is, sacrificing the cost of bit-rate increase is for better visual quality. However, it has essential differences from the bit-rate increase caused by steganography. Steganography-based bit-rate increase is because that the modifications of nonzero QDCT coefficients makes entropy coding, including Variable-Length Codes (VLCs) and Context-Adaptive Binary Arithmetic Coding (CABAC) [35, 36], need more bits to express and code. In our experiments Context-Adaptive Variable Length Coding (CAVLC) and Exp-Golom codes [35], which belong to VLCs, are used, therefore the reason of the bit-rate increase induced by steganography in VLCs is addressed in the following. For entropy coding CABAC, the reasons of the bit-rate increase caused by steganography will be addressed and presented in our future work.

VLCs: Nonzero QDCT coefficients can be classified into three types, i.e., the coefficients with the absolute values bigger than 2, equal to 2 and 1. For the first with the absolute value bigger than 2, assume the modification probability of adding or subtracting 1 on nonzero QDCT coefficients is equal (In the following analysis, we also assume the changed positions of the mentioned coefficients in a 4×4 block for the three types are same). Therefore, the average number of bits needed to code is unchanged. In H.264/AVC, each CAVLC codeword [35] can be expressed as:

$$\{Coeff\_token, Sign\_of\_TrailingOnes, Level, Total\_zeros, Run\_before\}$$

For nonzero QDCT coefficient-based steganography, only the $Level$ during encoding may be changed. Furthermore, each $Level$ codeword consists of a prefix ($level\_prefix$) and a suffix ($level\_suffix$) as

$$Level\ codeword = [level\_prefix], [level\_suffix]$$

Table 4 [35, 46] is used for taking an example to explain this. Table 4 shows $Levels$ with $suffixLength = 0$ and the corresponding $codewords$. Without loss of generality, for a nonzero QDCT coefficient valued -3, it is modified to -2 or -4 by adding or subtracting 1 and their corresponding codewords are "0001" and "00000001". Therefore, the average number of bits needed to code is unchanged when the changing probability is equal, i.e., $\frac{1}{2}$. For the second with the absolute value equal to 2, if its absolute value is changed to 3, the changed $Level$ will need two more bits to code compared with the original $Level$. When its absolute value is changed to 1, if the changed coefficient is a trailing coefficient (one can refer to [34, 35] for more details of the trailing coefficient), it needs only one bit to code. Otherwise, it needs two less bits to code. For the third with the absolute value equal to 1, the coefficients of this type are not allowed to change to 0 but ± 2. If the original coefficient is a trailing coefficient, the coefficient changed to ± 2 needs not only two more bits to code the $Level$ but also several bits to code the Run, which depends the length of Run. Otherwise, the changed coefficient needs two more bits to code. During video encoding, there exist more coefficients of the third than the second. When considering the second and the third at the same time, their modifications results in the bit-rate increase. Table 5 is given to evaluate the impact of video encoding in terms of BIR.

As observed from Table 5, for a fixed GOP and QP, when the embedding rates increase, BIRs also increase. As the above-mentioned, setting a larger embedding rate causes more modifications of nonzero QDCT coefficients. Combined with the reason of the bit-rate increase, therefore, setting a larger embedding rate leads to larger BIRs. For instance, when setting GOP:IPPPP-QP:15, DDCA obtains BIRs from 0.1769% to 2.1366% corresponding to the embedding rates from 0.05 to 0.40 bpnzAC on database DB1 (shown as Table 5). Similarly, the same conclusion for BIR variation can be drawn on databases DB1 and DB2 under different GOPs and QPs.

To further evaluate DDCA and fairly compare DDCA with Tar1, Tar2, Tar3, Fig. 13 is obtained under the settings as Table 1. Fig. 13 shows the average BIRs of DDCA, Tar1, Tar2 and Tar3. Lower BIRs for a cost assignment method means it has a better coding performance in terms of the bit-rate than others. As shown in Fig. 13, it is obvious that DDCA has the smallest BIRs under different settings. In contrast, Tar2 induces the largest BIRs. Moreover, Tar1 causes larger BIRs than Tar3 and DDCA. Although DDCA and Tar3 have a very close BIRs under the same settings, DDCA leads to less BIRs (shown as Fig. 13). In short, DDCA results in the least BIRs compared with Tar1, Tar2 and Tar3.

