Impact of Social Network Structure on Multimedia Fingerprinting Misbehavior Detection and Identification

H. Vicky Zhao, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Users in video-sharing social networks actively interact with each other, and it is of critical importance to model user behavior and analyze the impact of human factors on video sharing systems. In video-sharing social networks, users have access to extra resources from their peers, and they also contribute their own resources to help others. Each user wants to maximize his/her own payoff, and they negotiate with each other to achieve fairness and address this conflict. However, some selfish users may cheat to their peers and manipulate the system to maximize their own payoffs, and cheat prevention is a critical requirement in many social networks to stimulate user cooperation. It is of ample importance to design monitoring mechanisms to detect and identify misbehaving users, and to design cheat-proof cooperation stimulation strategies. Using video fingerprinting as an example, this paper analyzes the complex dynamics among colluders during multiuser collusion, and explores possible monitoring mechanisms to detect and identify misbehaving colluders in multiuser collusion. We consider two types of colluder networks: one has a centralized structure with a trusted ringleader, and the other is a distributed peer-structured network. We investigate the impact of network structures on misbehavior detection and identification, propose different selfish colluder identification schemes for different colluder networks, and analyze their performance. We show that the proposed schemes can accurately identify selfish colluders without falsely accusing others even under attacks. We also evaluate their robustness against framing attacks and quantify the maximum number of framing colluders that they can resist.**

***Index Terms*—Msbehavior detection and identification, multimedia fingerprinting, social network structure.**

I. Introduction

In the past decades, advances in communication, networking and multimedia have led to the proliferation of multimedia applications. We witness the emergence of large-scale multimedia social network communities (for example, Napster, YouTube, CoolStreaming, and PPLive), where millions of users form a distributed infrastructure to share multimedia content [1]–[3]. A critical issue in multimedia social networks is to understand the user dynamics that influence human's behavior and analyze the impact of human factors on multimedia systems [4], [5]. This investigation provides fundamental guidelines to the systematic design of multimedia systems and helps develop better technologies to offer secure and personalized services. The area of human and social dynamics has recently been identified by U.S. National Science Foundation as one of its five priority areas, which also shows the importance of this emerging interdisciplinary research area.

This paper analyzes the complex user dynamics in video-sharing social networks, where users cooperate with each other to share videos. Cooperation enables users to access extra resources from their peers and thus receiving higher payoffs, while each user also needs to contribute his or her own resource to help others. Each user wants to maximize his or her own payoff, and different users have different objectives. To address this conflict, users negotiate with each other and achieve *fairness*. However, some users might be selfish and wish to consume others' resources without contributing their own. Recent studies showed that it is easy for users to cheat and manipulate the system to further increase their payoffs in many online social networks, for example, peer-to-peer file sharing [6], peer-to-peer gaming [7], visual cryptography for secret sharing [8], etc. Therefore, cheat prevention is a fundamental requirement to achieve user cooperation in multimedia social networks. To analyze user dynamics in social networks containing selfish users, the first step is to study strategies that selfish users can use to cheat, and then design monitoring mechanisms to detect and identify misbehaving users. Such a monitoring mechanism facilitates the design of cheat-proof strategies, which makes non-cooperation non-profitable and thus unattractive to selfish users. This paper studies human dynamics in video fingerprinting, analyzes the cheating behavior in video fingerprinting, and explores monitoring mechanisms that may be used to detect and identify misbehaving users.

Video fingerprinting labels each distributed video copy with the corresponding user's ID, and this embedded "fingerprint" provides forensic tools for the content owner to track the distribution of video data and identify the source of illicit copies [9]–[12]. However, the uniqueness of each distributed copy also enables a group of colluders to collectively and effectively attenuate the embedded fingerprints [13]–[15]. During collusion, colluders share the reward from the illegal usage of multimedia as well as the risk of being detected by the digital rights enforcer.

Manuscript received October 14, 2009; revised February 08, 2010; accepted February 12, 2010. Date of publication May 20, 2010; date of current version July 16, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vikram Krishnamurthy.

H. Vicky Zhao is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4 Canada (e-mail: vzhao@ece.ualberta.ca).

K. J. Ray Liu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@umd.edu).

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/JSTSP.2010.2051256

Each colluder wishes to distribute the risk and the reward in a way that favors him or her the most, and a critical issue during collusion is to achieve fairness.

To achieve fair collusion, colluders are required to provide one another correct information about their fingerprinted copies. Then, they adjust the collusion parameters accordingly to distribute the risk evenly among them. However, some selfish colluders may wish to profit from multiuser collusion while taking no risk. To further reduce their own probability of being detected by the digital rights enforcer, they can process their received copies before multiuser collusion, and use the processed copies, instead of the originally received ones, during collusion [16]. Precollusion processing reduces the selfish colluders' relative risk with respect to their fellow colluders, and in some scenarios, it may even increase other colluders' probability of being detected [16]. To protect their own interest, colluders must examine all fingerprinted copies before collusion, detect and identify selfish colluders, and exclude them from collusion. Accurate and secure selfish colluder detection and identification can force all colluders to keep their fair-play agreement and build trust among colluders, and is an important part of multiuser collusion.

In this paper, we explore possible strategies to detect and identify selfish colluders, investigate the impact of network structures on misbehavior detection and identification, and analyze the performance of the proposed schemes. Addressing different structures in different social networks, we first consider a centralized social network with a trusted ringleader and investigate misbehavior detection with the trusted ringleader's help. We then consider the peer-structured social networks where all colluders take the same role, and examine autonomous selfish colluder identification. From video fingerprinting perspective, the work in [17] showed that probing and utilizing side information about how attackers collude enables the fingerprint detector to adaptively adjust the detection strategy and significantly improves the collusion resistance. As an important part of multiuser collusion, such an investigation on colluder dynamics helps us have a better understanding of the collusion attack, and provides important guidelines on collusion-resistant fingerprinting system design. From human behavior modeling perspective, it provides a case study of misbehavior detection and identification in video fingerprinting, analyzes how network structures affect the performance of social networks, and gives important insights for the design of cheat-proof cooperation strategies in other video social networks. To our knowledge, this is the first work that studies misbehavior detection in multiuser collusion and analyzes the impact of colluder network structures on collusion attacks.

The rest of this paper is organized as follows. We begin in Section II with the introduction of video fingerprinting systems and the formulation of colluder behavior dynamics. In Section III, we consider a centralized colluder social network with a trusted ringleader and explore how this trusted ringleader can help detect selfish colluders. Section IV investigates autonomous selfish colluder detection and identification in distributed peer-structured colluder social networks. Conclusions are drawn in Section V.

II. System Model and Problem Formulation

A. Video Fingerprinting Systems

In this paper, we consider spread spectrum embedding, which has been widely used in multimedia fingerprinting systems due to its robustness against many single-copy attacks [18]. In additive spread spectrum embedding for video applications, for the jth frame in the video sequence represented by a vector $\mathbf{S}_j$ of length N_j, the content owner generates a unique fingerprint $\mathbf{W}_j^{(i)}$ of length N_j for each user $u^{(i)}$ in the system. The fingerprinted copy that is distributed to $\mathbf{u}^{(i)}$ is $\mathbf{X}_j^{(i)} = \mathbf{S}_j + \mathrm{JND}_j \cdot \mathbf{W}_j^{(i)}$. Here, JND_j is the *just-noticeable-difference* from human visual models [18] to control the energy of the embedded fingerprints. Finally, the content owner transmits to each user $u^{(i)}$ the fingerprinted frames $\{\mathbf{X}_j^{(i)}\}$. In this paper, we assume that the total number of users is much smaller than the length of the embedded fingerprints, and consider orthogonal fingerprint modulation [10], [19], where fingerprints for different users are orthogonal to each other and have equal energy.[1] To resist intra-content collusion attacks on video watermarking [20]–[22], in each fingerprinted copy, fingerprints embedded in adjacent frames are correlated with each other.

During collusion, the colluders collect all the fingerprinted copies that they received, apply the multi-user collusion function to these copies, and generate a new copy in which the originally embedded fingerprints are removed or attenuated. For example, a simple average of all the fingerprinted copies reduces the energy of each contributing fingerprint and, therefore, lowers the colluders' probability of being detected. Another important class of collusion attack is based on operations as taking the minimum, maximum and median of corresponding components of the fingerprinted signals [13]. A recent investigation in [19] showed that, with orthogonal fingerprint modulation, under the constraints that the colluded copies from different collusion attacks have the same perceptual quality, the performance of nonlinear collusion attacks is similar to that of the averaging attack. After multiuser collusion, colluders can apply other single-copy attacks (for example, low-pass filtering, compression, etc.) to further hinder the fingerprint detection process.

