Temporal Forensics and Anti-Forensics for Motion Compensated Video

Matthew C. Stamm, *Member, IEEE*, W. Sabrina Lin, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Due to the ease with which digital information can be altered, many digital forensic techniques have been developed to authenticate multimedia content. Similarly, a number of anti-forensic operations have recently been designed to make digital forgeries undetectable by forensic techniques. However, like the digital manipulations they are designed to hide, many anti-forensic operations leave behind their own forensically detectable traces. As a result, a digital forger must balance the trade-off between completely erasing evidence of their forgery and introducing new evidence of anti-forensic manipulation. Because a forensic investigator is typically bound by a constraint on their probability of false alarm (P_{fa}), they must also balance a trade-off between the accuracy with which they detect forgeries and the accuracy with which they detect the use of anti-forensics. In this paper, we analyze the interaction between a forger and a forensic investigator by examining the problem of authenticating digital videos. Specifically, we study the problem of adding or deleting a sequence of frames from a digital video. We begin by developing a theoretical model of the forensically detectable fingerprints that frame deletion or addition leaves behind, then use this model to improve upon the video frame deletion or addition detection technique proposed by Wang and Farid. Next, we propose an anti-forensic technique designed to fool video forensic techniques and develop a method for detecting the use of anti-forensics. We introduce a new set of techniques for evaluating the performance of anti-forensic operations and develop a game theoretic framework for analyzing the interplay between a forensic investigator and a forger. We use these new techniques to evaluate the performance of each of our proposed forensic and anti-forensic techniques, and identify the optimal actions of both the forger and forensic investigator.**

***Index Terms*—Digital forensics, anti-forensics, frame deletion, video compression, game theory.**

I. Introduction

WITHIN the past decade, a great deal of research has been performed in the field of digital multimedia forensics. Digital forensic techniques seek to provide information about digital multimedia content without relying on external descriptors such as metadata tags or extrinsically implanted information such as digital watermarks. Instead, these techniques make use of fingerprints left in digital content by editing operations or the digital capture process. Because multimedia content can be easily altered using digital editing software, digital forensic techniques have become extremely important to organizations seeking to verify the integrity of digital content. Forensic techniques have been developed to perform a variety of tasks such as detecting evidence of editing or forgery, identifying a media file's origin, and tracing multimedia content's processing history for digital images [1]–[5], video [6]–[8], and audio [9]–[11].

Though these digital forensic techniques are quite successful at identifying standard digital content manipulations, researchers have demonstrated that many of them can be fooled if a forger makes use of *anti-forensics*. Anti-forensic techniques are designed to mislead forensic analysis by erasing or falsifying fingerprints left by editing operations. By studying anti-forensics, digital security researchers are able to identify weaknesses in existing forensic algorithms and quantify how much confidence they can have in their forensic results. In prior work, anti-forensic techniques have been proposed to remove traces of image resizing and rotation [12] and forge the photo-response nonuniformity (PRNU) fingerprint left in an image by a digital camera's electronic sensor [13]. A set of anti-forensic techniques has been proposed to erase or falsify an image's compression history [14]–[16]. These techniques can be used to fool forensic algorithms that identify image forgeries by searching for inconsistencies in an image's compression history [16], [17]. Furthermore, anti-forensic techniques to remove contrast enhancement fingerprints [18] and to artificially synthesize color filter array artifacts used for camera identification or forgery detection [19] have been proposed.

Just as digital editing operations leave behind fingerprints, anti-forensic operations may inadvertently leave behind their own fingerprints [14]. If these fingerprints can be identified, forensic techniques can be designed to detect them. This will allow forensic investigators to identify digital forgeries even when editing fingerprints have been anti-forensically removed. Researchers have recently developed techniques to identify anti-forensic manipulation of an image's PRNU [20] and compression history [21].

When confronted with a forensic technique capable of detecting the use of an anti-forensic operation, an intelligent forger will attempt to modify their anti-forensic operation in order to minimize the strength of the fingerprint it leaves behind. This leads to a cat-and-mouse game between a digital forger and a forensic investigator. Furthermore, a digital forger can opt not to completely anti-forensically remove all editing fingerprints left in their forgery. Instead, they may decrease the strength of their anti-forensic operation so that it reduces the strength of

Manuscript received October 17, 2011; revised May 22, 2012; accepted June 01, 2012. Date of publication June 21, 2012; date of current version July 09, 2012. This work is supported in part by AFOSR Grant FA95500910179. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alex ChiChung Kot.

M. C. Stamm and K. J. R. Liu are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: mcstamm@umd.edu; kjrliu@umd.edu).

W. S. Lin is with the Intel Corporation, Hillsboro, OR 97124 USA (e-mail: w.sabrina.lin@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2012.2205568

the editing operation's fingerprint to just below a forensic investigator's detection threshold. This will correspondingly reduce the strength of the anti-forensic operation's fingerprint, thus helping the attacker avoid detection. The forensic investigator, meanwhile, must ensure that the combination of the false alarm rates from their techniques to detect editing and the use of anti-forensics is below a constant false alarm rate.

This interplay between a forensic investigator and a digital forger raises a number of important questions. For example, if a forensic technique is effective at detecting a particular type of forgery but can easily be fooled if a forger makes use of anti-forensics, is it a good or bad detection algorithm? Similarly, if an anti-forensic operation is able to successfully remove fingerprints left by a particular forgery operation but introduces new fingerprints of its own, how do we evaluate its effectiveness? What is the optimal strategy for a forger to use to avoid forensic detection of both their forgery and their use of anti-forensics? What is the optimal detection strategy for a forensic investigator to follow when attempting to identify digital forgeries? Should decision thresholds in forensic detection techniques be chosen to yield the best performance under a worst case scenario, or can knowledge of the attacker's actions be used to improve detection results? Are there certain editing operations that an attacker will be unable to hide both forensic and anti-forensic evidence of their forgery?

To address these questions, we analyze the interaction between a digital forger and a forensic investigator in a particular forensic scenario. In this paper, we consider the problem of forensically detecting video frame deletion or addition. Frame deletion may be performed by a video forger who wishes to remove certain portions of a video sequence, such as a person's presence in a surveillance video. Similarly, a forger may wish to falsify an event by inserting a sequence of new frames into a video segment. In previous work, Wang and Farid demonstrated that frame deletion or addition followed by recompression introduces a forensically detectable fingerprint into MPEG video [6]. Though their technique is quite effective, it requires human identification of frame deletion or addition fingerprints and can only be used on videos compressed by a certain class of video encoders that employ a fixed group of picture (GOP) structure.

In this paper, we propose new video frame deletion or addition forensic and anti-forensic techniques along with a new framework for evaluating the interplay between a forger and forensic investigator. The main contributions of this work can be summarized as follows:

- We propose a mathematical model of video frame deletion and addition fingerprints that show themselves in a video's P-frame prediction error sequence.
- We use this model to develop two new automatic video frame deletion or addition detection technique. One of these techniques is targeted towards video codecs that use fixed length GOPs when compressing a video, while the other is suitable for use with newer compression standards that allow the GOP length to change adaptively.
- We propose an anti-forensic technique capable of hiding frame deletion or addition fingerprints in digital videos. This technique operates by first constructing a target P-frame prediction error sequence that is free from fingerprints, then selectively altering the video's predicted frames so that the prediction error sequence from the anti-forensically modified video matches the target one.
- We identify a new fingerprint that frame deletion or addition anti-forensics introduces into a modified video's motion vectors and propose a forensic scheme designed to detect it. Additionally, we modify our proposed anti-forensic technique to minimize detection by these means.
- We define a new set of terms to use when evaluating the performance of both forensic and anti-forensic algorithms.
- We propose a set of game theoretic techniques to study the dynamics between a digital forger and a forensic investigator. We do this by formulating each party's utility functions in terms of the probabilistic quantities associated with the performance of their forensic detection technique or anti-forensic operation.
- We use our new techniques to evaluate the forensic and anti-forensic algorithms proposed in this paper.

The remainder of this paper is organized as follows. In Section II we provide an overview of the background material relevant to frame deletion and addition fingerprints, and develop our mathematical model of these fingerprints. In Section III, we use this model to construct a set of automatic frame deletion or addition detection techniques. We propose our anti-forensic technique to remove frame deletion and addition fingerprints in Section IV. We then identify the new fingerprints left by this anti-forensic technique, use these fingerprints to develop an algorithm to detect the use of anti-forensics, and modify our proposed anti-forensic technique in response to this in Section V. We discuss the performance evaluation of forensics and anti-forensic algorithms in Section VI and develop our game theoretic techniques to evaluate the dynamics between a forger and forensic investigator. We present the results of several experiments designed to evaluate the performance of each of our proposed techniques in Section VII. Finally, we conclude the paper in Section VIII.

II. Frame Deletion Fingerprints

We begin this section with a brief overview of video compression, with an emphasis on the forensically significant aspects. Next, we discuss prior forensic work on video frame deletion or addition detection. We then propose a new mathematical model of frame deletion and addition fingerprints which we will use to develop our forensic and anti-forensic techniques.

A. Video Compression Overview

Due to the size of uncompressed digital video files, virtually all digital video undergoes compression during storage or transmission. Though a variety of different video compression techniques exist, the majority operate in the same basic manner. Since a scene typically changes very little over a short period of time, a great deal of redundancy exists between video frames. Video encoders exploit this redundancy by predicting certain frames from others, then storing the prediction error. The prediction error can be compressed at a higher rate than the frame itself, allowing for smaller file sizes.

In order to prevent the propagation of channel and decoding errors, not all frames are predicted. Instead, the video sequence

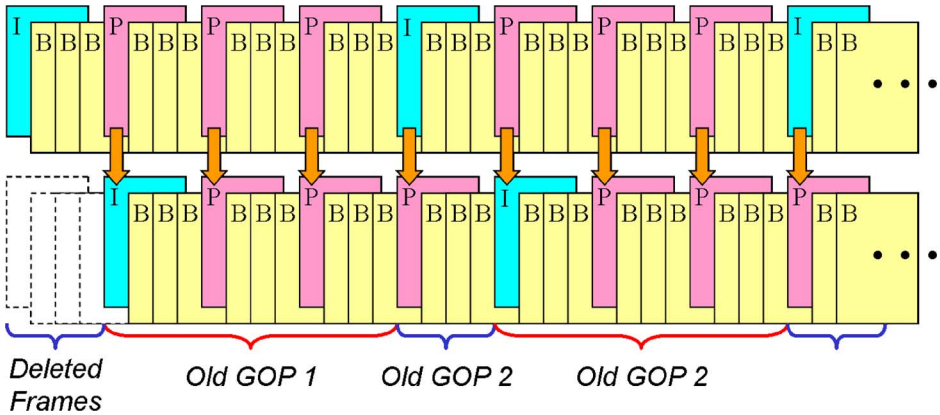

Fig. 1. Illustration of the effects of frame deletion on a video frame sequence. The original video sequence is shown along the top of this figure and the altered video sequence is shown along the bottom. Each GOP in the altered video contains frames from two different GOPs in the unaltered video sequence.

is segmented into sets of frames known as 'groups of pictures' (GOPs). Frames are predicted from other frames in the same GOP, but prediction does not occur across GOPs. Within each GOP, frames are assigned one of three types according to the manner in which they are predicted and compressed. These frame types are known as: intraframes (I-frames), predicted-frames (P-frames), and bidirectional-frames (B-frames).