For DDCA, the analysis of BIR variation is addressed above. For Tar1, Tar2 and Tar3, zero and nonzero QDCT coefficients are selected from particular coding blocks and used for steganography. Besides, there are far more zero QDCT coefficients than nonzero QDCT coefficients used for steganography. For nonzero QDCT coefficients with the absolute value bigger than 1, the analysis of the bit-rate variation is similar to that in DDCA. For nonzero QDCT coefficients with the absolute value equal to 1, the absolute

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

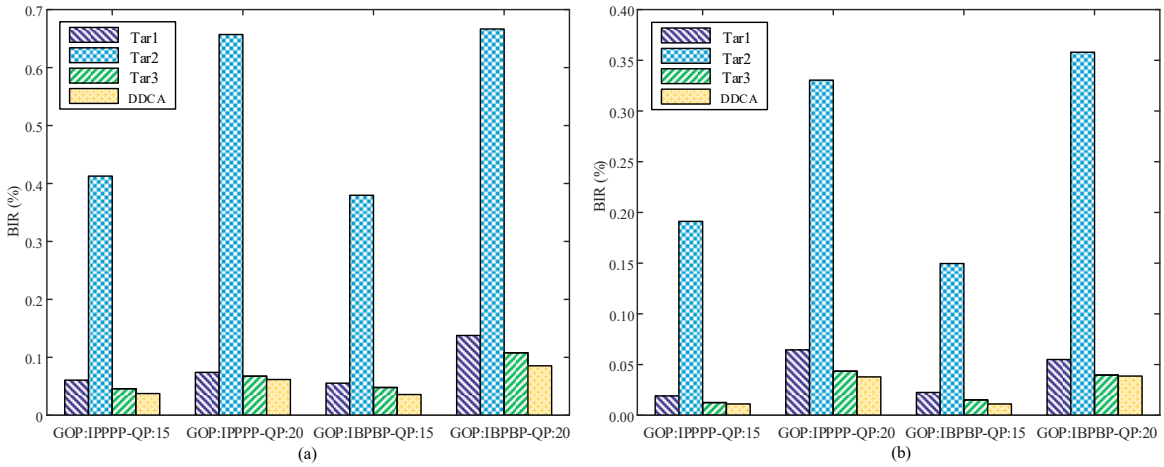


Fig. 13. Comparisons of average BIRs (%) of Tar1 [23], Tar2 [26], Tar3 [27] and DDCA on databases DB1 and DB2. (a) DB1. (b) DB2.

TABLE 5
Average BIRs (%) of DDCA with different embedding rates on DB1 and DB2.

Database	GOP	QP	Embedding Rate (bpnzAC)							
			0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
DB1	IPPPP	15	0.1769	0.4039	0.6561	0.9253	1.2113	1.5139	1.8225	2.1366
		20	0.1938	0.4717	0.7808	1.1004	1.4171	1.6279	1.8768	2.1327
	IBPBP	15	0.1419	0.3278	0.5351	0.7613	1.0011	1.2546	1.5185	1.7757
		20	0.2029	0.4607	0.7406	1.0254	1.3040	1.5668	1.7964	2.0112
DB2	IPPPP	15	0.1305	0.3019	0.4972	0.7128	0.9445	1.1912	1.4621	1.7444
		20	0.1963	0.4559	0.7524	1.0804	1.4354	1.8174	2.2180	2.6280
	IBPBP	15	0.1181	0.2761	0.4569	0.6579	0.8775	1.1139	1.3664	1.6314
		20	0.1691	0.3980	0.6633	0.9568	1.2768	1.6214	1.9771	2.3390

