SELFISH COLLUDER DETECTION AND IDENTIFICATION IN TRAITORS WITHIN TRAITORS

H. Vicky Zhao and K. J. Ray Liu

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742

ABSTRACT

During collusion attacks against multimedia forensics, an important issue that colluders need to address is the fairness of the attack, i.e., whether all colluders take the same risk of being detected. Although they might agree so, some selfish colluders may break away from their fair-collusion agreement and process their fingerprinted copies before collusion to further lower their risk. On the other hand, to protect their own interests, other attackers may wish to detect and prevent such selfish pre-collusion processing. It is important to study this problem of *traitors within traitors*, formulate the dynamics among colluders and build a complete model of multi-user collusion. This paper investigates techniques that attackers can use to detect and identify selfish colluders without revealing the secrecy of any fingerprinted copies. Our simulation results show that the proposed scheme accurately identifies all selfish colluders without falsely accusing any others.

Index Terms— security, multimedia systems, video signal processing

1. INTRODUCTION

The popularity of sharing and distributing multimedia over networks has raised the critical issue of multimedia content protection and intellectual property rights enforcement. In multimedia security and forensic systems, to address the dynamics among users with different objectives, it is important to analyze users' behaviors and investigate how they interact with and respond to each other. Such investigation helps us have a thorough understanding of multimedia security and forensic systems, and enables the digital rights enforcer to offer stronger protection of multimedia.

In multi-user collusion attacks against multimedia forensics, several attackers collectively and effectively mount attacks to undermine the traitor tracing capability of multimedia fingerprinting systems. During collusion, an important issue is the fairness of the attack, i.e., whether all colluders have the same probability of being detected. Most prior work assumed that all colluders keep their agreement to share the risk during collusion and focused on the analysis of collusion strategies and effectiveness [1–4].

However, the assumption of fair play may not always hold and there might exist selfish colluders who wish to further lower their own probability of being detected. It was shown in [5] that temporal filtering of the fingerprinted copies before collusion can help selfish colluders further reduce their risk. Such pre-collusion processing makes other attackers take a much higher risk of being detected than the selfish colluders. To protect their own interests, other colluders wish to be able to detect such selfish behaviors and force all attackers to keep their fair-collusion agreement. It is important to study this problem of traitors within traitors and build a complete model of multi-user collusion, which helps improve the collusion resistance. This paper explores the possible techniques to detect and identify selfish colluders in traitors within traitors.

The rest of the paper is organized as follows. Section 2 introduces the dynamics among attackers during collusion and formulates the problem. In Section 3, we propose an algorithm to detect pre-collusion processing and identify selfish colluders and analyze its performance. Conclusions are drawn in Section 4.

The authors can be reached at hzhao and kjrliu@eng.umd.edu.

2. SYSTEM MODEL AND PROBLEM FORMULATION

2.1. Multimedia Forensic Systems

Fingerprint Embedding Spread spectrum embedding is widely used in multimedia fingerprinting due to its robustness against many attacks [6, 7]. In spread spectrum embedding, for the jth frame in the video sequence represented by a vector $\mathbf{S}_j$ of length N_j, for user $\mathbf{u}^{(i)}$ in the system, the content owner generates a unique fingerprint $\mathbf{W}_j^{(i)}$ of length N_j. The fingerprinted frame j that will be distributed to $\mathbf{u}^{(i)}$ is $\mathbf{X}_j^{(i)} = \mathbf{S}_j + JND_j \cdot \mathbf{W}_j^{(i)}$, where JND_j is the just-noticeable-difference from human visual models [6] to control the energy of the embedded fingerprints.

Multi-user Collusion During collusion, the colluders combine information from all the copies that they have and generate a new copy $\{\mathbf{V}_j\}$ in which the originally embedded fingerprints are attenuated. Recent investigation in [4] showed that, under the constraints that the colluded copies under different collusions have the same perceptual quality, the performance of nonlinear collusion attacks is similar to that of the averaging attack. Thus, we only consider the averaging based collusion attacks in this paper.