Each GOP begins with an I-frame. I-frames are not predicted from any other frame and are independently encoded using a process similar to JPEG compression. The remainder of each GOP consists of P-frames and B-frames. These frames are predictively encoded using processes known as motion estimation and compensation. A predicted version of the encoded frame is formed from segments of an anchor frame or frames. Only I-frames and P-frames may act as anchor frames.

P-frame motion estimation is performed by first segmenting the frame into a series of macroblocks. Next, the preceding anchor frame is searched for the macroblock that best matches each macroblock in the current P-frame. The row and column displacements between each macroblock in a P-frame and its match in the anchor frame are recorded as that macroblock's row and column motion vectors. A motion-compensated, predicted version of the P-frame is formed by assembling each of the matching macroblocks from the anchor frame. The predicted frame is then subtracted from the actual P-frame, resulting in the P-frame's prediction error. This prediction error is compressed using the same JPEG-like process used to encode I-frames.

During storage and transmission, only the motion vectors and prediction errors are retained. To decompress these frames, the predicted version of the P-frame is reformed using its motion vectors and the previous anchor frame, which must be decoded first. Next, the prediction error is decompressed and added to the predicted frame, thus reconstructing the frame. B-frames are encoded in a similar manner, however each macroblock frame can be predicted from the anchor frame that immediately precedes the B-frame, immediately follows the B-frame, or an average of these two predictions can be used.

In older video compression standards, the structure of each GOP is fixed, i.e., the sequence of I-, P-, and B-frames always occurs in the same pattern. Newer video compression standards allow for the GOP structure to be adjusted depending on the amount of motion in the scene. For example, rapidly changing scenes can be encoded using shorter GOPs because the accuracy of motion compensation greatly decreases as new objects enter each frame.

B. Detection of Frame Deletion or Addition

In a number of scenarios, a video forger may wish to add or delete frames from a digital video sequence. To do this, the forger must decompress the video before frames are added or deleted, then recompress the video after it has been altered. Previous work by Wang and Farid has shown that recompression of MPEG video using a fixed GOP structure results in two distinct, forensically detectable fingerprints; one spatial and the other temporal [6]. The spatial fingerprint can be observed within a single I-frame and is similar in nature to the fingerprint left by double JPEG compression [1], [22]. This fingerprint occurs when either no frames are added or deleted, or when the number of frames added or deleted is an integer multiple of the fixed GOP length. The temporal fingerprint occurs in the sequence of P-frame prediction errors and occurs only if frames have been added to or deleted from the video sequence prior to recompression.

When frames are deleted from or added to a digital video, each GOP in the recompressed video will contain frames that belonged to different GOPs during the initial compression. This effect can be seen in Fig. 1, which shows an example of frame deletion for a video compressed using a fixed GOP sequence. Wang and Farid experimentally demonstrated that when a P-frame is predicted from an anchor frame that initially belonged to a different GOP, an increase in the total prediction error is observed [6]. Furthermore, they demonstrated that if a fixed GOP structure is used, this increase in prediction error occurs periodically in the sequence of P-frame prediction errors. As a result, they proposed detecting frame deletion or addition by visually inspecting the sequence

$$e(n) = \frac{1}{N_{xy}} \sum_{x} \sum_{y} |p_{x,y}(n)|, \tag{1}$$

for a periodic fingerprint, where N_{xy} is the number of pixels in each frame and $p_{x,y}(n)$ is the prediction error of the nth P-frame at pixel location (x, y) [6]. Alternately, the discrete Fourier transform (DFT) of this sequence $E(k) = \mathrm{DFT}\{e(n)\}$

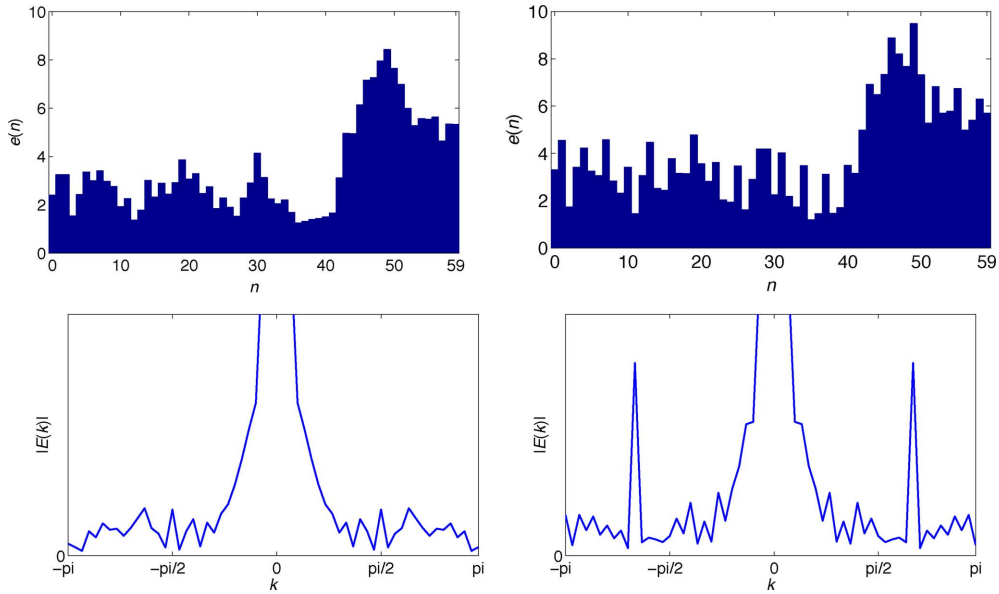

Fig. 2. P-frame prediction error sequence (top left) and the magnitude of its DFT (bottom left) obtained from an unedited, compressed version of the "Carphone" video sequence along with the P-frame prediction error sequence (top right) and the magnitude of its DFT (bottom right) obtained from the same video after frame deletion followed by recompression.

can be inspected for peaks resulting from the periodic fingerprint. An example of this fingerprint can be seen in Fig. 2 which shows the P-frame prediction error sequence of 250 frames of a compressed version of the commonly used "Carphone" video, along with the P-frame prediction error sequence of the same video after the first 6 frames have been deleted followed by recompression.

While this frame addition or deletion detection technique is quite successful, it possesses several shortcomings. Because it requires human inspection of the P-frame prediction error sequence or its DFT, Wang and Farid's detection technique can not be run automatically on large amounts of data and is subject to human error. Furthermore, its reliance on human inspection makes it difficult to characterize the performance of this detection technique using a receiver operating characteristic (ROC) curve or other statistical measure. Most importantly, because this detector relies on identifying periodic increases within the P-frame prediction error sequence, it can only be used on videos that are compressed by a codec with a fixed GOP pattern. It cannot be used on videos compressed using more recently developed encoders if their implementations adaptively change the GOP length. This is because the increase in the P-frame prediction error will not occur periodically unless a fixed GOP pattern is used.

C. Temporal Fingerprint Model

In order to design an automatic frame deletion or addition detection technique as well as an anti-forensic method to remove frame addition and deletion fingerprints, we have developed a model of the effect of frame deletion or addition followed by recompression on a video's P-frame prediction error sequence. To simplify our discussion, we will consider only frame deletion for the remainder of this paper. Each of the equations and techniques presented hereafter can be modified to accommodate frame addition by viewing it as the deletion of a negative number of frames.

Let $e_1(n)$ denote the P-frame prediction error sequence of an unaltered video that has been compressed once and let $e_2(n)$ denote the prediction error sequence of that same video after n_D frames have been deleted followed by recompression. We model the relationship between the altered and unaltered videos' P-frame prediction error sequences using the equation

$$e_2(n) = e_1(n - n_D)(1 + s(n)). \tag{2}$$

In this equation, the signal $s(n)$ denotes the temporal fingerprint caused by frame deletion. We propose two different models of the temporal fingerprint based on whether the video codec used to perform compression employed a fixed length GOP or an adaptively changing one.

1) Model for Fixed Length GOPs: As was discussed previously, Wang and Farid demonstrated that when using a video codec with a fixed GOP structure frame deletion followed by recompression introduces a periodic trace into a video's P-frame prediction error sequence. Naturally, this leads us to model $s(n)$ in this situation as a periodic signal. The temporal fingerprint's periodicity arises because frame deletion causes a constant shift in the position of each GOP used during the initial compression relative to the locations of the GOPs used during recompression. As a result, each new GOP will contain frames from exactly two GOPs present during the initial application of compression in a repetitive fashion. Using this information and defining T as the period of the temporal fingerprint, we can show that the temporal fingerprint exhibits the following three properties [23]:

Property 1: The temporal fingerprint's repetitive pattern corresponds to a disproportionate increase in $e(n)$ exactly once per fingerprint period.

Property 2: The period T of the temporal fingerprint is equal to the number of P-frames within a GOP.

Property 3: Define the phase ϕ of the temporal fingerprint as the number of P-frames within a GOP before the increase in $e(n)$ due to frame deletion. The phase is determined by the

equation $\phi = \lfloor |\mathcal{A}|/n_P \rfloor$, where n_P is the number of P-frames within a GOP, $\mathcal{A}$ is the set of frames at the beginning of each GOP that belonged to the same GOP during the initial application of compression, $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$, and $\lfloor \cdot \rfloor$ denotes the floor operation.

To justify these properties, we note that increases in the P-frame prediction error sequence due to the temporal fingerprint occur when a P-frame is predicted from an anchor frame that belonged to a different GOP during the initial compression. Since each new GOP is comprised of frames from only two GOPs used during the initial application of compression, a P-frame will only be predicted in this manner once per GOP. This justifies the first property. The second property arises because the sequence $e(n)$ consists only of P-frame prediction errors, thus spikes in $e(n)$ due to the temporal fingerprint will be separated by the number of P-frames in a GOP. The third property follows directly from the first two properties. We note that by defining n_G as the number of frames in a GOP and n_F as the number of frames in the video sequence that precede the deleted frames, $|\mathcal{A}|$ is given by the equation $|\mathcal{A}| = n_G - ((n_D + n_F) \mathrm{mod} n_G)$.