values changed to 2 leads to the same bit-rate variation like that in DDCA. When the absolute value is changed to 0, if the original coefficient is a trailing coefficient, the change reduces one bit needed to code. Otherwise (e.g., the value of the original coefficient is -1), the change decreases by not only two bits for the $Level$ (shown as Table 4) but also several bits for the Run, which is also dependent on the length of Run. Likewise, if the value of the original coefficient is 1, the change decreases by not only one bit for the $Level$ but also several bits for the Run. Consequently, the modifications on these coefficients result in a slight decrease of the bit-rate. For zero QDCT coefficients, they can be divided into two parts. Assume that the first part is used to eliminate the above-mentioned bit-rate decrease. The second part has far more zero QDCT coefficients than the first part. Zero QDCT coefficients are changed to ± 1, if the changed coefficient is a tailing coefficient, it need one bit to code. Otherwise, the change increases $Run - Level$ pairs which need more bits to code. Furthermore, too many zero QDCT coefficients used for steganography will induce a dramatic increase in the bit-rate. In practice, both zero and nonzero QDCT coefficients may cause the bit-rate increase at the same time that depends on the cost assignment methods designed by researchers. In coding blocks, high frequency area has more zero QDCT coefficients than low and middle frequency areas and middle frequency area has that than low frequency area. All QDCT coefficients in low, middle and high frequency areas of selected blocks by Tar1 and Tar3 are used for steganogragpy but only parts of QDCT coefficients in high frequency areas of selected blocks by Tar2 for steganography. That is to say, there are more zero QDCT coefficients used for steganography in Tar2 than Tar1 and Tar3. Therefore, Tar2 leads to the most significant BIR increases and Tar1 and Tar3 have close BIRs. In DDCA, only nonzero QDCT coefficients are exploited for steganography, thereby DDCA has lower BIRs compared with Tar1, Tar2 and Tar3.

5.5 Security against Steganalysis

Steganalysis is the technique to measure the security of steganography. Therefore, we exploit the state-of-the-art video steganalysis method [29] (denoted as Tar4) to measure the security of DDCA in this section. Table 6 is obtained by Tar4. Table 6 shows the steganalysis results of DDCA on databases DB1 and DB2 at eight embedding rates. It can be observed from Table 6 that, when the embedding rate increases fixing the GOP and QP, there is an evident rise in the detection accuracy of Tar4 [29] for DDCA. As mentioned above, setting a higher embedding rate leads to more modifications of nonzero QDCT coefficients. Furthermore, this provides more opportunities for steganalysers to detect stego videos. For example, when setting GOP:IPPPP-QP:15, the detection accuracy increases from 0.6942 to 0.8793

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3058134, IEEE Transactions on Dependable and Secure Computing

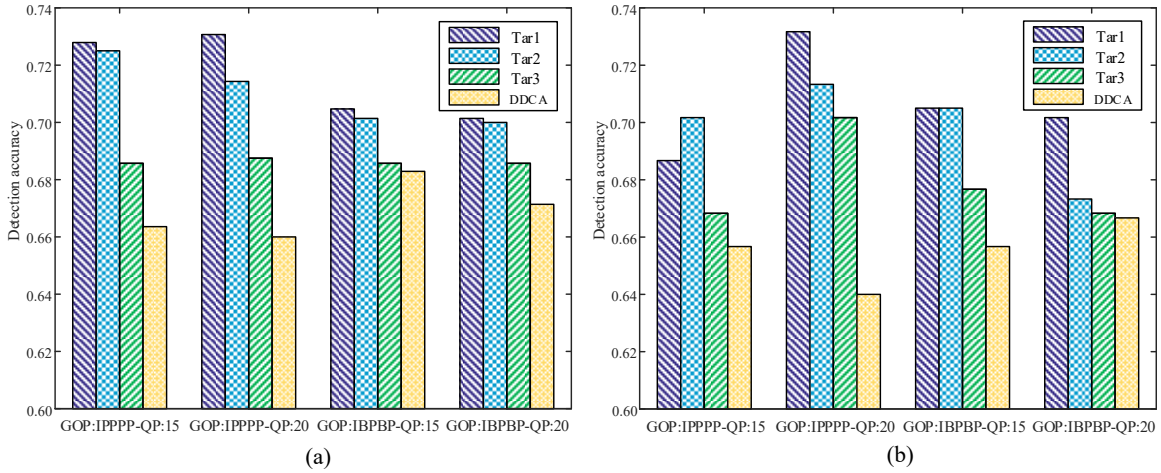


Fig. 14. Comparisons of detection accuracy of Tar1 [23], Tar2 [26], Tar3 [27] and DDCA on databases DB1 and DB2. (a) DB1. (b) DB2.