Fingerprint Detection During the fingerprint detection and colluder identification process, the detector first extracts the fingerprint $\mathbf{Y}_j$ from the jth frame $\mathbf{V}_j$ in the test copy. Then, he measures the similarity between the extracted fingerprinted $\mathbf{Y}$ and each of the original fingerprints $\{\mathbf{W}^{(i)}\}$, compares with a pre-determined threshold and outputs the estimated identities of the colluders.

2.2. Dynamics Among Attackers During Collusion

An important issue during collusion is the fairness of the attack, i.e., whether all colluders take the same risk and have equal probability of being detected. To achieve fairness of collusion, colluders should provide one another correct information about their received fingerprinted copies and adjust the collusion parameters accordingly. Most prior work assumed that all colluders keep their agreement of fair play during collusion.

In reality, there might exist some selfish colluders who wish to further lower their risk and break the fair-collusion agreement. For example, they process their fingerprinted copies before collusion and

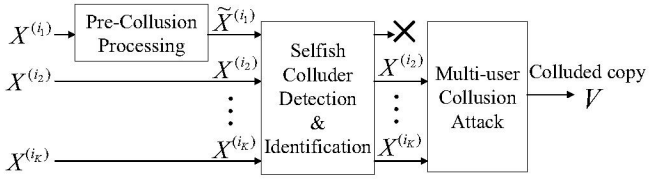


Fig. 1. The dynamics among attackers during collusion.

use the processed copies instead of the originally received ones during collusion. During pre-collusion processing, the selfish colluders select the most effective techniques to minimize their own risk of being captured. Meanwhile, to prevent others from discovering this selfish behavior and excluding them from collusion, the selfish colluders have to ensure that the processed copies are perceptually similar to the originally received ones.

On the other hand, such pre-collusion processing makes other attackers take a much higher risk of being detected than the selfish colluders. To protect their own interests, other attackers should examine all the fingerprinted copies before collusion, detect pre-collusion processing if any, and exclude those selfish colluders from collusion. The selfish colluder detection and identification process should protect the secrecy of all the fingerprinted copies. For each copy, the clear text of the fingerprinted coefficients is known to the corresponding user only, not any other attackers. Thus, proper encryption of the fingerprinted copy is required during the selfish colluder detection and identification process.

Figure 1 shows an example of this dynamics among attackers during collusion. Assume that $\mathbf{X}^{(i)}$ is the fingerprinted copy that $\mathbf{u}^{(i)}$ received from the content owner. $\mathbf{u}^{(i_1)}$ is a selfish colluder and $\mathbf{X}^{(i_1)}$ is his/her received fingerprinted copy. During pre-collusion processing, $\mathbf{u}^{(i_1)}$ generates another copy $\widetilde{\mathbf{X}}^{(i_1)}$ that is perceptually similar to $\mathbf{X}^{(i_1)}$, and tells other colluders that $\widetilde{\mathbf{X}}^{(i_1)}$ is the copy that he/she received from the content owner. Before collusion, by examining all the fingerprinted copies, $\mathbf{u}^{(i_2)}, \cdots$, and $\mathbf{u}^{(i_K)}$ find out that $\mathbf{u}^{(i_1)}$ is a selfish colluder and $\widetilde{\mathbf{X}}^{(i_1)}$ is not the copy that he/she originally received. They exclude $\mathbf{u}^{(i_1)}$ from collusion, and generate the colluded copy $\mathbf{V} = g\left(\mathbf{X}^{(i_2)}, \cdots, \mathbf{X}^{(i_K)}\right)$ where $g(\cdot)$ is the multi-user collusion function.

2.3. Problem Formulation

The existence of selfish colluders makes colluder have no trust in each other and this distrust among attackers forbids them to collude with each other. No one is willing to participate in collusion and take the risk of being detected unless he/she is assured that all others are sharing the same risk. If the attackers still wish to collude with each other and profit from the redistribution of multimedia, the attackers must share something in common that enables them to establish trust among themselves first. In this paper, we consider the scenario where there is a ringleader whom all colluders trust. All colluders believe that the trusted ringleader will not leak their fingerprinted copies to any other attackers; the ringleader himself will not frame any colluders; and the ringleader will give them the exact output of the selfish colluder detection and identification algorithm and will not modify the results.