Based on these properties, we model the temporal fingerprint as

$$s(n) = \beta \mathbb{1}((n - \phi) \mathrm{mod} T = 0), \tag{3}$$

where $\beta > 0$ and $\mathbb{1}(\cdot)$ denotes the indicator function. This corresponds to modeling the P-frame prediction error sequence of an altered video as a shifted version of the unaltered video's prediction error sequence that is scaled by $(1 + \beta)$ once per fingerprint period.

2) Model for Variable Length GOPs: Newer video compression standards allow the GOP length to vary based on the amount of motion in a scene. When frames are deleted from a video then recompressed using one of these codecs, GOPs in the recompressed video will be comprised of frames belonging to multiple different GOPs used during the first compression, but this will not occur in a repeating pattern. Some new GOPs may contain frames from more than two GOPs used during the original compression, while others will contain frames from only one. Nonetheless, frame deletion will alter the GOP which each frame belongs to, but in a random fashion rather than a fixed one. As a result, spikes in the P-frame prediction error sequence occur in a random fashion.

To capture this behavior, we model the P-frame prediction error sequence of a video compressed using variable GOP lengths as

$$s(n) = \beta \mathbb{1}(\Theta(n) = 0), \tag{4}$$

where $\beta > 0$ is a constant and $\Theta(n)$ is a random variable distributed over the set $\{0, 1\}$. Using this model corresponds to modeling the prediction error sequence of an altered video as a shifted version of the altered version's prediction error sequence with randomly selected values scaled by $(1 + \beta)$.

III. Detecting Frame Deletion

To address the weaknesses in Wang and Farid's detection technique, we propose two automatic frame deletion or addition detection techniques; one which exploits the periodic nature of frame deletion fingerprints for fixed GOP length encoders and another suitable for use on videos compressed using variable GOP lengths. We develop these techniques in this section by posing frame deletion detection as a hypothesis testing scenario. We keep the convention that $e(n)$ is the observed P-frame prediction error sequence associated with a video in question, $e_1(n)$ is the prediction error sequence of that video before frames have been deleted, and $e_2(n)$ is the prediction error sequence of the video after frames have been deleted followed by recompression.

Using the convention that the null hypothesis corresponds to the video being unaltered, along with our model from Section II.C, detecting frame deletion can be viewed as differentiating between the following two hypotheses:

$$\begin{aligned} H_0 &: e(n) = e_1(n), \\ H_1 &: e(n) = e_2(n) = e_1(n) + s(n)e_1(n). \end{aligned} \tag{5}$$

It is clear from this problem formulation that detecting frame deletion is equivalent to detecting the presence of the term $s(n)e_1(n)$. In order to do this, however, we require some knowledge of what the P-frame prediction error sequence of the unaltered video is. We obtain an estimate of this signal by median filtering the observed prediction error sequence according to the formula

$$\hat{e}(n) = \mathrm{median}\{e(n-1), e(n), e(n+1)\}. \tag{6}$$

This estimate has the property that it removes the impulsive spikes in prediction error corresponding to frame deletion fingerprints, while leaving the prediction error sequence of an unaltered video largely intact. We model the relationship between this estimate and the true value of $e_1(n)$ as

$$e_1(n) = \hat{e}(n) + \epsilon(n), \tag{7}$$

where $\epsilon(n)$ is a zero mean random variable representing estimation error.

Using this estimate of the unaltered video's P-frame prediction error sequence, we calculate $\hat{s}(n)$, an estimate of the fingerprint signal modulated by the prediction error sequence according to the equation

$$\hat{s}(n) = \max(e(n) - \hat{e}(n), 0). \tag{8}$$

If the frame deletion fingerprint is present, $\hat{s}(n)$ will be composed of the modulated fingerprint signal $e_1(n)s(n)$ plus the noise term ϵ. We take the maximum of the difference between $e(n)$ and $\hat{e}(n)$ and zero because we know that the term $e(n)s(n)$ is nonnegative.

Now we can reframe our detection problem as differentiating between the following two hypotheses:

$$\begin{aligned} H_0 &: \hat{s}(n) = \max(\epsilon(n), 0), \\ H_1 &: \hat{s}(n) = \max(s(n)e_1(n) + \epsilon(n), 0). \end{aligned} \tag{9}$$

This is equivalent to detecting the presence of the modulated frame deletion fingerprint signal $e_1(n)s(n)$ in noise.

If the video codec used to perform compression uses a fixed GOP structure, we are able to leverage the periodic nature

of $s(n)$ when performing detection. Because the number of P-frames in one GOP can be determined from the encoded video, the detector can assume knowledge of the fingerprint's period. The phase, however, is unknown to the detector because it depends on information (the number of frames deleted and the point in the video sequence at which frame deletion occurs) that is hidden from the forensic investigator. As a result, fingerprint detection is well suited for the frequency domain, where the presence of a periodic signal can be readily determined without requiring information about its phase.

To perform frame deletion detection when the video codec uses a fixed GOP structure, we first calculate $\hat{S}(k) = |\mathrm{DFT}\{\hat{s}(n)\}|$, the magnitude of the DFT of the video in question's P-frame prediction error sequence. For a prediction error sequence N frames long, a peak will occur in $\hat{S}(k)$ at $k^* = N/T$ if frame deletion fingerprints are present. We measure the strength of this peak using the detection statistic ρ, defined as

$$\rho = \frac{\hat{S}(k^*)}{\sum_{k\in\Omega} w(k)\hat{S}(k)} \tag{10}$$

where $w(k) = ce^{-\lambda(k-k^*)^2}$ and $\Omega = \{k | k \leq N/2, k \neq 1, k^*\}$. The function $w(k)$ is used to weight $\hat{S}(k)$ values closer to k^* more than those further away. The variable c is a normalizing constant chosen such that $\sum_{k\in\Omega} w(k) = 1$.

We have observed that for videos with very low average prediction error levels, the total prediction error for P-frames predicted from I-frames is slightly more than for P-frames predicted from other P-frames. By requiring videos with very low average prediction error powers to exhibit stronger periodic fingerprints as evidence of frame deletion, we are able to reduce the number of false alarms. We detect frame deletion using the following decision rule:

$$\delta_{\mathrm{fixed}} = \begin{cases} H_0 & \text{if } \rho e^{\gamma e_{\mathrm{avg}}} < \tau_{\mathrm{fixed}} \\ H_1 & \text{if } \rho e^{\gamma e_{\mathrm{avg}}} \geq \tau_{\mathrm{fixed}}, \end{cases} \tag{11}$$

where τ_{fixed} is a decision threshold, γ is a scalar constant, and e_{avg} is the average of the prediction error sequence $e(n)$.

If the video is compressed using a newer video compression standard that uses variable GOP lengths, the frame deletion fingerprint will not be periodic. In this case, frame deletion detection is equivalent to detecting an unknown signal in the presence of noise. As a result, we use an energy detector to identify the presence of $s(n)$. This yields the following decision rule

$$\delta_{\mathrm{var}} = \begin{cases} H_0 & \text{if } \frac{1}{N}\sum_{n=1}^{N} |\hat{s}(n)| < \tau_{\mathrm{var}} \\ H_1 & \text{if } \frac{1}{N}\sum_{n=1}^{N} |\hat{s}(n)| \geq \tau_{\mathrm{var}}. \end{cases} \tag{12}$$

where τ_{var} is a decision threshold. While the periodicity based decision rule δ_{fixed} cannot be used on videos compressed with variable GOP lengths, the energy detector based decision rule δ_{var} can be used on any video.

IV. Frame Deletion Anti-Forensics

If a forger wishes to undetectably delete a sequence of frames from a digital video, they must remove frame deletion fingerprints from the video's P-frame prediction error sequence. The forger is constrained, however, in how they accomplish this. Any anti-forensic technique designed to accomplish this must not introduce an unacceptable amount of distortion into the anti-forensically modified video. Furthermore, the anti-forensically modified video must be decodable by standard video decoders.

In order to develop an anti-forensic technique to remove frame deletion fingerprints, let us first examine how a video's prediction error sequence can be manipulated. Each frame's prediction error is dependent on the accuracy of the predicted version of that frame. Normally, video encoders attempt to create highly accurate predictions of each frame so that the total prediction error is minimized. This reduces the size of the compressed video file. If a less accurate prediction technique is used, the total prediction error for a frame increases. In fact, any total prediction error value associated with a valid frame prediction is achievable. This implies that the total prediction error for a frame can be increased by purposefully choosing motion vectors that yield a poor predicted frame. We note that doing this does not introduce new distortion into the video since each frame can still be recovered by reconstructing its predicted version from the set of encoded motion vectors, then adding the prediction error to the predicted frame.

Using this information, we propose an anti-forensic technique that operates roughly as follows. First, we construct a target P-frame prediction error sequence $\tilde{e}(n)$ that is free from frame deletion fingerprints. Next, we increase the prediction error for each P-frame until the target prediction error is reached. We do this by selectively setting the motion vectors for certain macroblocks to zero, then recalculating the prediction error associated with that macroblock. By modifying the video in this way, we are able to meet both of the previously mentioned criteria imposed upon the anti-forensic technique.

When constructing the target prediction error sequence, we must ensure that it is achievable. Since we can only increase the prediction error, this implies that $\tilde{e}(n) \geq e_2(n)$. Nonetheless, we still wish to keep the prediction error as small as is reasonably possible. With this in mind, we construct our target prediction error sequence by setting $\tilde{e}(n) = e_2(n)$ for values of n for which $e_2(n) = (1+\beta)e_1(n)$. If the encoder used to compress the video employs a fixed GOP structure, this will correspond to n values such that $(n-\phi) \bmod T = 0$. Otherwise, these n values can be identified by comparing the GOP sequence of the unaltered video to the GOP sequence used during recompression. We determine the remaining values of $\tilde{e}(n)$ by interpolating them using a cubic spline. This ensures that no frame deletion fingerprints will occur in the target P-frame prediction error sequence.

After we have generated the target prediction error sequence, we must modify the motion vectors and prediction errors of each P-frame so that the actual P-frame prediction error matches the target error. Since we chose $\tilde{e}(n) = e_2(n)$ for values of n where $e_2(n) = (1+\beta)e_1(n)$, we do not need to modify these P-frames. For the remaining P-frames, we determine the increase in the prediction error incurred by each macroblock if its motion vectors are set to zero. We then zero out the motion vectors of the macroblocks whose prediction error increases the least until the target prediction error level is reached. An explicit description of this procedure is provided below.