Once the content owner discovers the existence of an illegal copy in the market, for each frame in the colluded copy, the detector first extracts the fingerprint from the test copy. Then, the detector calculates the correlation-based detection statistics [10], [19], which is widely used in the literature to measure the similarity between the extracted fingerprint and the original fingerprint. Finally, the detector compares the detection statistics with a predetermined threshold, and outputs the estimated identities of the colluders.

[1] Another important class of multimedia fingerprinting is the coded fingerprint modulation, where fingerprints assigned to different users are correlated with each other [10], [12]. In this paper, we use orthogonal fingerprint modulation as an example to study how to detect precollusion processing and identify selfish colluders in multiuser collusion. We plan to investigate in the future the cheating behavior and the misbehavior detection in multiuser collusion with coded fingerprint modulation.

B. Behavior Dynamics in Colluder Social Networks

Colluders share the reward from unauthorized usage of video content as well as the risk of being detected by the fingerprint detector. They prefer to reach an agreement regarding how to distribute the risk and the reward rather than abstain from doing so since colluding with others helps reduce his or her risk of being detected. However, each colluder prefers the agreement that favors him or her most. To address the conflicting objectives, colluders usually apply fair collusion and let all colluders have the same probability of being detected. To achieve fairness, all colluders are required to provide correct information about their received fingerprinted copies. Then, they adjust the collusion attacks accordingly.

1) Cheating Behavior in Multiuser Collusion: Most prior work assumed that all colluders keep their agreement of fair collusion, which may not always hold in reality. There might exist some selfish colluders who wish to take no risk while still profiting from collusion. For a selfish colluder to further reduce his/her own risk, one possible solution is to attenuate the energy of the fingerprint embedded in his/her received copy even before multiuser collusion. An example is to explore the redundancy in and the correlation between neighboring samples in multimedia data, and replace each segment of the fingerprinted signal with another, seemingly similar segment from different regions of the content. As an example, temporal filtering was used in [16] to process the fingerprinted video before collusion.

In [16], given the received fingerprinted frames $\{\mathbf{X}_j^{(i)}\}_{j=1,2,...}$, the selfish colluder $u^{(i)}$ can use linear interpolation to produce a temporally filtered video [16]. For each frame j in the video sequence, $u^{(i)}$ linearly combines the current frame $\mathbf{X}_j^{(i)}$, the previous frame $\mathbf{X}_{j-1}^{(i)}$ and the next frame $\mathbf{X}_{j+1}^{(i)}$, and generates

$$\tilde{\mathbf{X}}_j^{(i)} = \frac{1-\lambda_j}{2}\mathbf{X}_{j-1}^{(i)} + \lambda_j \mathbf{X}_j^{(i)} + \frac{1-\lambda_j}{2}\mathbf{X}_{j+1}^{(i)} \quad (1)$$

where $0 \le \lambda_j \le 1$. Motion-based interpolation [23], [24] can be used to improve the quality of $\tilde{\mathbf{X}}_j^{(i)}$, and the analysis will be similar to that in [16]. The selfish colluder $u^{(i)}$ selects the parameter λ_j to minimize his/her chance of being detected under the constraint that the newly generated frame has small perceptual distortion when compared with the originally received one.

The work in [16] also investigated other possible techniques for selfish colluders to further reduce their own risk of being detected, for example, changing the resolution of their fingerprinted copies. It was shown in [16] that such a selfish behavior reduces the selfish colluders' own probability of being detected and makes other colluders take a relatively larger chance of being captured than the selfish colluders. In some scenarios, precollusion processing may also increase other colluders' absolute risk, i.e., their probability of being detected [16]. In these scenarios, precollusion processing is not only selfish but also malicious.

2) Misbehavior Detection and Identification: The existence of selfish colluders complicates multiuser collusion. No colluder knows what others have done to their fingerprinted copies and how it will affect his/her own probability of being detected. To continue collusion, colluders must detect and identify selfish behavior, force everyone to keep their fair-play agreement, and establish trust among themselves. In this paper, using temporal filtering as an example, we explore possible strategies to accurately detect precollusion processing and identify selfish colluders.

Define SC as the set containing the indices of all colluders. The set $SC_s \subseteq SC$ includes all selfish colluders, and $SC_h = SC \backslash SC_s$ is the set with the indices of all honest colluders who do not apply precollusion processing. Let $\tilde{\mathbf{X}}_j^{(i)}$ be the jth frame from colluder $u^{(i)}$. For honest colluders $u^{(k)}$ and $u^{(l)}$, we have

$$\begin{aligned} \tilde{\mathbf{X}}_j^{(k)} &= \mathbf{X}_j^{(k)} = \mathbf{S}_j + \mathbf{W}_j^{(k)} \quad \text{and} \\ \tilde{\mathbf{X}}_j^{(l)} &= \mathbf{X}_j^{(l)} = \mathbf{S}_j + \mathbf{W}_j^{(l)}. \end{aligned} \quad (2)$$

(Here, we drop the term JND to simplify the notations.) For a selfish colluder $u^{(i \in SC_s)}$

$$\begin{aligned} \tilde{\mathbf{X}}_j^{(i)} &= \frac{1-\lambda_j}{2}\mathbf{X}_{j-1}^{(i)} + \lambda_j \mathbf{X}_j^{(i)} + \frac{1-\lambda_j}{2}\mathbf{X}_{j+1}^{(i)} \\ &= \mathbf{S}_j + \Delta\mathbf{S}_j(\lambda_j) + \tilde{\mathbf{W}}_j^{(i)} \end{aligned}$$

where

$$\Delta\mathbf{S}_j(\lambda_j) = (1-\lambda_j)\left(\frac{\mathbf{S}_{j-1}}{2} + \frac{\mathbf{S}_{j+1}}{2} - \mathbf{S}_j\right)$$

and

$$\begin{aligned} \tilde{\mathbf{W}}_j^{(i)} &= \frac{1-\lambda_j}{2}\mathbf{W}_{j-1}^{(i)} + \lambda_j \mathbf{W}_j^{(i)} \\ &\quad + \frac{1-\lambda_j}{2}\mathbf{W}_{j+1}^{(i)}. \end{aligned} \quad (3)$$

From (3), temporal filtering not only averages fingerprints embedded in adjacent frames and attenuates their energies, it also filters neighboring frames in the host signal and introduces extra distortion $\Delta\mathbf{S}_j(\lambda_j)$.

For the jth fingerprinted frames from $u^{(k)}$ and $u^{(l)}$, define $D_j(k,l) \triangleq \|\tilde{\mathbf{X}}_j^{(k)} - \tilde{\mathbf{X}}_j^{(l)}\|^2$. Since $\{\mathbf{W}_j^{(k)}\}$, $\{\mathbf{W}_j^{(l)}\}$ and $\{\mathbf{W}_j^{(i)}\}$ are orthogonal to each other, from (2) and (3), we have

$$D_j(k,l) \approx \left\|\mathbf{W}_j^{(k)}\right\|^2 + \left\|\mathbf{W}_j^{(l)}\right\|^2$$

and

$$\begin{aligned} D_j(k,i) \approx &\left\|\mathbf{W}_j^{(k)}\right\|^2 + \left\|\tilde{\mathbf{W}}_j^{(i)}\right\|^2 \\ &+ \left|\Delta\mathbf{S}_j(\lambda_j)\right\|^2 \end{aligned}$$

where

$$\begin{aligned} \|\Delta\mathbf{S}_j(\lambda_j)\|^2 = &(1-\lambda_j)^2 \\ &\times \left\|\frac{\mathbf{S}_{j-1}}{2} + \frac{\mathbf{S}_{j+1}}{2} - \mathbf{S}_j\right\|^2. \end{aligned} \quad (4)$$

For honest colluders k and l, $D_j(k,l)$ can be approximated by the summation of the energies of the two embedded fingerprints $\mathbf{W}_j^{(k)}$ and $\mathbf{W}_j^{(l)}$. For the honest colluder k and the selfish colluder i, in addition to the summation of $\|\mathbf{W}_j^{(k)}\|^2$ and $\|\tilde{\mathbf{W}}_j^{(i)}\|^2$, $D_j(k,i)$ also includes the additional distortion $\|\Delta\mathbf{S}_j(\lambda_j)\|^2$ introduced by temporal filtering in (1). Therefore, $D_j(k,i)$ has a much larger value than $D_j(k,l)$. For a given video sequence, from (4), the difference between $D_j(k,i)$ and

$D_j(k,l)$ is more obvious when λ_j takes a smaller value. In addition, $|D_j(k,i)-D_j(k,l)|$ takes a larger value when the scene of the host video sequence changes fast and the difference between adjacent frames is larger. This observation suggests that $\{D_j(k,l)\}$ can help honest colluders detect precollusion processing and identify selfish colluders.