The trusted ringleader $\mathbf{R}$ helps colluders detect and identify selfish colluders before collusion. Each colluder $\mathbf{u}^{(i)}$ first establishes a secret key $K^{(i)}$ shared with the ringleader $\mathbf{R}$ only, encrypts his/her fingerprinted copy with $K^{(i)}$, and transmits the cipher stream to $\mathbf{R}$. Since $K^{(i)}$ is known to $\mathbf{u}^{(i)}$ and $\mathbf{R}$ only, only they can decrypt the cipher stream, and other attackers cannot access the clear text. After receiving and decrypting the transmitted bit streams from all colluders, the ringleader examines these fingerprinted copies for selfish colluder detection and identification purposes and broadcasts the results to all colluders. Finally, the attackers exclude those selfish colluders and apply the multi-user collusion attack.

This paper focuses on selfish colluder detection and identification in traitors within traitors. We investigate techniques to accurately detect pre-collusion processing and identify selfish colluders and analyze its performance.

2.4. Performance Criteria

Assume that SC is the set containing the indices of all colluders. SC_s includes the indices of all selfish colluders, and SC_h is the set with the indices of all the other colluders who do not apply pre-collusion processing. $SC_s \cap SC_h = \emptyset$ and $SC_s \cup SC_h = SC$.

Those colluders in SC_h wish to correctly identify all selfish colluders without falsely accusing any others. To measure the accuracy of the selfish colluder detection and identification algorithm, we consider two types of detection errors: the probability that a colluder in SC_h misses a selfish colluder in SC_s during detection (P_m), and the probability that a colluder in SC_h falsely accuse another colluder in SC_h as a selfish colluder (P_{fa}).

3. SELFISH COLLUDER DETECTION AND IDENTIFICATION

3.1. Review of Risk Minimization by Selfish colluders

For a selfish colluder to further reduce his/her own risk, one possible solution is to attenuate the energy of the embedded fingerprints even before multi-user collusion. For example, temporal filtering of adjacent frames was used in [5] to replace each segment of the fingerprinted signal with another, seemingly similar segment from different regions of the content.

Take the example in Figure 1, given the received fingerprinted frames $\{\mathbf{X}_j^{(i_1)}\}_{j=1,2,\cdots}$, for each frame j in the video sequence, the selfish colluder $\mathbf{u}^{(i_1)}$ linearly combines the current frame $\mathbf{X}_j^{(i_1)}$, the previous frame $\mathbf{X}_{j-1}^{(i_1)}$ and the next frame $\mathbf{X}_{j+1}^{(i_1)}$, and generates

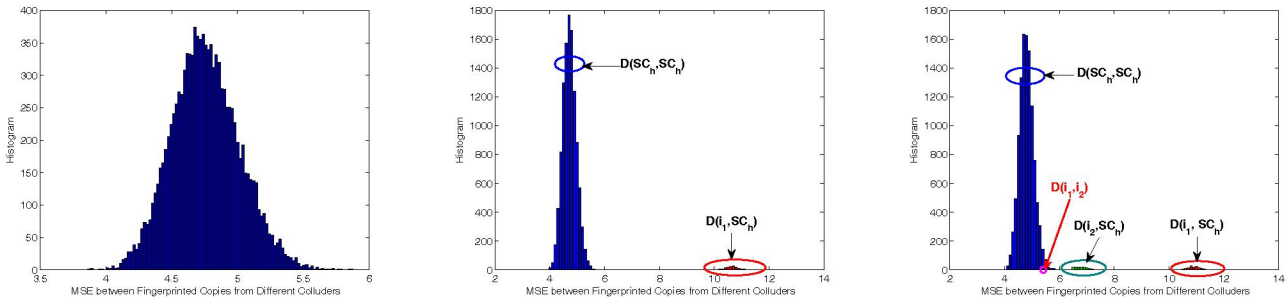
$$\widetilde{\mathbf{X}}_j^{(i_1)} = \frac{1-\lambda_j}{2}\mathbf{X}_{j-1}^{(i_1)} + \lambda_j \mathbf{X}_j^{(i_1)} + \frac{1-\lambda_j}{2}\mathbf{X}_{j+1}^{(i_1)}, \qquad (1)$$

where $0 \le \lambda_j \le 1$. To address the tradeoff between the risk of being detected and the perceptual quality of the newly generated frame, $\mathbf{u}^{(i_1)}$ chooses the parameter λ_j to minimize his own probability of being detected under the constraints that the MSE between the newly generated frame $\widetilde{\mathbf{X}}_j^{(i_1)}$ and the originally received frame $\mathbf{X}_j^{(i_1)}$ is no larger than a threshold ε. Details of the selection of the optimal λ_j are available in [5] and not repeated here. This process is repeated for all frames in the video sequence.