Let $b_{i,j}(n)$ denote of sum of the absolute value of the prediction error in the macroblock at location (i,j) in the nth P-frame when motion prediction is used and let $\hat{b}_{i,j}(n)$ be the sum of the absolute value of the prediction error in the same location when the macroblock's motion vector has been set to zero. We define the increase in the macroblock's prediction error caused by setting its motion vector to zero as

$$q_{i,j}(n) = \hat{b}_{i,j}(n) - b_{i,j}(n). \tag{13}$$

We note that $q_{i,j}(n) \geq 0$ because the zero motion vector is included in the search space for the optimal motion vector during compression.

Next, we define $\mathcal{Q}^{(l)}(n)$ as the set of indices of the macroblocks that result in the l smallest prediction error increases when their motion vectors are set to zero. More explicitly, $\mathcal{Q}^{(l)}(n)$ is defined as

$$\mathcal{Q}^{(l)}(n) = \left\{ (i,j) | q_{i,j}(n) \leq q^{(l)}(n) \right\}, \tag{14}$$

where $q^{(l)}(n)$ is the lth smallest entry of $q(n)$.

The total absolute prediction error $g_n(l)$ in the nth frame that results from setting the motion vectors of each macroblock whose indices are in $\mathcal{Q}^{(l)}(n)$ to zero is given by the equation

$$g_n(l) = \sum_{(i,j) \in \mathcal{Q}^{(l)}(n)} \hat{b}_{i,j}(n) + \sum_{(i,j) \notin \mathcal{Q}^{(l)}(n)} b_{i,j}(n). \tag{15}$$

The value of l that minimizes the absolute distance between the target prediction error level and the actual prediction error level is

$$l^* = \arg\min_l |g_n(l) - \hat{e}(n)|. \tag{16}$$

To remove the temporal fingerprint from the nth P-frame of the recompressed video, we set the motion vectors of each macroblock whose indices are in $\mathcal{Q}^{(l^*)}(n)$ to zero, then recompute the prediction error at these macroblock locations during recompression. Due to the relatively small number of macroblocks in each frame, we find l^* for each frame through an exhaustive search.

In some instances, the target prediction error value for a particular P-frame is greater than the error incurred by setting all of the frame's motion vectors to zero. If this is the case, we search first for the set of motion vectors that maximize the prediction error associated with each macroblock. Because many decoders place a limit on the maximal length of each motion vector, this search must be conducted over the set of allowable motion vectors for a given codec. We increase the frame's prediction error by changing several of its motion vectors to these new, maximal error motion vectors rather than by setting them to zero. The rest of our anti-forensic technique remains the same.

V. Detecting the Use of Frame Deletion Anti-Forensics

In the introduction to this paper, we discussed the possibility that anti-forensic operations may leave behind new fingerprints of their own. In this section, we show that this is true for the case of frame deletion and addition anti-forensics.

In order to remove frame deletion fingerprints from the P-frame prediction sequence of a video, that video's motion vectors must be altered in order to increase the prediction error. Despite this, the true motion present in the video does not change. As a result, there is a discrepancy between many of the motion vectors stored in an anti-forensically modified video and the true motion of that video scene. This is not the case for an unaltered video because normal video encoders will attempt to estimate scene motion as accurately as possible in order to minimize each frame's prediction error. Accordingly, these discrepancies between a video's stored motion vectors and the actual motion of the scene are fingerprints left by frame deletion anti-forensics.

To detect the use of frame deletion anti-forensics, we propose comparing a compressed video's P-frame motion vectors to an estimate of the true motion present in the video scene. We accomplish this by first decompressing the video in question, then performing motion estimation on the video to obtain a new set of row and column motion vectors. When estimating the true motion of the video, we use an exhaustive search to determine each motion vector. We note that care must be taken to ensure that each frame is predicted from the same anchor frame used by the compressed video.

Let $r_{u,v}(n)$ and $c_{u,v}(n)$ denote the stored row and column motion vectors at macroblock location (u,v) in the nth P-frame of a compressed video whose authenticity is questioned. Similarly, let $\hat{r}_{u,v}(n)$ and $\hat{c}_{u,v}(n)$ denote the row and column motion vectors estimated from the decompressed video. We compute the mean squared Euclidean distance d between the stored and estimated motion vectors at each frame as

$$d(n) = \frac{1}{UV} \sum_{u=1}^{U} \sum_{v=1}^{V} (r_{u,v}(n) - \hat{r}_{u,v}(n))^2 + (c_{u,v}(n) - \hat{c}_{u,v}(n))^2, \tag{17}$$

where U and V are the number of row and column macroblocks in each video frame.

Since not every frame requires anti-forensic modification to raise its error level to the anti-forensic target level, some frames will have distinctly larger $d(n)$ values than others. As a result, a signal similar to the fingerprint signal $s(n)$ occurs in $d(n)$ for anti-forensically modified videos. We exploit this information by measuring the strength of this periodic signal as $d_{\text{freq}} = D(k^*)$ where $D(k) = \text{DFT}\{d(n)\}$, and $k^* = N/T$ as defined in Section III. Additionally we obtain a measure $d_{\text{mean}} = (1/N)\sum_{n=1}^{N} d(n)$ of the mean $d(n)$ value. We combine both of these features into a feature vector $\mathbf{d} = [d_{\text{mean}} d_{\text{freq}}]$, then use principal component analysis to reduce its dimensionality to a one dimensional feature d_α.

Framing the detection of frame deletion anti-forensics as a hypothesis testing problem, we adopt the convention that the null hypothesis (H_0) is that the video has not undergone anti-forensic modification and the alternative hypothesis (H_1) is that the video has been anti-forensically modified. We detect the use of frame deletion anti-forensics using the following decision rule:

$$\delta_{mv} = \begin{cases} H_0 & \text{if } d_\alpha < \tau_{mv} \\ H_1 & \text{if } d_\alpha \geq \tau_{mv}, \end{cases} \tag{18}$$

where τ_{mv} is the decision threshold.

If a forensic investigator is aware of the possibility that anti-forensics have been used, we must assume that a digital forger will be aware of techniques designed to detect their use of anti-forensics. Since we detect the use of frame deletion anti-forensics by analyzing a video's motion vectors, an intelligent forger will modify their anti-forensic algorithm in an attempt to minimize the mean squared Euclidean distance between the anti-forensically modified motion vectors and the true scene motion.

Because the mean squared Euclidean distance is used to compare a video's motion vectors to an estimate of its true motion, large differences between the anti-forensically modified motion vectors will be penalized more than small differences. This is reasonable because while small errors might realistically occur during motion estimation, large motion estimation errors are far less likely. Naively setting the motion vectors of several macroblocks to zero has the potential to create sizable disparities between these motion vectors and the macroblock's true motion. If the target prediction error can be achieved by introducing small changes to a large set of motion vectors rather than large changes to a small set, the mean squared Euclidean distance between the anti-forensically modified motion vectors and the true motion will be reduced. This will correspondingly decrease the probability that the use of anti-forensics is detected. In light of this, we perform the following modifications to our proposed anti-forensic technique.

Rather than increasing a P-frame's prediction error by setting several of its motion vectors to zero, we instead fix a search radius with an initial value of one pixel around each true motion vector. We then search the set of motion vectors lying inside these search radii for the set of motion vectors that maximize the total prediction error. If the target prediction error is not achievable using motion vectors within the current search radius, the search radius is incremented by one pixel and the search is repeated. This process is iterated until the target prediction error is achievable at a particular search radius.

Once the appropriate radius is determined, the new anti-forensic motion vectors and prediction errors are determined using a process similar to that propose in Section IV. The only modification required is that we change $\hat{b}_{i,j}(n)$ to be the sum of the absolute value of a macroblock's prediction error when that macroblock's motion vectors are anti-forensically obtained using the final search radius. Similarly, we change $b_{i,j}(n)$ to the macroblock's total prediction error when the macroblock's motion vectors are anti-forensically determined using the final search radius minus one.

VI. Performance Analysis and Trade-Off

While digital forensic techniques have been studied for roughly a decade, anti-forensic techniques are relatively new. Presently, few tools exist to evaluate the performance of anti-forensic techniques. Still fewer tools exist to understand the optimal set of actions of a forensic investigator and forger when the forger's use of anti-forensics can be detected. In this section, we propose new techniques for evaluating the performance of an anti-forensic operation. Additionally, we propose a game theoretic framework for analyzing the interplay between a forensic investigator and a forger [24].

A. Evaluating the Performance of Anti-Forensic Techniques

Let ψ be a digital multimedia file and $m(\cdot)$ be an editing operation capable of manipulating ψ. In order to verify the authenticity of ψ, a forensic investigator will attempt to determine if ψ is actually a manipulated version of another, unaltered digital multimedia file ψ'. This forensic manipulation detection problem can be formulated as differentiating between the following two hypotheses:

$$\begin{aligned} H_0 &: \psi \neq m(\psi'), \\ H_1 &: \psi = m(\psi'). \end{aligned} \tag{19}$$

To identify the correct hypothesis, the forensic investigator will employ a detection algorithm δ_m designed to detect the use of m by measuring the strength of its fingerprints in ψ. Typically, this is done by calculating a detection statistic and comparing it to a decision threshold. The decision threshold is chosen to maximize the detection algorithm's probability of detection, defined as $P_d(\delta_m) = P(\delta_m = H_1 | \psi = m(\psi'))$, without violating a constraint on its probability of false alarm, defined as $P_{\text{fa}}(\delta_m) = P(\delta_m = H_1 | \psi \neq m(\psi'))$. We adopt the convention that $\delta_m^{(P_{\text{fa}})}$ specifies the detection algorithm δ_m operating using the decision threshold associated with the false alarm rate P_{fa}.

Once a detection technique has been established, a digital forger can create an anti-forensic technique α_m designed to fool δ_m. In the past, the performance of an anti-forensic technique has often been measured by the probability that δ_m will identify a multimedia file as unaltered when it has actually been edited using m then anti-forensically manipulated using α_m, i.e., $P(\delta_m(\alpha_m(\psi)) = H_0 | \psi = m(\psi'))$. This measure is biased, however, because a file altered using m will not be identified as manipulated with probability $1 - P_d$ even if anti-forensics are not used. As a result, this measure unfairly credits a number of missed detections to the effects of anti-forensics, thus overestimating the performance of α_m.