Note that before a colluder decides with whom to collude, he/she is unwilling to give others his/her received fingerprinted copy that contains his/her identification information. Therefore, selfish colluder detection and identification must prevent attackers from accessing the fingerprinted coefficients in others' copies. To meet this antiframing requirement in selfish colluder detection and identification, all fingerprinted copies must be encrypted appropriately during this selfish behavior detection process.

In this paper, we investigate how colluders can securely calculate $\{D_j(k,l)\}$, explore techniques for honest colluders to accurately identify selfish colluders, and analyze their performance. Note that different structures of social networks result in different strategies to detect and identify misbehaving users. Some social networks have a centralized structure where there are one or more entities whom all users trust and who can facilitate interaction among users. For example, the first generation peer-to-peer file-sharing networks (for example, the Napster music file-sharing system) used a set of central servers to provide content indexing and search services [2]. Although these servers cannot enforce user cooperation, they can help monitor users' behavior. Other media-sharing social networks have a distributed structure and a flat topology where users take the same role, for example, Gnutella and Chord [2]. There, users have to monitor other users and identify misbehavior themselves.

Addressing different network structures, in this paper, we first consider a centralized social structure where there is a ringleader whom all colluders trust. We investigate how the trusted ringleader can help detect precollusion processing and identify selfish colluders. We then consider the distributed peer social structure of the colluder social networks, and study the autonomous selfish colluder detection and identification, in which attackers help each other detect selfish behavior and identify selfish colluders. In this paper, we consider the scenario where there are only a few selfish colluders and most colluders honestly report private information of their fingerprinted copies to others. We investigate how honest colluders can collaborate with each other to accurately identify misbehaving users and analyze its performance. Furthermore, we consider the scenario where colluders have sufficient time to detect selfish colluders and generate the colluded copy, and assume that they have sufficient bandwidth to exchange fingerprinted copies with each other.

C. Performance Criteria

The selfish colluder detection and identification process aims to accurately identify all selfish colluders without falsely accusing any others. To measure the performance of the selfish colluder detection and identification algorithm, we consider two types of detection errors and use the following criteria:

- P_{md}: the rate that an honest colluder misses a selfish colluder during detection;
- P_{fa}: the rate that an honest colluder falsely accuses another honest colluder as a selfish colluder.

To evaluate the antiframing performance of the proposed scheme, assume that the fingerprinted frame j that colluder $u^{(i)}$ receives is $\mathbf{X}_j^{(i)} = [X_j^{(i)}(1), X_j^{(i)}(2), \ldots, X_j^{(i)}(N_j)]$, where $X_j^{(i)}(l)$ is the lth component in $\mathbf{X}_j^{(i)}$, and $\mathbf{X}_j^{(i)}$ is of length N_j. During the selfish colluder detection and identification process, without proper encryption, it is possible that another colluder $u^{(k)}$ can access some of the fingerprinted coefficients in $\mathbf{X}_j^{(i)}$. Assume that $\mathrm{Ind}_j^{(k,i)} \subseteq \{1,2,\ldots,N_j\}$ includes the indices of all the fingerprinted coefficients in $\mathbf{X}_j^{(i)}$ that $u^{(k)}$ can access, and define $\mathrm{Ind}_j^{(k)} \triangleq \bigcup_{i\in SC, i\neq k} \mathrm{Ind}_j^{(k,i)}$. If $\mathrm{Ind}_j^{(k)} = \{1,2,\ldots,N_j\}$, then $u^{(k)}$ can generate a new copy $\mathbf{Z}_j$ of high quality that does not contain any information of his/her own fingerprint, and $u^{(k)}$ can use $\mathbf{Z}_j$ to frame other colluders in SC.

To evaluate the resistance of the proposed algorithms to framing attacks, we define

$$\gamma_j \triangleq \frac{E\left[\left|\mathrm{Ind}_j^{(k)}\right|\right]}{N_j}, \quad 0 \le \gamma_j \le 1 \tag{5}$$

where $E[X]$ returns the statistical mean of X, and $|A|$ is the size of the set A. A smaller γ_j indicates that the selfish colluder detection and identification process is more robust against framing attacks.

III. Centralized Colluder Social Networks With Trusted Ringleaders

In this section, we consider a centralized colluder social network where there is a trusted ringleader, and we study how to detect and identify selfish colluders there. All colluders believe that the trusted ringleader will not give their fingerprinted copies to others; the ringleader himself will not frame any colluders; and the ringleader will not modify the selfish colluder detection and identification results.

To identify selfish colluders, each colluder $u^{(i)}$ first generates a secret key $K^{(i)}$ shared with the ringleader $\mathbf{R}$ only, encrypts his/her fingerprinted copy with $K^{(i)}$ to prevent others' eavesdropping of the communication, and transmits the encrypted version to $\mathbf{R}$. Since $K^{(i)}$ is known to $u^{(i)}$ and $\mathbf{R}$ only, no one but $u^{(i)}$ and $\mathbf{R}$ can decrypt the transmitted bit stream, and other colluders cannot access the fingerprinted coefficients. After receiving and decrypting the transmitted bit streams from all colluders, the ringleader examines these fingerprinted copies and helps detect and identify selfish colluders. Finally, colluders exclude those identified selfish colluders from multiuser collusion.

In this paper, we consider the scenario where colluders receive fingerprinted copies of the same quality (SNR). When they receive fingerprinted copies of different quality due to network heterogeneity and dynamically changing channel conditions, a challenging issue is to differentiate the scenario where the colluder intentionally changed his/her received fingerprinted copy from another one where this copy was transmitted through severely congested and erroneous networks. In our future work,

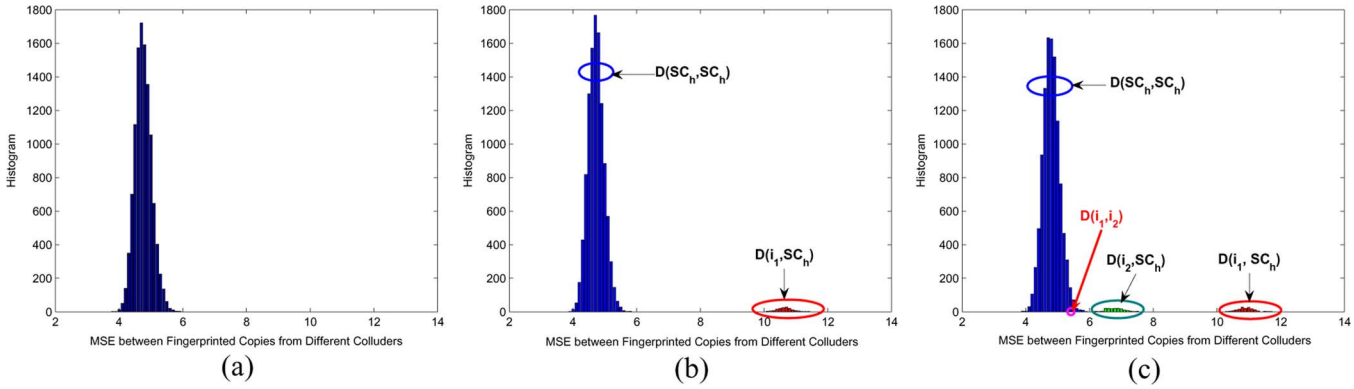

Fig. 1. Histogram of $\{D_j(k,l)\}_{k,l\in SC}$ on the second frame of sequence carphone. The length of the embedded fingerprint is 4179. There are a total $KC = 150$ colluders. (a) $SC_s = \emptyset$ and $SC_h = SC$. (b) $SC_s = \{i_1\}$ and $SC_h = SC\backslash\{i_1\}$. The selfish colluder $u^{(i_1)}$ uses $\lambda_j = 0.6031$ in (1) and $\bar{\mathbf{X}}_j^{(i_1)}$ has PSNR of 40 dB. (c) $SC_s = \{i_1, i_2\}$ and $SC_h = SC\backslash\{i_1, i_2\}$. During precollusion processing, $u^{(i_1)}$ uses $\lambda_j = 0.6031$ to generate $\bar{\mathbf{X}}_j^{(i_1)}$ and $\text{PSNR}_j = 40$ dB. $u^{(i_2)}$ uses $\lambda_j = 0.7759$ in (1) and generates a new frame $\bar{\mathbf{X}}_j^{(i_2)}$ with $\text{PSNR}_j = 45$ dB. For a selfish colluder $i \in SC_s$, $\mathfrak{D}_j(i, SC_h) = \{D_j(i,l) : l \in SC_h\}$.

we plan to investigate selfish colluder detection and identification when colluders receive fingerprinted copies of different quality.