TABLE 6
Detection accuracy of DDCA with different embedding rates by using Tar4[29].

Database	GOP	QP	Embedding Rate (bpnzAC)							
			0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
DB1	IPPPP	15	0.6943	0.7279	0.7450	0.7864	0.8014	0.8200	0.8457	0.8793
		20	0.7014	0.7457	0.7614	0.8129	0.8271	0.8571	0.8714	0.8786
	IBPBP	15	0.7257	0.7500	0.7821	0.8064	0.8307	0.8479	0.8571	0.8721
		20	0.7307	0.7636	0.7886	0.8250	0.8471	0.8671	0.8907	**0.8814**
DB2	IPPPP	15	0.7017	0.7400	0.7717	0.7917	0.8333	0.8733	0.9050	**0.9050**
		20	0.7333	0.7117	0.7867	0.8400	0.8783	0.9067	0.9217	0.9567
	IBPBP	15	0.7333	0.7500	0.7667	0.7933	0.8317	0.8400	0.9017	**0.8867**
		20	0.6750	0.6867	0.7417	0.7883	0.8400	0.8817	0.9167	0.9467

on database DB1 corresponding to the embedding rates from 0.05 to 0.40 bpnzAC. Likewise, the same conclusion for the detection accuracy can be drawn on databases DB1 and DB2 under different GOPs and QPs. Actually, although the detection accuracies for the embedding payload 0.40 bpnzAC are less than that for 0.35 bpnzAC when setting GOP:IBPBP-QP:20 on DB1 and GOP:IPPPP-QP:15 and GOP:IBPBP-QP:15 on DB2. This may be attributed to the lack of training samples. Therefore, in the future we will continue to collect more video sequences and enlarge the video databases.

To further measure the security of DDCA, the detecting accuracies of Tar1, Tar2, Tar3 and DDCA with different embedding payloads (listed as Table 1) by using Tar4 are shown in Fig. 14. DDCA is based on our proposed framework, it can provide full freedom to STCs to select coefficients with less impacts from all nonzero QDCT AC coefficients of a whole video to change for steganography. Moreover, DDCA is designed and considered from the distortion drift point of view and it can better reflect the real changing distortion. Therefore, Tar4 has the least detection accuracies for DDCA compared with Tar1, Tar2 and Tar3 (shown in Fig. 14). In addition, Tar2 and Tar3 consider the texture of coding blocks in steganography but Tar1 does not. Therefore, Tar4 has less detecting accuracies for Tar2 and Tar3 than Tar1. For Tar2 and Tar3, Tar3 is designed by considering the impacts of adjacent coding blocks in steganography but Tar2 does not. So the detecting accuracies of Tar3 is less than that of Tar2. In summary, DDCA outperforms Tar1, Tar2 and Tar3 against Tar4.

5.6 Discussion on Scaling Factors

With the DDCA, a different setting of (α,β,γ) will update different costs. Furthermore, this will lead to different coding performances and security of stego videos. Equations (20) and (21), i.e., the final distortion cost, can be simplified as:

$$\rho = \alpha\eta + \beta\phi + \gamma\varphi \tag{24}$$

for better analyzing (α,β,γ). η, ϕ and φ respectively denote the impacts of the inner-block, the inter-block and the inter-frame distortion drifts. Therefore, the settings of α, β, and γ reflect the impacts of η, ϕ, and φ. As analyzed in Section 3, the inner-block distortion drift is the reason of inducing the inter-block and the inter-frame distortion drifts by the intra-frame prediction and the inter-frame prediction, respectively. Thereby, setting a larger value for α than β and γ is better. In addition, the settings of β and γ depend on the use of the intra-frame prediction and the inter-frame

TABLE 7
Average PSNR, SSIM and BIR
on DB1 under different setting of scaling factors
and embedding rate is 0.40 bpnzAC and GOP:IBPBP-QP:15.