Instead, the performance of an anti-forensic technique should be measured in terms of its *anti-forensic effectiveness*, or its ability to cause the missed detection of an altered multimedia file given that the manipulation is detectable if anti-forensics is not used. As a result, we define the probability of anti-forensic effectiveness of α_m as

$$\begin{aligned} P_{ae}(\alpha_m) = \\ P(\delta_m(\alpha_m(\psi)) = H_0 | \delta_m(\psi) = H_1, \psi = m(\psi')). \end{aligned} \tag{20}$$

It is important to note, however, that an anti-forensic operation need not achieve a $P_{ae} = 1$ in order to render δ_m ineffective. In fact, α_m only needs to cause δ_m to miss a sufficient number of detections for its performance to become equivalent to making a random decision, or in other words $P_d(\delta_m^{(P_{\text{fa}})}) = P_{\text{fa}}$. In light of this, it is important to measure the degree to which a forensic technique is susceptible to an anti-forensic attack. As a result, we define the *anti-forensic susceptibility* of a forensic detection

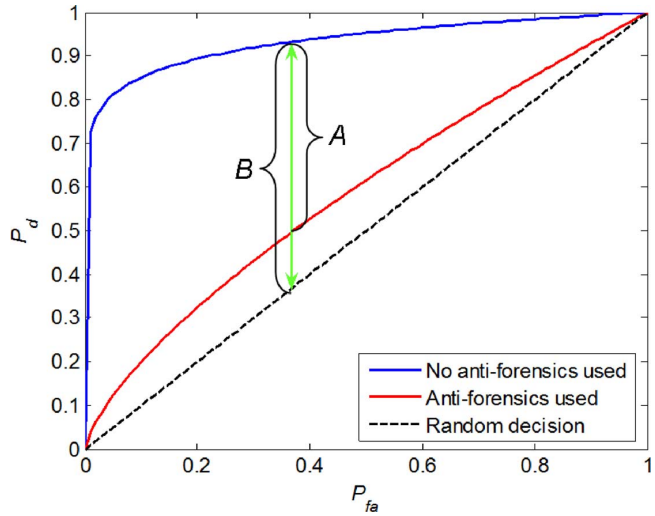

Fig. 3. Example relating the anti-forensic effectiveness of an anti-forensic operation to the ROC curves achieved by a forensic technique when anti-forensics is and is not used. The anti-forensic effectiveness at a given false alarm level is the ratio A/B.

technique δ_m operating with a false alarm constraint of P_{fa} to an anti-forensic attack α_m as

$$S_\alpha(\delta_m, P_{\text{fa}}) = \frac{P_d\left(\delta_m^{(P_{\text{fa}})}\right) - \max\left(P_d\left(\delta_m^{(P_{\text{fa}})}\right)(1 - P_{ae}(\alpha_m)), P_{\text{fa}}\right)}{P_d\left(\delta_m^{(P_{\text{fa}})}\right) - P_{\text{fa}}}. \tag{21}$$

At a particular false alarm level, the numerator of S_α is the difference between the probability that δ_m will detect manipulation if anti-forensics is not used and the probability that δ_m will detect manipulation if anti-forensics is used to disguise manipulation fingerprints. More explicitly, it is the decrease in the performance of δ_m due to the use of α_m, as shown by the distance A in Fig. 3. When computing this distance, we take the maximum between probability that δ_m will detect manipulation if anti-forensics is used, i.e., $P_d(\delta_m^{(P_{\text{fa}})})(1 - P_{ae}(\alpha_m))$, and the probability of false alarm because the forensic investigator can always achieve $P_d = P_{\text{fa}}$ by randomly deciding that a multimedia file is manipulated with probability P_{fa}. Any decrease in the performance of δ_m beyond this point is unnecessary to render δ_m ineffective.

To normalize S_α, its denominator is the difference between the probability of detection achieved by $\delta_m^{(P_{\text{fa}})}$ and its corresponding false alarm rate. This difference, which corresponds to the distance B shown in Fig. 3, is the maximum decrease in the performance of the forensic detection technique that an anti-forensic attack can cause. As a result, the anti-forensic susceptibility is a measure between 0 and 1 of the decrease in the effectiveness of δ_m caused by α_m. An anti-forensic susceptibility of one indicates that α_m is able to cause δ_m to perform no better than a random decision, while an anti-forensic susceptibility of zero signifies that α_m is unable to cause any reduction in the performance of δ_m. We note that S_α is undefined for $P_{\text{fa}} = 1$ because under this condition, no anti-forensic technique is able to cause any reduction in the performance of the forensic detector (it will always decide that the file has been manipulated).

B. Analysis the Interplay Between a Forger and Forensic Investigator Using Game Theory

In many instances, an anti-forensic operation will leave behind forensically detectable fingerprints of its own. If this is the case, a new forensic detection technique δ_α can be designed to detect the use of α_m. Under this scenario, the forensic detector must determine whether a digital multimedia file is a manipulated and anti-forensically modified version of another unaltered file or not. This problem can be framed as a hypothesis test by defining the two hypotheses as

$$\begin{aligned} H_{0\alpha} &: \psi \neq \alpha_m(m(\psi')), \\ H_{1\alpha} &: \psi = \alpha_m(m(\psi')). \end{aligned} \tag{22}$$

To avoid confusion, we rename the previous hypotheses used in the manipulation detection scenario as H_{0m} and H_{1m}. By formulating the detection of anti-forensic manipulation in this manner, the performance of δ_α can be measured using the probabilities of detection and false alarm as before.

The existence of a detection technique capable of identifying the use of anti-forensics poses a new problem for a forger: should anti-forensics be used to disguise a forgery if the use of anti-forensics can itself be detected? A multimedia file will be identified as forged if either manipulation or the use of anti-forensics is detected, therefore a forger must attempt to hide evidence of both. In response, the forger may design their anti-forensic operation in such a way that the strength with which it is applied can be adjusted. By reducing the strength of their anti-forensic attack, a forger decreases the strength of fingerprints left by anti-forensics and correspondingly decreases the probability that their use of anti-forensics will be detected. This is not without a cost, however, because as the strength with which anti-forensics is applied is decreased, the strength of manipulation fingerprints remaining in a multimedia file will increase. This will correspond to an increase in the probability that manipulation will be detected. As a result, the forger must identify the strength with which to apply their anti-forensic operation that minimizes the probability that either their manipulation of the multimedia file or their use of anti-forensics will be detected.

Additionally, some anti-forensic operations degrade the quality of the digital multimedia file that they are used on. If this occurs, it is possible that a human inspecting a forgery may be able to perceptually identify the forgery even if it does not contain detectable manipulation or anti-forensic fingerprints. Alternately, a human may not be able to identify that the multimedia file has been forged, but the perceptual quality of the forgery may be so low that it is rendered useless. In these cases, the forger must also take the perceptual quality of their forgery into account when choosing the appropriate anti-forensic strength.

It is fairly obvious that the optimal anti-forensic strength for the forger to use depends on the decision thresholds used by δ_m and δ_α. Consequently, a forensic detector will choose the decision thresholds for both δ_m and δ_α that maximize

their probability of detecting a forgery. Typically, however, a forensic investigator is not free to choose any set of decision thresholds because they must operate in accordance with some false alarm constraint. Since the probabilities of false alarms associated with both δ_m and δ_α contribute to the total probability of false alarm, the forensic detector must decide how much to allow each detection technique to contribute to the total probability of false alarm. This implies that the probability of false alarm allocation that maximizes the forensic detector's probability of detecting a forgery depends on the anti-forensic strength used by the forger. As a result, both the forger and forensic investigator's optimal actions depend on the actions of their counterpart.

The dependence of the forensic investigator's and forger's optimal actions on the actions of the other naturally leads to the following question; is there a set of actions (i.e., anti-forensic strength and probability of false alarm allocation) for the both the forger and forensic investigator that neither have any incentive to deviate from? Furthermore, if this set of actions exists and both parties take these actions, what is the probability that a forgery will be detected? To answer these questions we use game theory to evaluate the dynamics between the forensic investigator and the forger.

To formulate this situation as a game, we let player 1 denote the forensic investigator and player 2 denote the forger. We adopt the convention that player 1 moves first, or in other words, the forensic investigator chooses their probability of false alarm allocation and corresponding decision thresholds first, then allows the forger to respond. Given a total probability of false alarm constraint ξ, the set of strategies that the forensic investigator can employ is the set of false alarm levels $\eta \in [0, \xi]$ that can be allocated to δ_m. The corresponding false alarm level $\tilde{\eta}$ allocated to δ_α is the maximum false alarm level such that

$$\begin{aligned} P_{\text{fa}}^{\text{Tot}} &= P\left(\delta_m^{(\eta)}(\psi) = H_{1m} \bigcup \right. \\ &\qquad \left. \delta_\alpha^{(\tilde{\eta})}(\psi) = H_{1\alpha} \middle| \psi \neq m(\psi'), \psi \neq \alpha_m(m(\psi'))\right) \\ &\leq \xi. \end{aligned} \tag{23}$$

Let $\alpha_m^{(k)}$ be an anti-forensic operation operating at strength $k \in [0, 1]$, where $k = 1$ corresponds to using anti-forensics at full strength and $k = 0$ is equivalent to not using anti-forensics at all. The set of strategies that the forger can employ is the set of anti-forensic strengths $k \in [0, 1]$.

For a particular pairing of strategies (η, k), the utility of player 1 is the probability that either manipulation or the use of anti-forensics will be detected, i.e.,

$$\begin{aligned} U_1(k, \eta) = P\left(\delta_m^{(\eta)}(\psi) = H_{1m} \bigcup \right. \\ \left. \delta_\alpha^{(\tilde{\eta})}\left(\alpha_m^{(k)}(\psi)\right) = H_{1\alpha} | \psi = m(\psi')\right). \end{aligned} \tag{24}$$

Because this corresponds to the probability that a forgery will be detected, player 1 wishes to maximize this utility. By contrast, player 2 wishes to minimize this quantity along with some measure $\gamma(m(\psi), \alpha_m^{(k)}(m(\psi)))$ of the perceptual distortion introduced into their forgery by the use of anti-forensics. As a result, the utility of player 2 is

$$U_2(k, \eta) = -U_1(k, \eta) - \gamma\left(m(\psi), \alpha_m^{(k)}(m(\psi))\right). \tag{25}$$

By substituting in the appropriate expressions for the probabilistic quantities in each utility function, we can find the Nash equilibrium strategies (η^*, k^*) that neither player has an incentive to deviate from. In practice, however, the analytical evaluation of these utilities is often difficult or impossible. In many forensic scenarios, no known equation exists to express the probabilistic quantities used in each utility function. As a result, the Nash equilibria must often be sought out numerically.