3.2. Detection of Pre-Collusion Processing

Assume that $\mathbf{S}_j$ is the jth frame in the original host signal, $\mathbf{W}_j^{(i)}$ is user $\mathbf{u}^{(i)}$'s fingerprint embedded in $\mathbf{S}_j$, and $\mathbf{X}_j^{(i)} = \mathbf{S}_j + \mathbf{W}_j^{(i)}$ is the fingerprinted frame j that $\mathbf{u}^{(i)}$ received from the content owner. (Note that we drop the term JND_j here to simplify the notations.) We further assume that $\mathbf{u}^{(i)}$ tells other colluders and the ringleader that $\widetilde{\mathbf{X}}_j^{(i)}$ is the jth frame that he/she received.

Assume that colluder $\mathbf{u}^{(k \in SC_h)}$ and $\mathbf{u}^{(l \in SC_h)}$ do not modify their received copies before collusion, and a selfish colluder $\mathbf{u}^{(i \in SC_s)}$



(a) (b) (c)

Fig. 2. Histogram of $\{D_j(k,l)\}$ with 150 colluders. (a): $SC_s = \emptyset$. (b): $SC_s = \{i_1\}$. PSNR of $\widetilde{\mathbf{X}}_j^{(i_1)}$ is $40dB$. (c): $SC_s = \{i_1, i_2\}$. $\widetilde{\mathbf{X}}_j^{(i_1)}$ has PSNR of $40dB$, and PSNR of $\widetilde{\mathbf{X}}_j^{(i_2)}$ is $45dB$. $\mathfrak{D}_j(i, SC_h) = \{D_j(i,l) : l \in SC_h\}$ for $i \in SC_s$.

uses (1) to temporally filter his/her copy during pre-collusion processing. Therefore, we have

$$\begin{aligned}
\widetilde{\mathbf{X}}_j^{(k)} &= \mathbf{X}_j^{(k)} = \mathbf{S}_j + \mathbf{W}_j^{(k)}, \\
\widetilde{\mathbf{X}}_j^{(l)} &= \mathbf{X}_j^{(l)} = \mathbf{S}_j + \mathbf{W}_j^{(l)}, \text{ and} \\
\widetilde{\mathbf{X}}_j^{(i)} &= \mathbf{S}_j + \Delta\mathbf{S}_j(\lambda_j) + \widetilde{\mathbf{W}}_j^{(i)}, \text{ where} \\
\Delta\mathbf{S}_j(\lambda_j) &= (1-\lambda_j)\left(\mathbf{S}_{j-1}/2 + \mathbf{S}_{j+1}/2 - \mathbf{S}_j\right), \text{ and} \\
\widetilde{\mathbf{W}}_j^{(i)} &= \frac{1-\lambda_j}{2}\mathbf{W}_{j-1}^{(i)} + \lambda_j \mathbf{W}_j^{(i)} + \frac{1-\lambda_j}{2}\mathbf{W}_{j+1}^{(i)}. \quad (2)
\end{aligned}$$

From (2), temporal filtering in (1) not only averages fingerprints embedded in adjacent frames and attenuates their energies, it also filters neighboring frames in the host signal and introduces additional distortion $\Delta\mathbf{S}_j(\lambda_j)$.