(α,β,γ)	PSNR(dB)	SSIM	BIR(%)
(0.50,0.25,0.25)	42.5035	0.9874	1.7765
(0.50,0.15,0.35)	**42.5090**	0.9874	**1.7757**
(0.50,0.35,0.15)	42.3966	0.9874	1.7763
(0.60,0.20,0.20)	42.3479	0.9873	1.7758
(0.60,0.10,0.30)	42.2380	0.9874	1.7759
(0.60,0.30,0.10)	42.2326	0.9874	1.7759
(0.70,0.15,0.15)	42.2117	0.9874	1.7758
(0.70,0.10,0.20)	42.1908	0.9874	1.7758
(0.70,0.20,0.10)	42.1699	0.9874	1.7762

prediction in practice. To select a suitable setting of (α,β,γ) according to PSNR, SSIM and BIR, we set different values of (α,β,γ) on DB1 to obtain PSNR, SSIM and BIR shown in Table 7.

Table 7 shows average PSNRs, SSIMs and BIRs on DB1 when setting different values of (α,β,γ), embedding rate at 0.40 bpnzAC and GOP:IBPBP-QP:15. Generally speaking, the larger values PSNR and SSIM have and the smaller values BIR has, the better coding performance stego videos have. As observed from Table 7, the maximum values of PSNR and SSIM are 42.5090 dB and 0.9874 and the minimum value of BIR is 1.7757 under different settings of (α,β,γ). The settings of (α,β,γ) have lower impacts on SSIM and higher impacts on PSNR. Almost all values of SSIM are the same, i.e., 0.9874. In all the settings of (α,β,γ), since DDCA obtains the best coding performances in terms of PSNR, SSIM and BIR when (α,β,γ)=(0.50,0.15,0.35), we set (α,β,γ)=(0.50,0.15,0.35) in Sections 5.3 to 5.5. The settings of (α,β,γ) could be defined as a multi-objective optimisation problem in a future work.

6 Conclusion

In this paper, we proposed a novel video steganographic framework, designed to facilitate full freedom for STCs in modifying all transform coefficients of an entire video. Then, we designed a cost assignment method (DDCA), based on the analysis of the inner-block, the inter-block and the inter-frame distortion costs. All nonzero transform coefficients are fully utilized to improve both the coding performance and the steganogaphic security. We showed that DDCA can effectively measure the modification distortions propagated due to both the intra-frame and the inter-frame predictions. We also demonstrated that the proposed steganogaphic framework achieves distinct improvements in the coding performance and the security, compared with three other state-of-the-art methods [23, 26, 27].

Future research agenda includes the following:

1) Extend the proposed video steganographic framework and DDCA to HEVC [36] and VP9 [37]. When extending our work to HEVC and VP9, a number of modifications are necessary due to the differences between H.264, HEVC and VP9. For instance, H.264 and HEVC respectively have 13 and 35 intra frame prediction modes, but DDCA considers the 13 intra frame prediction modes of H.264 to calculate the inter-block distortion costs for video steganography. Thus, we must also consider the 35 intra frame prediction modes of HEVC to calculate the inter-block distortion costs for video steganography.
2) Design cost assignment methods for other cover elements, such as motion vector and prediction modes, from the distortion drift point of view.
3) Design a non-additive cost assignment method for transform coefficients based on the proposed video steganographic framework.
4) Implement a prototype of the extended work (e.g., comprising the above three extensions) in a real-world context.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61972269 and 61902263), the Fundamental Research Funds for the Central Universities (YJ201881), and the China Scholarship Council (201907000055). K.-K. R. Choo is supported only by the Cloud Technology Endowed Professorship, and National Science Foundation (NSF) CREST Grant HRD-1736209. Dr X. Zhang is the recipient of an ARC DECRA (project number DE210101458) funded by the Australian Government.

References

[1] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.
[2] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans Inf Forensic Security*, vol. 6, no. 3, pp. 920–935, 2011.
[3] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Int Workshop Inf Hiding*. Springer, 2010, pp. 161–177.
[4] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *2012 Int Workshop Inf Forensic Security (WIFS)*. IEEE, 2012, pp. 234–239.
[5] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans Inf Forensic Security*, vol. 10, no. 9, pp. 1905–1917, 2015.
[6] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J Inf Security*, vol. 2014, no. 1, p. 1, 2014.
[7] K. Chen, H. Zhou, W. Zhou, W. Zhang, and N. Yu, "Defining cost functions for adaptive JPEG steganography at the microscale," *IEEE Trans Inf Forensic Security*, vol. 14, no. 4, pp. 1052–1066, 2018.
[8] Y. Tew and K. Wong, "An overview of information hiding in H. 264/AVC compressed video," *IEEE Trans Circuits Syst Video Technol*, vol. 24, no. 2, pp. 305–319, 2014.