Once the Nash equilibrium strategies have been identified, we can evaluate the probability that the forensic investigator will detect a forgery. To do this, we simply need to evaluate $U_1(\eta^*, k^*)$ because this probability is the utility of player 1. Since the strategy of player 1 is influenced by the false alarm constraint ξ placed on the forensic investigator, it is possible that different Nash equilibrium strategies and different probabilities of forgery detection will be achieved at different ξ levels. By varying ξ between 0 and 1, we can determine the probability of detecting a forgery at the Nash equilibrium associated with each ξ value. Using this information, we can construct a receiver operating characteristic (ROC) curve that displays the forensic investigator's ability to detect a forgery at each false alarm level if both players act rationally. We call this ROC curve the Nash equilibrium receiver operating characteristic (NE ROC) curve. It is this curve, rather than the individual ROC curves of each forensic detection technique, that most accurately characterizes a forensic investigator's ability to detect a digital forgery.

VII. Experiments and Results

We conducted a series of experiments to evaluate the performance of each of our proposed forensic and anti-forensic techniques. In order to create data suitable for our experiments, we compiled a set of 36 standard video test sequences in the QCIF format (i.e., a frame size of 176 × 144 pixels). A complete list of the names of these sequences, along with information regarding where these sequences can be downloaded, is provided in the Appendix. Because these sequences are distributed in an unaltered and uncompressed state, we were able to completely control each video's processing history and ensure that no fingerprints left by other signal processing operations affected our experimental results.

Next, we simulated motion compensated video compression and decompression in Matlab. In this implementation, we used a fixed twelve frame GOP structure *IBBPBBPBBPBB* along with standard MPEG DCT coefficient quantization tables. During motion estimation, we determined the set of motion vectors for a predicted frame using an exhaustive search. We then compressed the first 250 frames of each uncompressed video sequence, creating a database of unaltered videos compressed using a fixed GOP length.

Because newer compression schemes allow the GOP structure to vary during encoding, we modified our encoder so

that it randomly chose between the GOP structures *IBBPBB*, *IBBPBBPBB*, and *IBBPBBPBBPBB* for each GOP during encoding. By allowing the encoder to use variable GOP lengths, we were able to simulate the forensically significant manner in which newer codecs differ from older codecs. We compressed the first 250 frames of each of the uncompressed video sequences using this variable GOP length encoder, creating a second database of unaltered videos. Frame deletion experiments run on these videos were used to simulate the aperiodic frame deletion fingerprints introduced by newer video compression techniques.

A. Frame Deletion Detection

To test the forensic effectiveness of our proposed frame deletion detectors, we first created a database of forged videos. To do this, we deleted 3, 6, and 9 frames from the beginning of each unaltered video sequence compressed using a fixed length GOP, then recompressed each video. This corresponded to removing 1/4, 1/2, and 3/4 of a GOP respectively. To test against frame addition, we added 6 frames to the beginning of each unaltered video sequence compressed with a fixed length GOP, then recompressed these videos. Additionally, we deleted 6 frames from the videos compressed using randomly varying GOP lengths. We then used each of our proposed detection techniques to determine if frame deletion or addition had occurred in each video.

When testing for frame deletion or addition, we varied the value of the decision threshold used in each detector over a range of values. The probabilities of detection P_d and false alarm P_{fa} were determined for each threshold by respectively calculating the percentage of forged videos that were correctly classified and the percentage of unaltered videos that were incorrectly classified. We used these results to generate the series of ROC curves for δ_{fixed} shown in Fig. 4 and for δ_{var} shown in Fig. 5. We can see from these ROC curves that both detectors' performance remains consistent regardless of the number of frames deleted. Furthermore, we can see that frame addition can be detected with the same accuracy as frame deletion. By examining the ROC curves for each detector corresponding to the average performance across all frame deletion amounts, we can see that both detectors were able to achieve at a P_d of at least 85% at a false alarm rate less than 5%. Both detectors also achieved a P_d of at least 90% at a false alarm rate less than 10%. These results indicate that both detectors can be used to reliably detect frame deletion. Additionally, results presented in Fig. 5 suggest that δ_{var} can be used to detect frame deletion in videos compressed using randomly varying GOP lengths as well.

B. Frame Deletion Anti-Forensics

To evaluate the performance of our proposed frame deletion anti-forensic technique, we deleted six frames from each unaltered video compressed using a fixed GOP structure, then recompressed each video while applying our anti-forensic technique. When implementing our anti-forensic technique, we incorporated the modifications to our algorithm discussed in Section V.

An example of typical results achieved by our proposed anti-forensic technique is shown in Fig. 6. This figure displays the

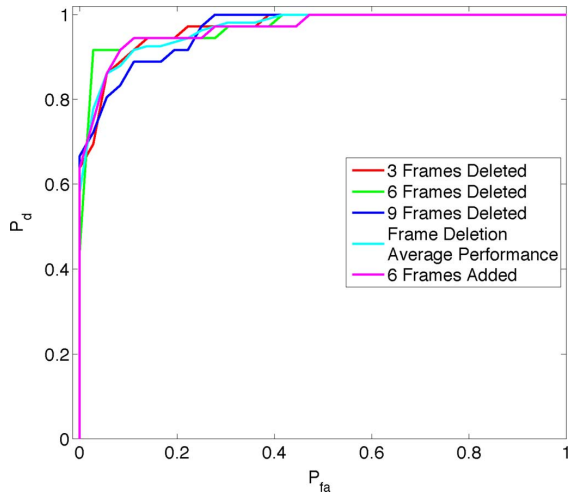

Fig. 4. ROC curves for δ_{fixed} obtained by testing against different amounts frame deletion and addition.

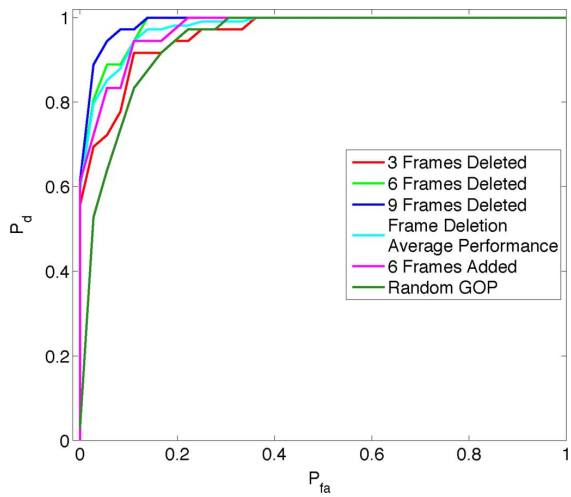

Fig. 5. ROC curves for δ_{var} obtained by testing against different amounts frame deletion and addition.

P-frame prediction error sequence taken from an untampered version of the 'Foreman' video compressed using a fixed GOP length, as well as the P-frame prediction error sequences obtained after deleting the first six frames then recompressing the video with and without applying our anti-forensic temporal fingerprint removal technique. The temporal fingerprint features prominently in the prediction error sequence of the video in which frames are deleted without the use of our anti-forensic technique, particularly in the frequency domain. By contrast, these fingerprints are absent from the prediction error sequence when our anti-forensic technique is used to hide evidence of frame deletion.

Next, we examined the ability of our proposed anti-forensic technique to fool each of our automatic frame deletion detection techniques. To do this, we used both of our proposed detection techniques to classify each video in our databases of unaltered and anti-forensically modified videos as unaltered or one from which frames had been deleted. This was done using a series of different decision thresholds, then the probabilities of detection and false alarm corresponding to each decision threshold were calculated from the results. We used this data to generate a new set of ROC curves for δ_{fixed} and δ_{var} when frame deletion has

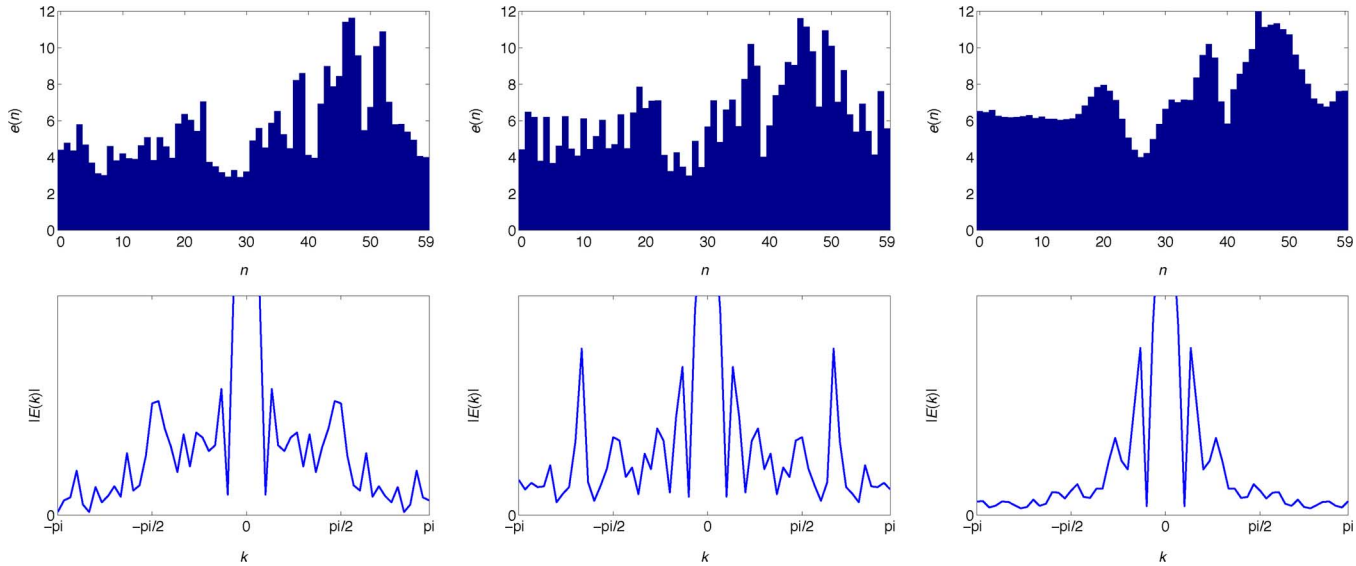

Fig. 6. P-frame prediction error sequences (top row) and the magnitudes of their respective DFTs (bottom row) obtained from an untampered compressed version of the "Foreman" video (left column), as well as from the same video after the first six frames were deleted followed by recompression without anti-forensic modification (middle column) and with the use of our proposed anti-forensic technique (right column).

been disguised using anti-forensics. These ROC curves are displayed in Fig. 7(a).