A. Detection of Temporal-Interpolation-Based Precollusion Processing

Following the discussion in Section II-B2, in this paper, we use $\{D_j(k,l)\}$ to detect precollusion processing and identify selfish colluders. Fig. 1 shows an example of the histogram of $\{D_j(k,l)\}_{k,l\in SC}$ for the second frame in sequence carphone. Other frames and other sequences give the same trend. In our simulations, we adopt the human visual model based spread spectrum embedding [18] and embed fingerprints in the DCT domain. Fingerprints are generated from Gaussian distribution $\mathcal{N}(0, \sigma_W^2)$ with $\sigma_W^2 = 1/9$. During collusion, we assume that there are a total of $KC = 150$ colluders and SC is the set containing their indices.

In Fig. 1(a), there are no selfish colluders and all colluders provide one another correct information of their fingerprinted copies. In Fig. 1(b), there is one selfish colluder $u^{(i_1)}$ who applies temporal filtering before collusion. During precollusion processing, $u^{(i_1)}$ selects the parameter $\lambda_j = 0.6031$ to generate a new frame $\tilde{\mathbf{X}}_j^{(i_1)}$ with peak signal-to-noise ratio (PSNR) of 40 dB when compared with the originally received one $\mathbf{X}_j^{(i_1)}$. In Fig. 1(c), there are two selfish colluders, $u^{(i_1)}$ and $u^{(i_2)}$, who process their fingerprinted copies independently and impose different fidelity constraints. Same as in Fig. 1(b), $u^{(i_1)}$ uses (1) to generate a new frame $\tilde{\mathbf{X}}_j^{(i_1)}$ of 40 dB. During precollusion processing, $u^{(i_2)}$ selects the parameter $\lambda_j = 0.7759$ such that the new copy $\tilde{\mathbf{X}}_j^{(i_2)}$ has PSNR of 45 dB. From Fig. 1, when all colluders give each other correct information about their fingerprinted signals, $\{D_j(k,l)\}_{k,l\in SC}$ are from the same distribution with a single mean. If some selfish colluders process their fingerprinted copies before collusion, $\{D_j(k,l)\}_{k,l\in SC}$ are from different distributions with distinct means.

Let us define $\mathfrak{D}_j(SC_h, SC_h) \triangleq \{D_j(k,l) : k,l \in SC_h, k \neq l\}$ and $\mathfrak{D}_j(SC_s, SC_h) \triangleq \{D_j(k,l) : k \in SC_s, l \in SC_h\}$. From (4), the distance between $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_h, SC_s)$ depends on the selected parameter λ_j as well as the host video sequence. In Fig. 1(c), the sample means of $\mathfrak{D}_j(SC_h, SC_h)$, $\mathfrak{D}_j(i_1, SC_h)$, and $\mathfrak{D}_j(i_2, SC_h)$ are 4.7, 10.8, and 6.7, respectively. Thus, the difference between $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_h, SC_s)$ is larger when the selfish colluders select λ_j of smaller values. We also consider video sequences of different characteristics: "carphone" that has moderate motion and "flower" whose scene changes very fast. Fig. 2 shows the histogram of $\{D_j(k,l)\}$. Here, λ_j in (1) is fixed as 0.7. In Fig. 2(a), $\mathfrak{D}_j(SC_h, SC_h)$ has a sample mean of 4.7 and $\mathfrak{D}_j(i_1, SC_h)$ has a sample mean of 8.2. In Fig. 2(b), the sample means of $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(i_1, SC_h)$ are 15.8 and 108.2, respectively. Comparing Fig. 2(b) with Fig. 2(a), $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_h, SC_s)$ are separated further away from each other when the scene changes fast (for example, in sequence flower) since the norm $\|\Delta\mathbf{S}_j(\lambda_j)\|$ is larger.

The above analysis suggests that the histogram of $\{D_j(k,l)\}$ can be used to determine the existence of selfish colluders. The ringleader $\mathbf{R}$ calculates $D(k,l)$ for every pair of colluders $(u^{(k)}, u^{(l)})$ and broadcasts $\{D_j(k,l)\}$ to all colluders. If $\{D(k,l)\}$ are from the same distribution with a single mean, then all colluders keep their fair-collusion agreement and there are no selfish colluders. If $\{D(k,l)\}$ are from two or more distributions with different means, there exists at least one selfish colluder who applies precollusion processing.

In the above examples, $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ do not overlap, which enables honest colluders to easily detect the existence of selfish attackers. We now consider the scenario where $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ overlap. Let $D_j^{\max}(SC_h, SC_h) = \max(\mathfrak{D}_j(SC_h, SC_h))$ and $D_j^{\min}(SC_s, SC_h) = \min(\mathfrak{D}_j(SC_s, SC_h))$ be the largest and the smallest values in $\mathfrak{D}_j(SC_h, SC_h)$ and $\mathfrak{D}_j(SC_s, SC_h)$, respectively. Given the total number of colluders K, we define the overlap ratio as

$$c \triangleq \sum_{k,l\in SC} \frac{I\left[D_j^{\min}(SC_s, SC_h) \le D_j(k,l) \le D_j^{\max}(SC_h, SC_h)\right]}{K(K-1)/2} \tag{6}$$

where $I[\cdot]$ is the indicator function and the denominator is the total number of colluder pairs. The two distributions overlap by a larger ratio when c takes a larger value, and the two distributions do not overlap if $c = 0$, that is, $D_j^{\max}(SC_h, SC_h) < D_j^{\min}(SC_s, SC_h)$.

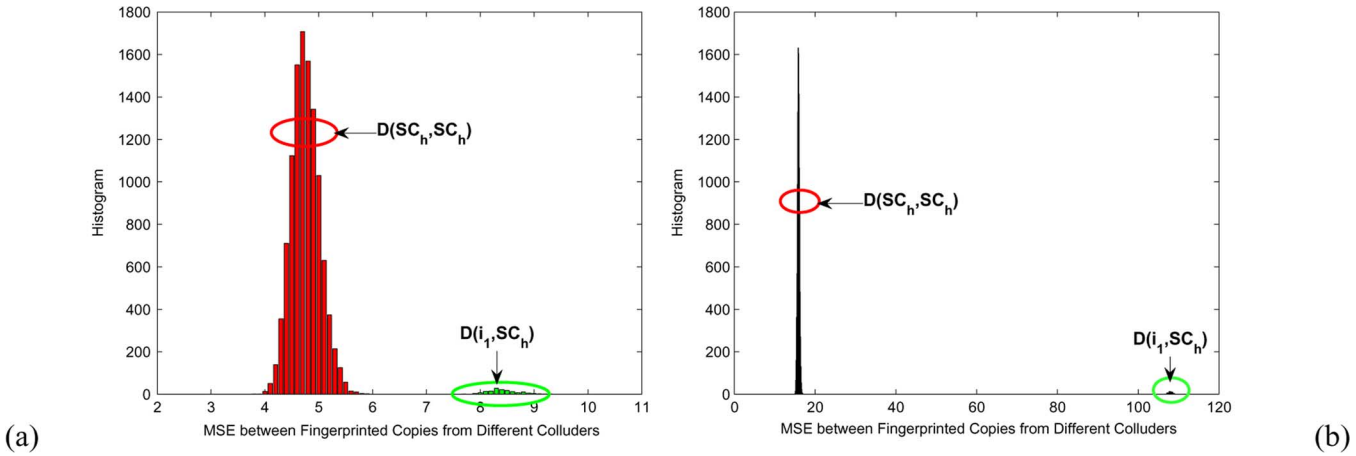

Fig. 2. Histogram of $\{D_j(k,l)\}_{k,l\in SC}$. (a) the second frame in sequence carphone. The length of the embedded fingerprint is 4179. (b) The second frame in sequence flower. The fingerprint is of length 23 010. Assume that there are a total of 150 colluders and $u^{(i_1)}$ is the only selfish colluder. Here, $u^{(i_1)}$ selects $\lambda_j = 0.7$ in (1).