Fingerprints for different users are generated independently. Define $D_j(k,l) \triangleq ||\widetilde{\mathbf{X}}_j^{(k)} - \widetilde{\mathbf{X}}_j^{(l)}||^2$. From (2), since $\{\mathbf{W}_j^{(k)}\}$, $\{\mathbf{W}_j^{(l)}\}$ and $\{\mathbf{W}_j^{(i)}\}$ are independent of each other, we have

$$\begin{aligned}
D_j(k,l) &\approx ||\mathbf{W}_j^{(k)}||^2 + ||\mathbf{W}_j^{(l)}||^2 \text{ and} \\
D_j(k,i) &\approx ||\mathbf{W}_j^{(k)}||^2 + ||\widetilde{\mathbf{W}}_j^{(i)}||^2 + ||\Delta\mathbf{S}_j(\lambda_j)||^2. \quad (3)
\end{aligned}$$

From (3), for three colluders $k \in SC_h$, $l \in SC_h$ and $i \in SC_s$, $D_j(k,l)$ can be approximated by the summation of $||\mathbf{W}_j^{(k)}||^2$ and $||\mathbf{W}_j^{(l)}||^2$; while $D_j(k,i)$ also includes $||\Delta\mathbf{S}_j(\lambda_j)||^2$ that is the additional distortion introduced by temporal filtering (1). Thus, $D_j(k,i)$ has a much larger value than $D_j(k,l)$, and the difference between $D_j(k,i)$ and $D_j(k,l)$ is more obvious when λ_j in (1) is smaller.

Figure 2 shows an example of the histogram of $\{D_j(k,l)\}$ for the 2nd frame in sequence carphone. We adopt the human visual model based spread spectrum embedding [6] and additively embed the fingerprints in the DCT domain. Fingerprints $\{\mathbf{W}^{(i)}\}$ follow Gaussian distribution $\mathcal{N}(0, 1/9)$ and fingerprints for different users are generated independently. There are a total of 150 colluders in Figure 2, and SC is the set containing their indices.

In Figure 2 (a), all colluders give each other correct information about their fingerprinted copies and $SC_s = \emptyset$. In Figure 2 (b), there is one selfish colluder $\mathbf{u}^{(i_1)}$, and he/she selects the parameter λ_j in (1) such that PSNR of the newly generated frame $\widetilde{\mathbf{X}}_j^{(i_1)}$ is at least 40dB when compared with the originally received one $\mathbf{X}_j^{(i_1)}$. In Figure 2 (c), there are two selfish colluders $\mathbf{u}^{(i_1)}$ and $\mathbf{u}^{(i_2)}$, and they use (1) to process their fingerprinted copies independently. PSNRs of $\widetilde{\mathbf{X}}_j^{(i_1)}$ and $\widetilde{\mathbf{X}}_j^{(i_2)}$ are 40dB and 45dB, respectively.

Define $\mathfrak{D}_j(SC_h, SC_h) \triangleq \{D_j(k,l) : k,l \in SC_h, k \neq l\}$ and $\mathfrak{D}_j(SC_h, SC_s) \triangleq \{D_j(k,l) : k \in SC_h, l \in SC_s\}$. From Figure 2, $\{D(k,l)\}$ follow the same distribution with a single mean when no colluder applies pre-collusion processing; while when some selfish colluders process their copies before collusion, $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_h, SC_s)$ are from different distributions with distinct means. The smaller the value of λ_j in (1), the larger the distance between $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_h, SC_s)$.

The above analysis suggests that the histogram of $\{D_j(k,l)\}$ can be used to determine the existence of selfish colluders. If $\{D(k,l)\}$ are from the same distribution with a single mean, then all colluders keep their fair-collusion agreement. If $\{D(k,l)\}$ are from two or more distributions with different means, there exists at least one selfish colluder who applies pre-collusion processing.

3.3. Selfish Colluder Identification

Identification of the selfish colluders requires detailed examination of each $D_j(k,l)$. Given $\{D_j(k,l)\}$, the ringleader can only separate the colluders into two subgroups, while it is difficult for the ringleader to tell which subgroup contains the selfish colluders and which subgroup is SC_h. Since each colluder knows which subgroup he/she belongs to, given $\{D_j(k,l)\}$, it is much easier for the attackers themselves to identify the selfish colluders.