In this figure, the dashed line represents the performance of a decision rule that randomly classifies a video as forged with a probability equal to P_{fa}. Reducing a detection technique's performance to this level corresponds to making it equivalent to a random guess. As we can see from Fig. 7(a), both frame deletion detection techniques perform at or near this level when our anti-forensic technique is applied to a video. Similarly, we used this data to compute the anti-forensic susceptibility of each detector to our proposed anti-forensic frame deletion technique. These results, which are displayed in Fig. 7(b), show that the detector δ_{var} is entirely susceptible to our anti-forensic technique at all false alarm levels. The detector δ_{fixed} was slightly less susceptible to our anti-forensic attack, however, our anti-forensic technique achieved an anti-forensic susceptibility of .7 or greater for all $P_{\text{fa}} \leq 80\%$ for this detector. These results demonstrate that our proposed anti-forensic technique is able to render forensic frame deletion detection techniques nearly completely ineffective.

C. Detecting Frame Deletion Anti-Forensics

In order to evaluate the performance of our technique designed to detect the use of frame deletion anti-forensics, we reexamined the videos in our database of unaltered and anti-forensically modified videos compressed using a fixed GOP structure. We used our proposed detector δ_{mv} to classify each video as unmodified or anti-forensically modified at a variety of different decision thresholds, then used these results to generate the ROC curve shown in Fig. 8.

The results of this experiment show that our proposed detector achieved perfect detection (i.e., a P_d of 100% at a P_{fa} of 0%). These results are slightly misleading, however, because the motion vectors of the videos in the unaltered database are obtained using an exhaustive search. Since an exhaustive search is

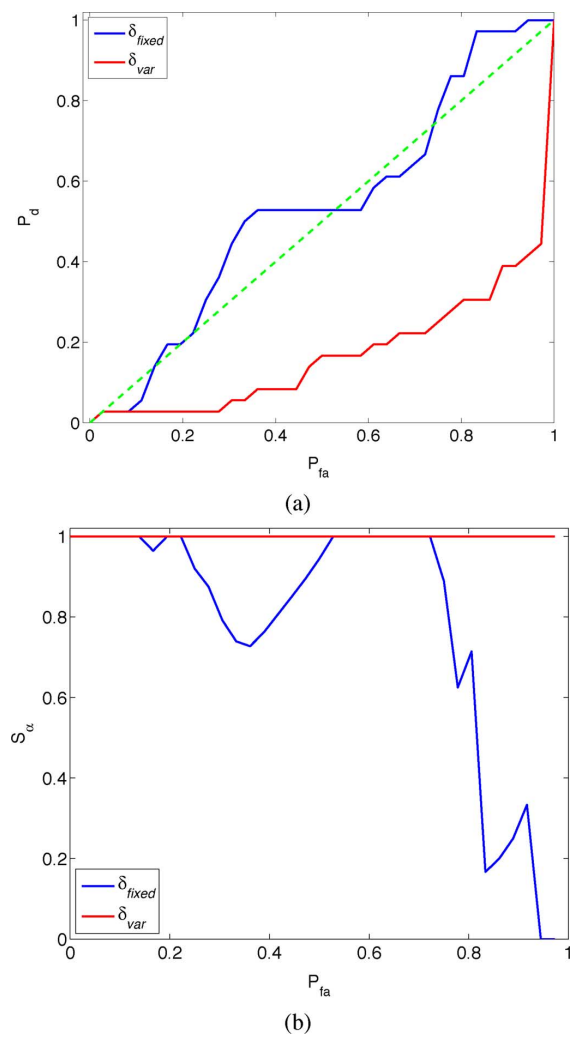

Fig. 7. Experimental results showing (a) ROC curves for δ_{fixed} and δ_{var} and (b) anti-forensic susceptibility plots for δ_{fixed} and δ_{var} obtained by testing on anti-forensically modified videos compressed using a fixed GOP structure.

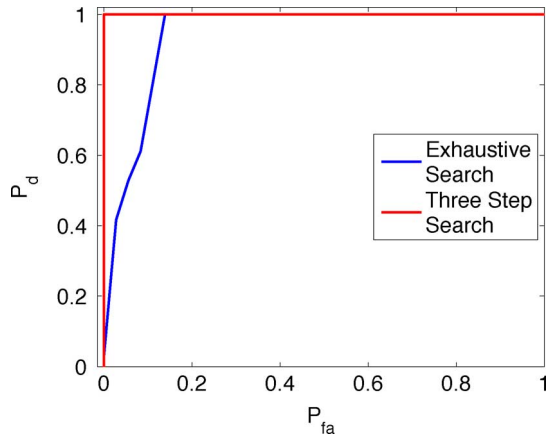

Fig. 8. ROC curves for the anti-forensics detector δ_{mv} when tested on video data compressed using an exhaustive search to determine motion vectors and video data encoded using a three step motion vector search algorithm.

also used when estimating a video's true motion during the detection of anti-forensics, there will be very little difference between an unaltered video's stored and recalculated motion vectors.

In reality, many video encoders use efficient algorithms to perform motion estimation. These algorithms greatly reduce the time needed to encode a video and produce a near optimal set of motion vectors. Nonetheless, the motion vectors obtained using these algorithms differ slightly from those obtained using an exhaustive search. As a result, it is more difficult to differentiate between an anti-forensically modified video and an unaltered video if one of these algorithms is used during encoding.

To evaluate the performance of our proposed frame deletion anti-forensics detection technique under less favorable conditions, we modified our video coder to perform motion estimation using the three step search algorithm proposed by Zhu and Ma [25]. We then created a new database of compressed unaltered videos whose motion vectors were obtained using this efficient search algorithm. We repeated the previous experiment with this data and used the results to generate the ROC curve shown in Fig. 8.

We can see from Fig. 8 that the performance of our proposed detector is degraded in this scenario. While the detection of frame deletion anti-forensics can still be performed, it must be done with a higher false alarm rate. This suggests that if a forensic investigator's maximum acceptable false alarm rate is sufficiently low, a video forger using anti-forensics is likely to avoid detection. To mitigate this, a forensic investigator may wish to repeat frame deletion anti-forensics detection using a decision threshold corresponding to a higher false alarm rate, but not immediately assume that detections correspond to forged videos. Instead, these videos can be flagged for closer investigation using additional forensic techniques.

D. Game Theoretic Evaluation of Video Forensics and Anti-Forensics

Once we evaluated the performance of each proposed forensic detection technique as well as the proposed video frame deletion anti-forensic technique, we used our game theoretic framework to identify the optimal strategies of both the forensic investigator and video forger. To do this, we modified our frame deletion anti-forensic technique to operate at variable strengths. This was accomplished by choosing the target P-frame prediction error sequence associated with strength k as

$$\tilde{e}_k(n) = k\tilde{e}(n) + (1-k)e(n), \tag{26}$$

where $\tilde{e}(n)$ denotes the fingerprint-free target prediction error sequence described in Section IV.

Because our proposed anti-forensic technique introduces virtually no distortion into a forged video, we set the term $\gamma(\,\cdot\,) = 0$ in the utility function of player 2. As a result, $U_2(k,\eta) = -U_1(k,\eta)$ causing our video forensic scenario to reduce to a zero sum game. This allowed us to find the Nash equilibrium strategies by solving the following equation

$$(k^*, \eta^*) = \arg \max_{\eta} \min_{k} U_1(k,\eta). \tag{27}$$

Since no closed form expression for $U_1(k,\eta)$ exists in this scenario, we evaluated (27) numerically. This was done by first deleting frames from each single compressed video in our database, then anti-forensically modifying each video with strengths ranging between 0 and 1. For each anti-forensically modified video, we performed frame deletion detection and anti-forensics detection using a variety of different decision thresholds then calculated the P_{fa} and P_d associated with each decision threshold and anti-forensic strength pairing. Using this data, the Nash equilibrium strategies and probability of forgery detection were calculated.

Fig. 9 shows the utility function $U_1(k,\eta)$ when the forensic investigator operates under the false alarm constraint $P_{\text{fa}}^{\text{Tot}} = 8.3\%$. Under this condition, the Nash equilibrium strategy is $k = 0.4$, $\eta = 0.0$, which corresponds to the forger reducing the strength of their anti-forensic attack to half and the forensic investigator allowing all of their false alarms to come from the anti-forensic detector δ_{mv}. The probability with which the forensic investigator will detect a forgery, i.e., the value of $U_1(k^*,\eta^*)$, is 38.9%. We note that it is less than the probability of detection achieved by both the frame deletion detector and the anti-forensics detector at the same false alarm level. This reinforces the notion that the forger can create a more successful anti-forensic attack by decreasing its strength.

We determined the Nash equilibrium strategies and calculated the probability of forgery detection for a set of total probability of false alarm constraints between 0% and 100%. We used these results to create the NE ROC curve displayed in Fig. 10. From this curve we can see that if the forensic investigator must operate with a total probability of false alarm constraint of 10% or less, frame deletion forgeries are difficult to detect. If the forensic examiner is able to relax their probability of false alarm constraint to roughly 15% or greater, then they will be able to detect frame deletion forgeries at a rate of at least 85%.

Table I shows the Nash equilibrium strategies for a variety of total probability of false alarm levels ξ. In some cases, multiple values of k are Nash equilibrium strategies for a particular value of ξ. We note that here, the value of U_1 corresponding to each Nash equilibrium strategy at a particular ξ value is the same. From the data presented in this table, we can observe two trends. The first is that as the false alarm constraint increases, the optimal strategy for the forger is to decrease the strength with

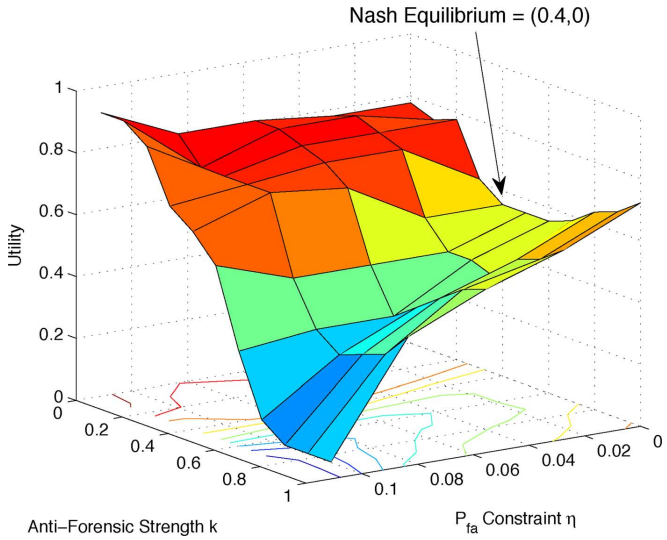

Fig. 9. Utility function of the forensic investigator $U_1(k, \eta)$ when the total probability of false alarm constraint is $P_{\text{fa}}^{\text{Tot}} = 8.3\%$.