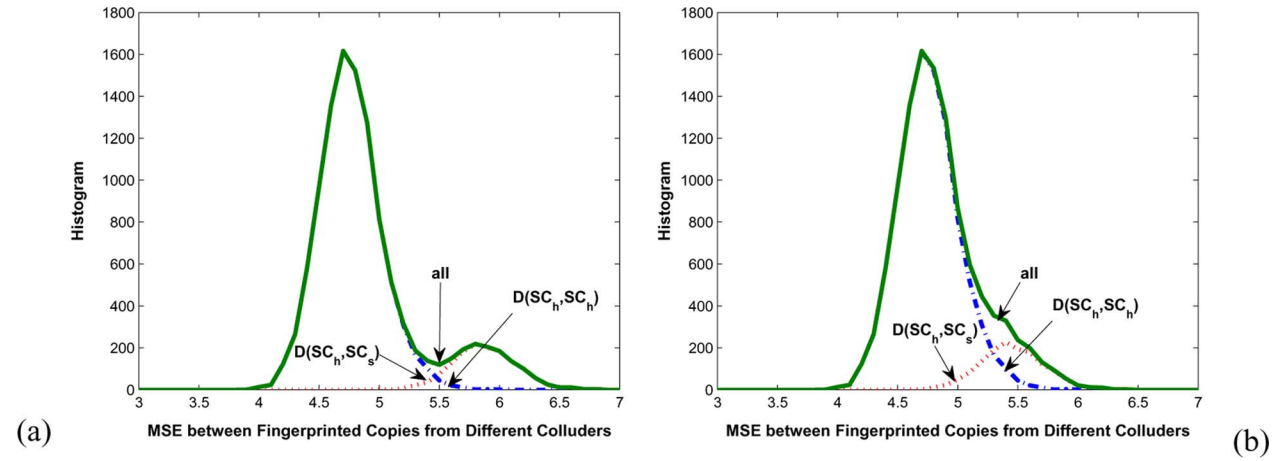

Fig. 3. Histogram of $\{D_j(k,l)\}_{k,l\in SC}$ for the second frame in sequence carphone with overlapping $\mathfrak{D}_j(SC_s,SC_h)$ and $\mathfrak{D}_j(SC_h,SC_h)$. $KC = 150$, and there are ten selfish colluders who process their copies independently. (a) $c = 20\%$. (b) $c = 75\%$.

We use the second frame in sequence carphone as an example, and assume that 10 colluders out of 150 colluders are selfish and process their fingerprinted copies independently before collusion. We observe a similar trend for other frames and other parameters. We intentionally move the two distributions $\mathfrak{D}_j(SC_s,SC_h)$ and $\mathfrak{D}_j(SC_h,SC_h)$ and let them overlap. Fig. 3(a) and (b) show the resulting histograms of $\{D_j(k,l)\}_{k,l\in SC}$ when $\mathfrak{D}_j(SC_s,SC_h)$ and $\mathfrak{D}_j(SC_h,SC_h)$ overlap by 20% and 75%, respectively. From Fig. 3, we can still observe the bimodality of $\{D_j(k,l)\}_{k,l\in SC}$ when c takes a small value. When $c \geq 75\%$, $\mathfrak{D}_j(SC_s,SC_h)$ and $\mathfrak{D}_j(SC_h,SC_h)$ merge together, which prevents the detection of selfish behavior. Thus, the bimodality of $\{D_j(k,l)\}$ can help detect the existence of precollusion processing when the overlap between $\mathfrak{D}_j(SC_s,SC_h)$ and $\mathfrak{D}_j(SC_h,SC_h)$ does not exceed 75%.

B. Identification of Selfish Colluders

After receiving the broadcasted $\{D_j(k,l)\}$ from the ringleader, colluders examine the histogram plot of $\{D_j(k,l)\}$ to determine the existence of selfish colluders. Further identification of selfish colluders requires detailed examination of $\{D_j(k,l)\}$, in particular, $\mathfrak{D}_j(SC_h,SC_s)$. For each $D_j(k,l) \in \mathfrak{D}_j(SC_h,SC_s)$, the two corresponding colluders, $u^{(k)}$ and $u^{(l)}$, are in different subgroups: one belongs to SC_h and the other is a selfish colluder in SC_s. Thus, analysis of each individual $D_j(k,l)$ in $\mathfrak{D}_j(SC_h,SC_s)$ can help separate SC into two subgroups and, therefore, enables selfish colluder identification.

To identify selfish colluders, a simple solution is to examine the histogram of all $\{D_j(k,l)\}$ and use a threshold to separate $\mathfrak{D}_j(SC_h,SC_s)$ from $\mathfrak{D}_j(SC_h,SC_h)$. However, the values of $\{D_j(k,l)\}$ change from sequence to sequence, and $\mathfrak{D}_j(SC_h,SC_s)$ and $\mathfrak{D}_j(SC_h,SC_h)$ may overlap. Thus, the thresholding-based method may introduce errors and thus affect the accuracy of the identification algorithm. To address this issue, from Fig. 3, a larger value of $D_j(k,l)$ gives higher confidence that $D_j(k,l)$ is in $\mathfrak{D}_j(SC_h,SC_s)$ and that the two corresponding colluders, $u^{(k)}$ and $u^{(l)}$, belong to different subgroups. Thus, our proposed algorithm starts with $D_j(k,l)$ that has the largest value (thus gives the detector the highest confidence) and determines which of the corresponding two colluders is selfish. Then, it moves to the next largest $D_j(k,l)$. It repeats this procedure until every colluder in SC has been identified either as a selfish colluder or an honest colluder. Thus, instead of using all $\{D_j(k,l)\}$, our proposed algorithm

only uses those that give higher confidence to accurately identify selfish colluders even when $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ overlap.

In addition, given $\{D_j(k,l)\}$, even if $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ do not overlap, the ringleader can only separate colluders into two subgroups, while he/she cannot tell which contains the honest colluders. Instead, an honest colluder knows that he/she is in SC_h, and given a $D_j(k,l)$ and the two corresponding colluders $u^{(k)}$ and $u^{(l)}$, he/she can immediately determine the subgroups that they are in. Therefore, in our proposed algorithm, the honest colluders themselves (instead of the ringleader) identify selfish colluders.

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

Set $\Psi_t = \{i\}$, $\mathbf{\Phi}^{(i)} = -\mathbf{1}_{1\times KC}$, $\Phi^{(i)}(i) = 0$, and $m = 0$;
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 colluders;
  if $k \notin \Psi_t$ *AND* $l \notin \Psi_t$ **then**
    if $D_j(i,k) > D_j(i,l)$ **then**
      $\Phi^{(i)}(k) = 1; \Phi^{(i)}(l) = 0; \Psi_t = \Psi_t \cup \{k,l\}$;
    else if $D_j(i,k) < D_j(i,l)$ **then**
      $\Phi^{(i)}(k) = 0; \Phi^{(i)}(l) = 1; \Psi_t = \Psi_t \cup \{k,l\}$;
  else if $k \in \Psi_t$ *AND* $l \notin \Psi_t$ **then**
    $\Phi^{(i)}(l) = 1 - \Phi^{(i)}(k)$, $\Psi_t = \Psi_t \cup \{l\}$;
  else if $l \in \Psi_t$ *AND* $k \notin \Psi_t$ **then**
    $\Phi^{(i)}(k) = 1 - \Phi^{(i)}(l)$, $\Psi_t = \Psi_t \cup \{k\}$;
end
return $\widehat{SC}_s^{(i)} = \{k : \Phi^{(i)}(k) = 1\}$.