[9] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Trans Circuits Syst Video Technol*, vol. 28, no. 9, pp. 2131–2153, 2018.

[10] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans Image Process*, vol. 6, no. 12, pp. 1673–1687, 1997.

[11] Q. Nie, X. Xu, B. Feng, and L. Y. Zhang, "Defining embedding distortion for intra prediction mode-based video steganography," *Comput Mater Contin*, vol. 55, no. 1, pp. 59–70, 2018.

[12] L. Zhang and X. Zhao, "An adaptive video steganography based on intra-prediction mode and cost assignment," in *Int Workshop Digtal Watermakring*. Springer, 2016, pp. 518–532.

[13] X. Y. Yang, L. Y. Zhao, and K. Niu, "An efficient video steganography algorithm based on sub-macroblock partition for H. 264/AVC," in *Advanced Materials Research*, vol. 433. Trans Tech Publ, 2012, pp. 5384–5389.

[14] H. Zhang, Y. Cao, X. Zhao, W. Zhang, and N. Yu, "Video steganography with perturbed macroblock partition," in *Proc 2nd ACM Workshop Inf Hiding Multimedia Security*, 2014, pp. 115–122.

[15] L. Zhai, L. Wang, and Y. Ren, "Multi-domain embedding strategies for video steganography by combining partition modes and motion vectors," in *IEEE Int Conf Multimedia Expo (ICME)*. IEEE, 2019, pp. 1402–1407.

[16] H. A. Aly, "Data hiding in motion vectors of compressed video based on their associated prediction error," *IEEE Trans Inf Forensic Security*, vol. 6, no. 1, pp. 14–18, 2011.

[17] Y. Yao, W. Zhang, N. Yu, and X. Zhao, "Defining embedding distortion for motion vector-based video steganography," *Multimedia Tools Appl*, vol. 74, no. 24, pp. 11 163–11 186, 2015.

[18] Y. Cao, H. Zhang, X. Zhao, and H. Yu, "Covert communication by compressed videos exploiting the uncertainty of motion estimation," *IEEE Commun Lett*, vol. 19, no. 2, pp. 203–206, 2014.

[19] H. Zhang, Y. Cao, and X. Zhao, "Motion vector-based video steganography with preserved local optimality," *Multimedia Tools Appl*, vol. 75, no. 21, pp. 13 503–13 519, 2016.

[20] K. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, "Complete video quality-preserving data hiding," *IEEE Trans Circuits Syst Video Technol*, vol. 19, no. 10, pp. 1499–1512, 2009.

[21] T. Shanableh, "Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering," *IEEE Trans Inf Forensic Security*, vol. 7, no. 2, pp. 455–464, 2012.

[22] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H. 264/AVC video streams without intra-frame distortion drift," *IEEE Trans Circuits Syst Video Technol*, vol. 20, no. 10, pp. 1320–1330, 2010.

[23] Y. Cao, Y. Wang, X. Zhao, M. Zhu, and Z. Xu, "Cover block decoupling for content-adaptive H. 264 steganography," in *Proc 6th ACM Workshop Inf Hiding Multimedia Security*. ACM, 2018, pp. 23–30.

[24] Y. Chen, H. Wang, H. Wu, and Y. Liu, "An adaptive data hiding algorithm with low bitrate growth for H. 264/AVC video stream," *Multimedia Tools Appl*, vol. 77, no. 15, pp. 20 157–20 175, 2018.

[25] Y. Chen, H. Wang, H. Wu, Y. Chen, and Y. Liu, "A data hiding scheme with high quality for H. 264/AVC video streams," in *Int Conf Cloud Comput Security (ICCCS)*. Springer, 2018, pp. 99–110.