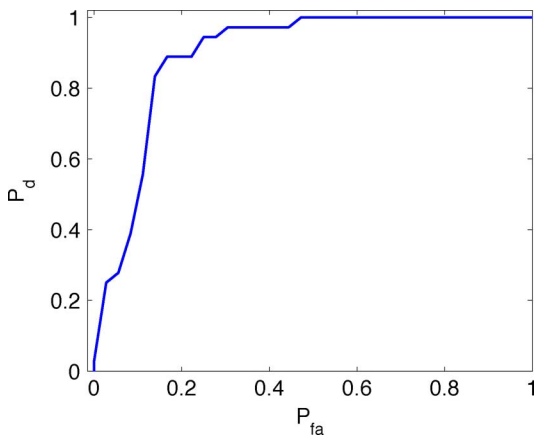

Fig. 10. Nash equilibrium ROC curve for video frame deletion detection.

TABLE I
NASH EQUILIBRIUM STRATEGIES k^* AND η^* OBTAINED FOR THE FORGER AND FORENSIC INVESTIGATOR RESPECTIVELY AT DIFFERENT CONSTRAINTS ξ ON THE FORENSIC INVESTIGATOR'S TOTAL PROBABILITY OF FALSE ALARM

ξ	k^*	η^*
0.0%	0.6, 0.7, 0.8, 0.9, 1.0	*
2.8%	0.6	0.0%
5.6%	0.7	0.0%
8.3%	0.4	0.0%
11.1%	0.4, 0.5, 0.6	0.0%
13.9%	0.1, 0.2, 0.4	0.0%
16.7%	0.1, 0.2, 0.3, 0.5	0.0%
...	...	...

which they apply anti-forensics. The second is that regardless of the value of ξ, the optimal strategy for the forensic investigator is to allow all of their false alarm contributions to come from the anti-forensics detector δ_{mv}. This is because the effectiveness of the frame deletion detection technique drops off quickly as k is increased. By contrast, the anti-forensics detector can still operate effectively even at low anti-forensic strengths. As a result, it is in the best interest of the forensic investigator to place the maximum load on δ_{mv}.

VIII. CONCLUSION

In this paper, we have proposed a set of automatic frame deletion or addition detection techniques that operate by identifying increases in a video's P-frame prediction error that correspond to frame deletion or addition fingerprints. To do this, we first developed a model of a video's P-frame prediction error sequence before and after frame deletion or addition has occurred. Additionally, we used this model to design an anti-forensic technique capable of removing frame deletion or addition fingerprints form a forged video. This technique operates by first constructing a target prediction error sequence free from frame deletion or addition fingerprints, then modifying the motion vectors of each P-frame so that its total absolute prediction error matches the target value. Furthermore, we have proposed a forensic technique to detect the use of frame addition or deletion anti-forensics by comparing a compressed video's motion vectors to an estimate of the true motion in the video.

Through a series of simulations and experiments, we have evaluated the performance of each of our proposed forensic and anti-forensic techniques. Our results show that both of our proposed frame deletion or addition detection techniques can automatically detect video forgeries with a high degree of accuracy if anti-forensics is not used. These results also show that our proposed anti-forensic frame deletion or addition technique can successfully fool both forensic techniques. If this technique is applied at full strength, however, our anti-forensics detector is able to identify that anti-forensics has been used with a high degree of accuracy.

In addition, we have proposed a set of methods to evaluate the performance of anti-forensic techniques. Furthermore, we have proposed a game theoretic framework that can be used to understand the interplay between a forensic detector and a forger when the forger's use of anti-forensics can be detected. This framework allows us to identify the optimal set of actions for both the forensic investigator and forger, as well as to evaluate the ability of the forensic investigator to identify forgeries. We have applied our game theoretic framework to the video frame deletion or addition forgery scenario and identified the optimal strategies for a forensic investigator and video forger to employ. These results show that as the forensic investigator's probability of false alarm constraint is increased, the strength with which the forger should apply anti-forensics is decreased. By contrast, the forensic investigator should allow their video frame addition or deletion detector to operate at a P_{fa} of 0% and allow all of their false alarms to come from their anti-forensics detector, regardless of the constraint on their total probability of false alarm. Furthermore, we have found that if the forensic investigator is bound by a total probability of false alarm constraint of approximately 10% or less, the forensic investigator will have less than a 50% chance of detecting a video forgery. If the total probability of false alarm constraint is above 15%, video forgeries can be detected at a rate of 85% or greater.

APPENDIX

The following video sequences were used to perform the experiments in this paper: *Akiyo, Bowing, Bridge-Close, Bridge-Far, Carphone, City, Claire, Coastguard, Container,*

Crew, Deadline, Flower Garden, Football, Foreman, Galleon, Grandma, Hall, Harbour, Highway, Husky, Intros, Pamphlet, Mobile, Mother and Daughter, News, Paris, Salesman, Sign-Irene, Silent, Soccer, Stefan, Students, Table, Tempete, Vtc1nw, WashDC.

These videos were obtained from the online video databases:

- http://trace.eas.asu.edu/yuv/
- http://media.xiph.org/video/derf/
- ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/

REFERENCES

[1] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Int. Workshop Information Hiding*, Toronto, Canada, 2004, pp. 128–147.

[2] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[3] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[4] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A classifier design for detecting image manipulations," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, vol. 4, pp. 2645–2648.

[5] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 492–506, Sep. 2010.

[6] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proc. ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006, pp. 37–47.

[7] W. Wang and H. Farid, "Exposing digital forgeries in interlaced and de-interlaced video," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 438–449, Jun. 2007.

[8] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source digital camcorder identification using sensor photo response non-uniformity," in *Proc. SPIE Electronic Imaging, Photonics West*, Feb. 2007, vol. 6505.

[9] C. Kraetzer, A. Oermann, J. Dittmann, and A. Lang, "Digital audio forensics: A first practical evaluation on microphone and environment classification," in *Proc. 9th Workshop on Multimedia and Security*, New York, 2007, pp. 63–74, ACM.

[10] D. Garcia-Romero and C. Y. Espy-Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Mar. 2010, pp. 1806–1809.

[11] C. Grigoras, "Digital audio recording analysis: The electric network frequency (ENF) criterion," *Int. J. Speech Language and the Law*, vol. 12, no. 1, pp. 63–76, Mar. 2005.

[12] M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 582–592, Dec. 2008.

[13] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 78–86.

[14] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Mar. 2010, pp. 1694–1697.

[15] M. C. Stamm and K. J. R. Liu, "Wavelet-based image compression anti-forensics," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2010, pp. 1737–1740.

[16] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1050–1065, Sep. 2011.

[17] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Undetectable image tampering through JPEG compression anti-forensics," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2010, pp. 2109–2112.

[18] G. Cao, Y. Zhao, R. Ni, and H. Tian, "Anti-forensics of contrast enhancement in digital images," in *Proc. 12th ACM Workshop on Multimedia and Security*, 2010, pp. 25–34, ACM.

[19] M. Kirchner and R. Böhme, "Synthesis of color filter array pattern in digital images," in *Proc. SPIE-IS&T Electronic Imaging: Media Forensics and Security*, Feb. 2009, vol. 7254.

[20] M. Goljan, J. Fridrich, and M. Chen, "Defending against fingerprint-copy attack in sensor-based camera identification," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 227–236, Mar. 2011.

[21] G. Valenzise, V. Nobile, M. Taglisacchi, and S. Tubaro, "Countering JPEG anti-forensics," in *Proc. IEEE Int. Conf. Image Processing*, Brussels, Belgium, Sep. 2011.

[22] T. Pevny and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 247–258, Jun. 2008.

[23] M. C. Stamm and K. J. R. Liu, "Anti-forensics for frame deletion/addition in MPEG video," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 1876–1879.

[24] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Forensics vs. anti-forensics: A decision and game theoretic framework," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012, pp. 1749–1752.

[25] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

Matthew C. Stamm (S'08–M'11) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland, College Park in 2004, 2011, and 2012, respectively. He is currently a postdoctoral researcher in the Department of Electrical and Computer Engineering at the University of Maryland, College Park.

From 2004 until 2006, Dr. Stamm was a radar systems engineer at the Johns Hopkins University Applied Physics Laboratory. His current research interests are in information security, specifically digital multimedia forensics and anti-forensics. Dr. Stamm received a Distinguished Teaching Assistant Award in 2006, a Future Faculty Fellowship in 2010, and the Ann G. Wylie Fellowship in 2011 from the University of Maryland, College Park. For his dissertation research, he was awarded the Dean's Doctoral Research Award in 2012 from A. James Clark School of Engineering at the University of Maryland, College Park.

W. Sabrina Lin (M'06) received the Ph.D. degree from the Electrical and Computer Engineering Department, University of Maryland, College Park in 2009. She received the B.S. and M.S. degrees in electrical engineering from National Taiwan University in 2002 and 2004, respectively. Her research interests are in the area of information security and forensics, multimedia signal processing, and multimedia social network analysis. She received the University of Maryland Innovation Award in 2011 and has coauthored the book *Behavior Dynamics in Media-Sharing Social Networks* (Cambridge, U.K.: Cambridge Univ. Press, 2011).

K. J. Ray Liu (F'03) was named a Distinguished Scholar-Teacher of University of Maryland, College Park, in 2007, where he is the Christine Kim Eminent Professor of Information Technology. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of signal processing and communications with recent focus on cooperative communications, cognitive networking, social learning and networks, and information forensics and security.

Dr. Liu is the recipient of numerous honors and awards including IEEE Signal Processing Society Technical Achievement Award and Distinguished Lecturer. He also received various teaching and research recognitions from University of Maryland including the university-level Invention of the Year Award; and Poole and Kent Senior Faculty Teaching Award and Outstanding Faculty Research Award, both from A. James Clark School of Engineering. An ISI Highly Cited Author in Computer Science, Dr. Liu is a Fellow of IEEE and AAAS.

Dr. Liu is President of IEEE Signal Processing Society where he has served as Vice President—Publications and Board of Governor. He was the Editor-in-Chief of *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of *EURASIP Journal on Advances in Signal Processing*.