Algorithm 1 gives the details of how $u^{(i)}$ in SC_h identifies selfish colluders. For a total of KC colluders whose indices are $i_1, i_2, \ldots, i_{KC}$, $\mathbf{\Phi}^{(i)} = [\Phi^{(i)}(i_1), \Phi^{(i)}(i_2), \ldots, \Phi^{(i)}(i_{KC})]$. Colluder $u^{(i)}$ sets $\Phi^{(i)}(k) = 1$ when he/she detects that $u^{(k)}$ is a selfish colluder, and $\Phi^{(i)}(k) = 0$ if $u^{(i)}$ believes that $u^{(k)}$ is an honest colluder. The set $\Psi_t = \{k : \Phi^{(i)}(k) \neq -1\}$ includes the indices of all colluders that $u^{(i)}$ has identified which subgroups they belong to in the previous rounds.

Colluder $u^{(i)}$ first initializes $\mathbf{\Phi}^{(i)}$ to an undetermined status -1 and sets $\Phi^{(i)}(i)$ to 0 since he/she is in subgroup SC_h. Then, $u^{(i \in SC_h)}$ examines every $D_j(k,l)$ and starts with the largest one. Given a $D_j(k,l)$, $u^{(i)}$ first checks if he/she has determined the values of $\Phi^{(i)}(k)$ and $\Phi^{(i)}(l)$ in the previous rounds.

- If both $\Phi^{(i)}(k)$ and $\Phi^{(i)}(l)$ have been decided, $u^{(i)}$ 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 with value -1, without loss of generality, assume that $\Phi^{(i)}(k)$ has been determined previously, then $u^{(i)}$ sets $\Phi^{(i)}(l) = 1 - \Phi^{(i)}(k)$.
- If $u^{(i)}$ is unable to determine either $\Phi^{(i)}(k)$ or $\Phi^{(i)}(l)$ in the previous rounds, he/she then compares the values of $D_j(k,i)$ and $D_j(l,i)$. Without loss of generality, assume that $D_j(k,i) > D_j(l,i)$. In this scenario, compared with $u^{(l)}$, $u^{(k)}$ is more likely to be a selfish colluder. Thus, $u^{(i)}$ sets $\Phi^{(i)}(l) = 0$ and $\Phi^{(i)}(k) = 1$.

Colluder $u^{(i)}$ repeats the above process and stops when $\Psi_t = SC$ and all the components in $\mathbf{\Phi}^{(i)}$ have been set to either 0 or 1. Algorithm 1 outputs $\widehat{SC}_s^{(i)} = \{k : \Phi^{(i)}(k) = 1\}$, which is the set containing the indices of all colluders whom $u^{(i)}$ detects as selfish colluders.

C. Selfish Colluder Detection and Identification and Performance Evaluation

1) Selfish Colluder Detection and Identification: To summarize, if the colluders' social network has a centralized structure with a trusted ringleader, the key steps in the selfish colluder detection and identification process are: for each frame j:

Step 1) **Encryption:** Each colluder $u^{(i)}$ first generates a secret key $K^{(i)}$ shared with the ringleader only, encrypt his/her fingerprinted copy with $K^{(i)}$, and transmits the encrypted copy to the ringleader.

Step 2) **Calculation of $\{D\}$:** After decrypting the bit streams received from all colluders, the ringleader calculates $D_j(k,l)$ for each pair of colluders $(u^{(k)}, u^{(l)})$. The ringleader then broadcasts $\{D_j(k,l)\}$ to all colluders, together with his/her digital signature [25].

Step 3) **Detection of Precollusion Processing:** Colluders in SC_h first examine the histogram of $\{D_j\}$ to detect precollusion processing. If $\{D_j\}$ are from the same distribution with a single mean, then there are no selfish colluders, and the colluders skip Step 4 and collude with each other. If $\{D_j\}$ are from two or more distributions with different means, there is at least one selfish colluder and honest colluders go to Step 4 to identify selfish colluders.

Step 4) **Selfish Colluder Identification:** If Step 3 detects the existence of selfish colluders, each honest colluder in SC_h applies Algorithm 1 to estimate the identities of the selfish colluders.

2) Performance Evaluation: In the above selfish colluder detection and identification process, all the fingerprinted copies are encrypted during transmission. For each copy, only the corresponding user and the trusted ringleader can access the fingerprinted coefficients, while other colluders do not have the decryption key and cannot decrypt the transmitted bit stream. Therefore, $\gamma_j = 0$ and the selfish colluder detection and identification process is robust against framing attacks.

To evaluate the detection performance of the proposed algorithm, we select three typical video sequences, "miss america," "carphone," and "flower," and test on the first ten frames in each sequence as an example. Other frames and other sequences give the same result. The simulation setup is the same as that in Section III-A. Orthogonal fingerprints are generated from Gaussian distribution $\mathcal{N}(0, \sigma_W^2)$ with $\sigma_W^2 = 1/9$. In each fingerprinted copy, fingerprints that are embedded into neighboring frames are correlated with each other, depending on the similarity between the host frames. Human visual model-based spread spectrum embedding [18] is applied to embed fingerprints into the host signal. We assume that the total number of colluders is 150. There are ten selfish colluders and each processes his/her fingerprinted copy independently before collusion. Among the ten selfish colluders, five of them select the parameter λ_j in (1) to generate new frames with PSNR of 40 dB, and the other five selfish colluders generate new frames with PSNR of 45 dB.

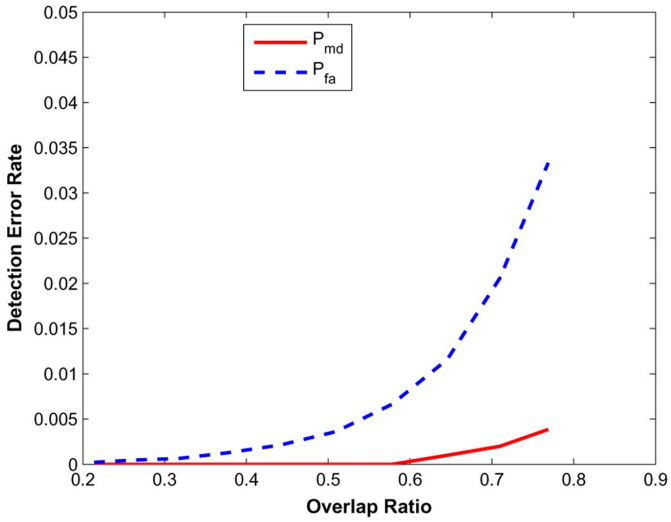

Fig. 4. Performance of the selfish colluder identification algorithm with overlapping $\mathfrak{D}(SC_h, SC_s)$ and $\mathfrak{D}(SC_h, SC_h)$. $KC = 150$.

For each frame in every sequence, we run 1000 simulation runs to test the performance of the proposed algorithm. In all our simulation runs, Algorithm 1 accurately identifies all selfish colluders in SC_s without falsely accusing any honest colluder as selfish, and the proposed selfish colluder detection and identification algorithm does not make either type of detection errors. This is because, temporal filtering in (1) not only averages fingerprints embedded in adjacent frames and reduces the selfish colluder's risk, it also filters adjacent host frames and introduces extra distortion into the host signal. This extra distortion makes the two distributions, $\mathfrak{D}(SC_h, SC_s)$ and $\mathfrak{D}(SC_h, SC_h)$ in Fig. 1 separate from each other, and it enables the proposed algorithm to correctly identify the selfish colluders without falsely accusing any others.

We then consider the scenario where $\mathfrak{D}(SC_h, SC_s)$ and $\mathfrak{D}(SC_h, SC_h)$ overlap with each other, and Fig. 4 shows the simulation results of the proposed algorithm. We use the second frame of carphone as an example, and assume that there are ten selfish colluders who process their copies independently. We observe the same trend for other frames and other parameters. In Fig. 4, we stop the simulations when $c \geq 75\%$, since in those scenarios, $\mathfrak{D}(SC_h, SC_s)$ and $\mathfrak{D}(SC_h, SC_h)$ merge together and the bimodality of $\{D_j(k,l)\}$ cannot be observed. From Fig. 4, the miss detection rate is below 0.5% and the false alarm rate does not exceed 3.5%, and our proposed algorithm can accurately identify selfish colluders even if the two distributions overlap.

IV. Distributed Peer-Structured Colluder Social Networks

When there is not such a trusted ringleader, colluders form a peer-structured social network and they help each other detect and identify selfish colluders. In this section, we consider the scenario where there are only a few selfish colluders, and study autonomous selfish colluder identification. We also address potential attacks on the proposed autonomous identification scheme, and analyze its attack resistance.