[26] Y. Chen, H. Wang, H.-Z. Wu, Z. Wu, T. Li, and A. Malik, "Adaptive video data hiding through cost assignment and STCs," *IEEE Trans Depend Secur Comput*, 2019. doi: 10.1109/TDSC.2019.2932983

[27] Y. Xue, J. Zhou, H. Zeng, P. Zhong, and J. Wen, "An adaptive steganographic scheme for H. 264/AVC video with distortion optimization," *Signal Process: Image Commun*, vol. 76, pp. 22–30, 2019.

[28] P. Xie, H. Zhang, W. You, X. Zhao, J. Yu, and Y. Ma, "Adaptive VP8 steganography based on deblocking filtering," in *Proc ACM Workshop Inf Hiding Multimedia Security (IH&MMSec)*, 2019, pp. 25–30.

[29] P. Wang, Y. Cao, X. Zhao, and M. Zhu, "A steganalytic algorithm to detect DCT-based data hiding methods for H. 264/AVC videos," in *Proc 5th ACM Workshop Inf Hiding Multimedia Security*, 2017, pp. 123–133.

[30] L. Zhai, L. Wang, and Y. Ren, "Universal detection of video steganography in multiple domains based on the consistency of motion vectors," *IEEE Trans Inf Forensic Security*, 2019. doi: 10.1109/TIFS.2019.2949428

[31] K. Wang, H. Zhao, and H. Wang, "Video steganalysis against motion vector-based steganography by adding or subtracting one motion vector value," *IEEE Trans Inf Forensic Security*, vol. 9, no. 5, pp. 741–751, 2014.

[32] H. Zhang, Y. Cao, and X. Zhao, "A steganalytic approach to detect motion vector modification using near-perfect estimation for local optimality," *IEEE Trans Inf Forensic Security*, vol. 12, no. 2, pp. 465–478, 2017.

[33] K. Tasdemir, F. Kurugollu, and S. Sezer, "Spatio-temporal rich model-based video steganalysis on cross sections of motion vector planes," *IEEE Trans Image Process*, vol. 25, no. 7, pp. 3316–3328, 2016.

[34] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans Circuits Syst Video Technol*, vol. 13, no. 7, pp. 560–576, 2003.

[35] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[36] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated circuit and systems, algorithms and architectures*. Springer, 2014, vol. 39, pp. 49–90.

[37] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje, "The latest open-source video codec VP9-an overview and preliminary results," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 390–393.

[38] Y. Yao, W. Zhang, and N. Yu, "Inter-frame distortion drift analysis for reversible data hiding in encrypted H. 264/AVC video bitstreams," *Signal Process*, vol. 128, pp. 531–545, 2016.

[39] Y. Li and H.-X. Wang, "Robust H. 264/AVC video watermarking without intra distortion drift," *Multimedia Tools Appl*, vol. 78, no. 7, pp. 8535–8557, 2019.

[40] 2019, "2019 Global Media Formats Report," 2019. [Online]. Available: https://www.encoding.com/blog/2019/04/04/encoding-com-releases-its-2019-global-media-format-report/

[41] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans Circuits Syst Video Technol*, vol. 22, no. 12, pp. 1649–1668, 2012.

[42] K. Suehring. (2015) H.264/AVC Joint Model Reference Software Group. [Online]. Available: http://iphone.hhi.de/suehring/tml/download/

[43] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid–a framework for video transmission and quality evaluation," in *International conference on modelling techniques and tools for computer performance evaluation*. Springer, 2003, pp. 255–272.

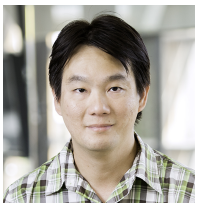
[44] A. Lie and J. Klaue, "Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video," *Multimedia Systems*, vol. 14, no. 1, pp. 33–50, 2008.

[45] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans Inf Forensic Security*, vol. 7, no. 2, pp. 432–444, 2011.

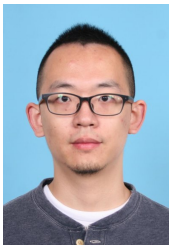
[46] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H. 264/AVC video streams by codeword substitution," *IEEE Trans Inf Forensic Security*, vol. 9, no. 4, pp. 596–606, 2014.