Given $\{D_j(k,l)\}$, a colluder $\mathbf{u}^{(i)}$ in SC_h applies Algorithm 1 to identify the selfish colluders. For a total of K colluders, $\mathbf{\Phi} = (\Phi_{i_1}, \Phi_{i_2}, \cdots, \Phi_{i_K})$ in Algorithm 1, where $\Phi(k) = 1$ when $\mathbf{u}^{(k)}$ is believed to be a selfish colluder and $\Phi(k) = 0$ if $\mathbf{u}^{(k)}$ is considered to be in subgroup SC_h. $\mathbf{u}^{(i)}$ first initializes $\mathbf{\Phi}$ to an undetermined status -1 and sets $\Phi(i)$ to 0 since $i \in SC_h$.

During the selfish colluder identification process, $\mathbf{u}^{(i \in SC_h)}$ examines every $D_j(k,l)$ and starts with the one with the largest value. This is because, a larger value of $D_j(k,l)$ gives $\mathbf{u}^{(i)}$ higher confidence that the two corresponding colluders $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ belong to different subgroups. For each $D_j(k,l)$ and the corresponding two colluders $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$, $\mathbf{u}^{(i)}$ first checks if he/she has determined the values of $\Phi(k)$ and $\Phi(l)$ in the previous rounds.

- If both $\Phi(k)$ and $\Phi(l)$ have been decided, $\mathbf{u}^{(i)}$ moves to the next largest $D_j(k,l)$.
- If one of them is set to either 0 or 1 while the other is still undetermined, without loss of generality, assume that $\Phi(k)$ has been determined, then $\Phi(l) = 1 - \Phi(k)$.

Algorithm 1: Selfish colluder identification by $\mathbf{u}^{(i)}$ in SC_h.

Algorithm: $SelfishColluderIDAlg(\{D_j\})$
Set $\Psi_t = \{i\}$, $\mathbf{\Phi} = -\mathbf{1}_{1\times K}$, $\Phi(i) = 0$;
Set $m = 0$;
while $\Psi_t \neq SC$ **do**
  $m = m + 1$;
  select $D_j(k,l)$ with the m^{th} largest value and take the indices of the two corresponding colluder k, l;
  if $k \notin \Psi_t$ *AND* $l \notin \Psi_t$ **then**
    if $D_j(i,k) > D_j(i,l)$ **then**
      $\Phi(k) = 1; \Phi(l) = 0; \Psi_t = \Psi_t \cup \{k,l\}$;
    else
      $\Phi(k) = 0; \Phi(l) = 1; \Psi_t = \Psi_t \cup \{k,l\}$;
    end
  else
    if $k \in \Psi_t$ *AND* $l \notin \Psi_t$ **then**
      $\Phi(l) = 1 - \Phi(k), \Psi_t = \Psi_t \cup \{l\}$;
    end
    if $l \in \Psi_t$ *AND* $k \notin \Psi_t$ **then**
      $\Phi(k) = 1 - \Phi(l), \Psi_t = \Psi_t \cup \{k\}$;
    end
  end
end
return $\widehat{SC}_s(i) = \{k : \Phi(k) = 1\}$.

- If $\mathbf{u}^{(i)}$ is unable to determine either $\Phi(k)$ or $\Phi(l)$ in the previous rounds, he/she compares the values of $D_j(k,i)$ and $D_j(l,i)$. Without loss of generality, assume that $D_j(k,i) > D_j(l,i)$. Compared with $\mathbf{u}^{(l)}$, $\mathbf{u}^{(k)}$ is more likely to be a selfish colluder. Thus, $\mathbf{u}^{(i)}$ sets $\Phi(l) = 0$ and $\Phi(k) = 1$.

$\mathbf{u}^{(i)}$ repeats the above process and stops when all the components in $\mathbf{\Phi}$ have been set to either 0 or 1. The output $\widehat{SC}_s(i)$ contains the indices of the attackers whom $\mathbf{u}^{(i)}$ considers as selfish colluders.

3.4. Selfish Colluder Detection and Identification Algorithm

To summarize, with a trusted ringleader, the key steps in selfish colluder detection and identification are:

Step 1 Encryption: Each colluder $\mathbf{u}^{(i)}$ encrypts his/her fingerprinted copy with a secret key K^i shared with the ringleader $\mathbf{R}$ only and transmits the encrypted bit stream to the ringleader.