A. Selfish Colluder Detection and Identification Without a Trusted Ringleader

Without a trusted ringleader, the challenge is to accurately calculate $\{D_j(k,l)\}$ while still protecting the secrecy of the fingerprinted coefficients. In this section, we will first study how to calculate $D_j(k,l)$ for a given pair of colluders $(u^{(k)}, u^{(l)})$ without a trusted ringleader. Then, we will investigate autonomous selfish colluder identification for a group of colluders.

1) Calculation of $D_j(k,l)$: For each pair of colluders $(u^{(k)}, u^{(l)})$, assume that $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$ are the fingerprinted copies from $u^{(k)}$ and $u^{(l)}$, respectively. Colluder $u^{(k)}$ and $u^{(l)}$ cannot calculate $D_j(k,l)$ themselves, since it will leak the fingerprinted coefficients in $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$ to each other and violate the anti-framing requirement. Thus, without a trusted ringleader, they have to find a third colluder $u^{(i)}$ to help them. To prevent $u^{(i)}$ from accessing the fingerprinted coefficients in these two copies, $u^{(k)}$ and $u^{(l)}$ should process their fingerprinted copies first, and let $u^{(i)}$ calculate $D_j(k,l)$ from the processed copies.

Define $f(\cdot)$ as the function that $u^{(k)}$ and $u^{(l)}$ use to process their copies, and let $\mathbf{Y}_j^{(k)}$ and $\mathbf{Y}_j^{(l)}$ be the processed copies of $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$, respectively. To enable $u^{(i)}$ to calculate $D_j(k,l)$ from $\mathbf{Y}_j^{(k)}$ and $\mathbf{Y}_j^{(l)}$, it is required that $f(\cdot)$ does not change the MSE between these two copies and

$$\tilde{D}_j(k,l) = \left\| \mathbf{Y}_j^{(k)} - \mathbf{Y}_j^{(l)} \right\|^2 = \left\| \tilde{\mathbf{X}}_j^{(k)} - \tilde{\mathbf{X}}_j^{(l)} \right\|^2 = D_j(k,l). \tag{7}$$

In addition, it is required that given $\mathbf{Y}_j^{(k)}$ and $\mathbf{Y}_j^{(l)}$, $u^{(i)}$ cannot estimate the fingerprinted coefficients in $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$.

In this paper, we use a simple component-wise addition-based method to process $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$. Other methods that protect the fingerprinted coefficients and satisfy (7) (for example, the isometry rotation and the permutation-complement-based encryption [26]) can also be applied. Assume that $\tilde{\mathbf{X}}_j^{(k)}$ and $\tilde{\mathbf{X}}_j^{(l)}$ are of length N_j. Given a key $K^{k,l}$ shared by $u^{(k)}$ and $u^{(l)}$ only, they use $K^{k,l}$ as the seed of the pseudo random number generator and generate a random sequence $\mathbf{v}_j^{(k,l)}$ of length N_j. The N_j components in $\mathbf{v}_j^{(k,l)}$ are independent and identically distributed (i.i.d.) and uniformly distributed in $[-\mathcal{U}, \mathcal{U}]$. Then, $u^{(k)}$ and $u^{(l)}$ add $\mathbf{v}_j^{(k,l)}$ to their fingerprinted copies component by component, and calculate

$$\mathbf{Y}_j^{(k)} = f\left(\tilde{\mathbf{X}}_j^{(k)}, K^{k,l}\right) = \tilde{\mathbf{X}}_j^{(k)} + \mathbf{v}_j^{(k,l)} \text{ and}$$
$$\mathbf{Y}_j^{(l)} = f\left(\tilde{\mathbf{X}}_j^{(l)}, K^{k,l}\right) = \tilde{\mathbf{X}}_j^{(l)} + \mathbf{v}_j^{(k,l)} \tag{8}$$

respectively. Thus, $\|\mathbf{Y}_j^{(k)} - \mathbf{Y}_j^{(l)}\|^2 = \|\tilde{\mathbf{X}}_j^{(k)} + \mathbf{v}_j^{(k,l)} - \tilde{\mathbf{X}}_j^{(l)} - \mathbf{v}_j^{(k,l)}\|^2 = \|\tilde{\mathbf{X}}_j^{(k)} - \tilde{\mathbf{X}}_j^{(l)}\|^2$, and (7) is satisfied. To hide information of the embedded fingerprints, colluders should select a large $\mathcal{U}$ and let the random sequence $\mathbf{v}_j^{(k,l)}$ have large amplitude.

Let $\mathrm{Enc}(X, K)$ denote the encryption of message X with key K. As shown in Fig. 5, to calculate $D_j(k,l)$:

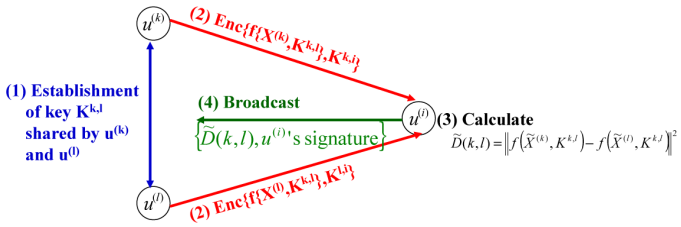

Fig. 5. Calculation of $D(k,l)$ without a trusted ringleader. $u^{(i)}$ is selected to help $u^{(k)}$ and $u^{(l)}$ calculate $D(k,l)$. The notation $\mathrm{Enc}\{X,K\}$ denotes the encryption of message X using key K.

- colluder $u^{(k)}$ and $u^{(l)}$ first generate a secret key $K^{k,l}$. Then, $u^{(k)}$ generates a secret key $K^{k,i}$ shared with $u^{(i)}$ only, and $K^{l,i}$ is a key shared by $u^{(l)}$ and $u^{(i)}$.
- Colluder $u^{(k)}$ first processes his/her fingerprinted copy $\tilde{\mathbf{X}}_j^{(k)}$ using (8), then encrypts it with key $K^{k,i}$ to protect the fingerprinted coefficients in $\tilde{\mathbf{X}}_j^{(k)}$. (Similar to the scenario with a trusted ringleader, encryption here is only used to secure communications between two parties and to prevent eavesdropping, and it will not affect the later steps in selfish colluder identification as well as multiuser collusion.) Then, $u^{(k)}$ transmits the encrypted copy $\mathrm{Enc}(f(\tilde{\mathbf{X}}_j^{(k)}, K^{k,l}), K^{k,i})$ to $u^{(i)}$. $u^{(l)}$ repeats the same process.
- Colluder $u^{(i)}$ calculates $\tilde{D}_j(k,l) = \|f(\tilde{\mathbf{X}}_j^{(k)}, K^{k,l}) - f(\tilde{\mathbf{X}}_j^{(l)}, K^{k,l})\|^2$, and broadcasts $\tilde{D}_j(k,l)$ together with his/her digital signature.

2) Autonomous Detection and Identification of Selfish Colluders: To extend the above algorithm to a group of colluders, for each frame j in the video sequence:

- Colluders randomly divide themselves into two subgroups SC_1 and SC_2, where $SC_1 \cup SC_2 = SC$ and $SC_1 \cap SC_2 = \emptyset$.[2] Colluders in SC_1 randomly select an *assistant* $u^{(i_1 \in SC_1)}$ to help colluders in SC_2 calculate $\{D_j(k,l)\}_{k,l \in SC_2}$. Similarly, $u^{(i_2 \in SC_2)}$ is randomly selected to help colluders in SC_1 calculate $\{D_j(k,l)\}_{k,l \in SC_1}$.
- Assume that K^{SC_1} is a key that is shared by colluders in SC_1. Each colluder $u^{(l)}$ in SC_1 generates a secret key K^{l,i_2} shared with the selected assistant $u^{(i_2 \in SC_2)}$. Then, $\mathbf{u}^{(l)}$ uses (8) to process his/her fingerprinted copy $\tilde{\mathbf{X}}_j^{(l)}$ and generates $f(\tilde{\mathbf{X}}_j^{(l)}, K^{SC_1})$. Then, $u^{(l)}$ encrypts his/her copy with key K^{l,i_2} and transmits the encrypted version $\mathrm{Enc}(f(\tilde{\mathbf{X}}_j^{(l)}, K^{SC_1}), K^{l,i_2})$ to the selected assistant $u^{(i_2)}$ in SC_2. Colluders in SC_2 follow the same procedure, process and encrypt their fingerprinted copies, and transmit them to the selected assistant $u^{(i_1)}$ in SC_1.
- After decrypting the bit streams received from all colluders in SC_1, for each pair of colluders $(u^{(k)}, u^{(l)})$ in subgroup SC_1, the selected assistant $\mathbf{u}^{(i_2 \in SC_2)}$ calculates $\tilde{D}_j(k,l) = \|f(\tilde{\mathbf{X}}_j^{(k)}, K^{SC_1}) - f(\tilde{\mathbf{X}}_j^{(l)}, K^{SC_1})\|^2$. Then, $u^{(i_2)}$ broadcasts $\{\tilde{D}_j(k,l)\}_{k,l \in SC_1}$ to colluders in SC_1, together with his/her digital signature. Note that $u^{(i_2)}$ only calculates $\tilde{D}_j(k,l)$ where both k and l are in subgroup SC_1. The selected assistant $u^{(i_1 \in SC_1)}$ in subgroup SC_1 repeats the same process to help colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}$ for all $k,l \in SC_2$.
- Given $\{\tilde{D}_j(k,l)\}_{k,l \in SC_1}$, colluders in SC_1 apply the same method as in Section III.C.1 to detect and identify selfish colluders in SC_1. Similarly, colluders in SC_2 examine $\{\tilde{D}_j(k,l)\}_{k,l \in SC_2}$ and identify selfish colluders in SC_2.