Yi Chen received his B.S degree in Information Security in 2015 from Southwest Jiaotong University, Chengdu, China, where He is currently pursuing his Ph.D. degree in Information and Communication Engineering. His main areas of interest are steganography\steganalysis.

Hongxia Wang is a professor in School of Cyber Science and Engineering at Sichuan University. She received her B.S. degree from Hebei Normal University, Shijiazhuang, in 1996, and her M.S. and Ph.D. degrees from University of Electronic Science and Technology of China, Chengdu, in 1999 and 2002, respectively. She pursued postdoctoral research work in Shanghai Jiao Tong University from 2002 to 2004. She was a professor at Southwest Jiaotong University from 2004 to 2018. Her research interests include multimedia information security, digital forensics, information hiding & digital watermarking, and intelligent information processing. She has published over 200 peer research papers and wined 15 authorized patents.



Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). He currently serves as the Department Editor of IEEE Transactions on Engineering Management, the Associate Editor of IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Big Data. He is an IEEE Computer Society Distinguished Visitor (2021 - 2023), and included in Web of Science's Highly Cited Researcher in the field of Cross-Field - 2020. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), and several other awards.



Peisong He received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2013 and the Ph.D. degree in Cybersecurity from Shanghai Jiao Tong University, Shanghai, China, in 2018.

He was working as a visiting student at the Rapid-Rich Object Search Lab of the Nanyang Technological University, Singapore, from 2016 to 2017. In 2019, he joined Sichuan University, Chengdu, China, where he is currently an Assistant Researcher. His current research interest includes multimedia forensics and security.

Zoran Salcic (Life Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical and computer engineering from Sarajevo University in 1972, 1974, and 1976, respectively. He is a Professor and the Chair of computer systems engineering with The University of Auckland, New Zealand. He has published more than 400 peer-reviewed journals and conference papers, and several books. His main research interests include various aspects of cyber-physical systems that include complex digital systems design, custom-computing machines, design automation tools, hardware–software co-design, formal models of computation, sensor networks and Internet of Things, and languages for concurrent and distributed systems and their applications such as industrial automation, intelligent buildings and environments, and collaborative systems with service robotics and many more. He is a Fellow of the Royal Society of New Zealand. He was a recipient of the Alexander von Humboldt Research Award in 2010.

Dali Kaafar is the Executive Director of the Optus-Macquarie Cyber Security Hub and Professor of Privacy Preserving Technologies in the Faculty of Sciences and Engineering at Macquarie University. He is also the founder of the Information Security and Privacy group at CSIRO Data61. His current research interests and expertise include Privacy Preserving Technologies, Privacy Risks assessment, Networks Security, Web Security and malware detection, Next generation Authentication systems with a focus on Information Theory and Applied Data Mining, Cryptanalysis and Applied Cryptography and Distributed Systems and Risks Modelling. He is the associate editor of the IEEE Transactions on Information Forensics and Security and serves in the Editorial Board of the journal on Privacy Enhancing Technologies. Professor Kaafar holds a PhD in Computer Science from INRIA Sophia Antipolis, Polytechnic University of Nice Sophia Antipolis in France where he pioneered research in the security of Internet Coordinate Systems Professor Kaafar published over 300 scientific peer-reviewed papers with several and repetitive publications in the prestigious ACM SIGCOMM, IEEE INFOCOM, NDSS and PETS.

Xuyun Zhang is currently working as a senior lecturer in Department of Computing at Macquarie University (Sydney, Australia). Besides, he has the working experience in University of Auckland and NICTA (now Data61, CSIRO). He received his PhD degree in Computer and Information Science from University of Technology Sydney (UTS) in 2014, and his MEng and BSc degrees from Nanjing University. His research interests include scalable and secure machine learning, big data mining and analytics, cloud/edge/service computing and IoT, big data privacy and cyber security, etc. He is the recipient of 2021 ARC DECRA Award and several other prestigious awards. He has served as special issue guest editors for several high-quality journals like IEEE Trans. Industrial Informatics and IEEE Trans. Emerging Topics in Computational Intelligence.