Step 2 Calculation of $\{D\}$: After decrypting the cipher streams received from all colluders, the ringleader calculates $D_j(k,l)$ for each pair of colluders $(\mathbf{u}^{(k)}, \mathbf{u}^{(l)})$. Then, $\mathbf{R}$ signs $\{D_j(k,l)\}$ with his/her digital signature and broadcasts to all colluders.

Step 3 Detection of Pre-collusion Processing: Colluders examine the histogram of $\{D_j(k,l)\}$. If $\{D_j(k,l)\}$ are from the same distribution with a single mean, then there are no selfish colluders and the attackers go to Step 5 to collude with each other. If $\{D_j(k,l)\}$ are from two or more distributions with different means, there is at least one selfish colluder and the attackers in SC_h go to Step 4 to identify selfish colluders.

Step 4 Selfish Colluder Identification: Each colluder in SC_h applies Algorithm 1 to estimate the identities of the selfish colluders.

Step 5 Multi-user Collusion: Colluders in SC_h exclude those identified selfish colluders from collusion and generate a colluded copy as shown in Figure 1.

To verify the accuracy of the proposed algorithm, we select three typical video sequences, "miss america", "carphone" and "flower", and test on the first 10 frames in each sequence. $\{\mathbf{W}^{(i)}\}$ follow distribution $\mathcal{N}(0, 1/9)$ and fingerprints for different users are generated independently. Human visual model based spread spectrum embedding [6] is used to embed fingerprints into the DCT domain of the host signal.

We assume that the total number of colluders is 150. There are 10 selfish colluders and each processes his/her fingerprinted copy independently before collusion. Among the 10 selfish colluders, 5 of them select λ_j in (1) to ensure that PSNR of the newly generated frames is at least 40dB; while the other 5 selfish colluders select λ_j so that $PSNR \geq 45dB$ for all the newly generated frames.

For each sequence, we run 1000 simulation runs to test the performance of the proposed algorithm. In our 3000 simulation runs, all colluders in SC_h can accurately identify all selfish colluders in SC_s and no one in SC_h falsely accuses any others in SC_h as selfish colluders, i.e., $P_m = 0$ and $P_{fa} = 0$. Therefore, the proposed selfish colluder detection and identification algorithm does not make either type of detection errors. This is because, the two distributions $\mathfrak{D}(SC_h, SC_s)$ and $\mathfrak{D}(SC_h, SC_h)$ are well separated from each other in Figure 2, and it ensures the error-free performance of the proposed algorithm.

4. CONCLUSIONS

In this paper, we investigate the traitor-within-traitor dynamics among attackers during collusion and explore the techniques that attackers can use to detect and identify selfish colluders in order to protect their own interests. We propose a selfish colluder detection and identification algorithm, which uses the difference between fingerprinted copies from different colluders to detect pre-collusion processing and identify selfish colluders. The proposed algorithm protects the secrecy of all the fingerprinted copies. Our simulation results show that the proposed algorithm can accurately identify all selfish colluders without falsely accusing any others.

5. REFERENCES

[1] F. Ergun, J. Killian, and R. Kumar, "A note on the limits of collusion-resistant watermarks," *Advances in Cryptology – EuroCrypto '99, Lecture Notes in Computer Science*, vol. 1592, pp. 140–149, 2001.

[2] J. Su, J. Eggers, and B. Girod, "Capacity of digital watermarks subject to an optimal collusion attacks," *European Signal Processing Conference (EUSIPCO 2000)*, 2000.

[3] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Tran. on Image Processing*, vol. 14, no. 5, pp. 646–661, May 2005.

[4] Z. J. Wang, M. Wu, H. Zhao, W. Trappe, and K. J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. on Image Processing*, vol. 14, no. 6, pp. 804–821, June 2005.

[5] H. V. Zhao and K. J. R. Liu, "Risk minimization in traitors within traitors in multimedia forensics," *IEEE Int. Conf. on Image Processing*, vol. 3, pp. 89–92, Sept. 2005.

[6] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE Journal on Sel. Area in Comm.*, vol. 16, no. 4, pp. 525–540, May 1998.

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