Finally, honest colluders combine the detection results from all frames in the sequence, and exclude those identified selfish colluders from collusion.

[2]We use 2 subgroups as an example, and the proposed algorithm can be easily extended to scenarios with more than two subgroups.

B. Performance of the Autonomous Selfish Colluder Detection and Identification Scheme

In this section, we investigate how selfish colluders can *actively* attack the proposed algorithm and manipulate the detection results in order to avoid being detected. We also propose techniques to ensure accurate identification of selfish colluders even under such attacks. Here, we consider the scenario where there are only a limited number of selfish colluders. We assume that if honest colluders are selected as assistants to help calculate $\{\tilde{D}_j\}$, they will give others correct values of $\{\tilde{D}_j(k,l)\}$.

1) A Group of Selfish Colluders: The performance of the proposed selfish colluder detection and identification algorithm depends on the correctness of $\{\tilde{D}_j(k,l)\}$. If all the selected assistants give the other colluders correct values of $\{\tilde{D}_j(k,l)\}$, the above autonomous selfish colluder detection and identification scheme has the same performance as that in Section III-C2, and honest colluders can correctly identify selfish colluders in SC_s without falsely accusing others. However, during the autonomous selfish colluder detection and identification process, it is possible that two or more selfish colluders collaborate with each other to change the detection results. Fig. 6 shows an example.

In Fig. 6, the simulation setup is the same as that in Fig. 1(b). We assume that there are two selfish colluders $u^{(i_1)}$ and $u^{(i_2)}$, and they are in different subgroups during the autonomous selfish colluder detection and identification process. Without loss of generality, assume that $i_1 \in SC_1$ and $i_2 \in SC_2$. In SC_2, there are $K_2 = 75$ colluders and we assume that all the other 74 colluders in SC_2 do not process their received copies. Fig. 6(a) plots the unchanged histogram of $\{D_j(k,l)\}_{k,l \in SC_2}$, from which Algorithm 1 can correctly identify colluder i_2 as a selfish colluder. If $u^{(i_1)}$ is selected as the assistant to help colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}_{k,l \in SC_2}$, $u^{(i_1)}$ can modify the values of $\{\tilde{D}_j(k,l)\}_{k,l \in SC_2}$ and let them be from the same distribution, for example, as shown in Fig. 6(b). Then, Algorithm 1 can not identify $u^{(i_2)}$ as a selfish colluder and it makes a miss-detection error. The selfish colluder $u^{(i_1)}$ can also change the values of $\{\tilde{D}_j(k,l)\}_{k,l \in SC_2}$ and let the histogram be the same as in Fig. 6(c). Here, Algorithm 1 not only misses the real selfish colluder $u^{(i_2)}$, it also falsely accuses another two honest colluders, $u^{(i_3)}$ and $u^{(i_4)}$. Using the same method, $u^{(i_2)}$ can also prevent colluders from detecting $u^{(i_1)}$'s precollusion processing, or make them falsely accuse honest colluders as selfish.

2) Multiple Assistants for Each Subgroup: To reduce the probability that these selfish colluders can successfully change

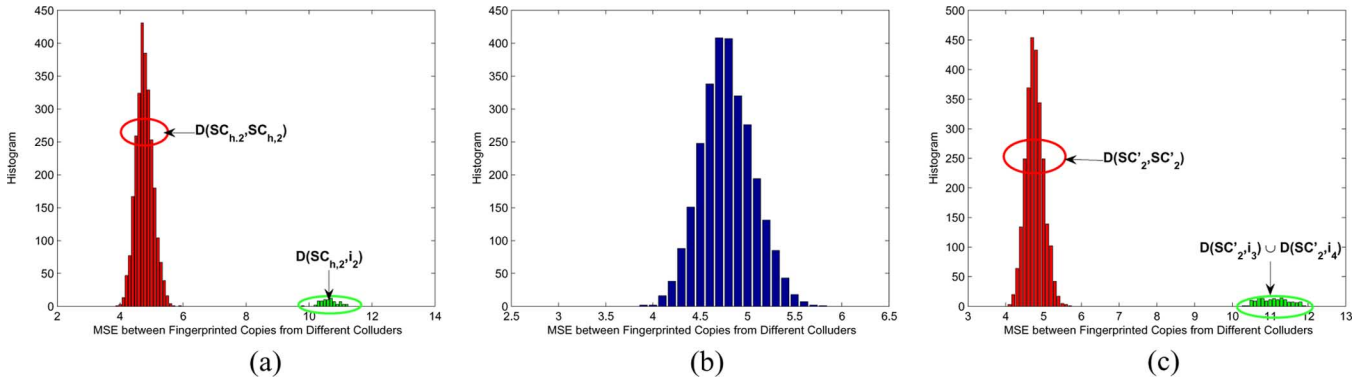

Fig. 6. Histogram plots of $\{\bar{D}_j(k,l)\}_{k,l\in SC_2}$ for the second frame in sequence carphone. There are 75 colluders in SC_2, and one of them, colluder i_2, applies temporal filtering (1) with $\lambda_j = 0.6031$ before collusion. (a) Histogram of the unchanged $\{\mathbf{D}_j(k,l)\}_{k,l\in SC_2}$. $SC_{h,2} = SC_2\backslash\{i_2\}$. Here, Algorithm 1 can correctly identify the selfish colluder $u^{(i_2)}$. (b) Histogram of the manipulated $\{\mathbf{D}_j(k,l)\}_{k,l\in SC_2}$. In this scenario, Algorithm 1 fails to detect the selfish colluder $u^{(i_2)}$. (c) Histogram of the manipulated $\{\mathbf{D}_j(k,l)\}_{k,l\in SC_2}$. Colluder $u^{(i_3)}$ and $u^{(i_4)}$ did not apply precollusion processing, but they are falsely accused by Algorithm 1 to be selfish. $SC'_2 = SC_2\backslash\{i_3,i_4\}$.

the detection results, in each subgroup, a straightforward solution is to select multiple assistants to calculate $\{\tilde{D}_j\}$ and use majority vote when identifying selfish colluders.

For each frame j, same as in Section IV-A2, the colluders first randomly divide themselves into two non-overlapping subgroups SC_1 and SC_2. To detect and identify selfish colluders in SC_1:

- m colluders are randomly selected from SC_2 to help calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_1}$, and $\mathbf{A}_j(SC_2) = \{i_{2,1}, i_{2,2}, \ldots, i_{2,m}\}$ contains their indices.
- For each selected assistant $i_{2,n} \in \mathbf{A}_j(SC_2)$, colluders in SC_1 follow Step 2 in Section IV-A2, process and encrypt their fingerprinted copies, and transmit them to $u^{(i_{2,n})}$.
- Each selected assistant $i_{2,n}$ in $\mathbf{A}_j(SC_2)$ follows Step 3 in Section IV-A2 to calculate $\tilde{D}_j^{i_{2,n}}(k,l) = \|f(\tilde{\mathbf{X}}_j^{(k)}, K^{SC_1}) - f(\tilde{\mathbf{X}}_j^{(l)}, K^{SC_1})\|^2$ for all $k,l \in SC_1$, and broadcasts the results to colluders in SC_1 together with $u^{(i_{2,n})}$'s digital signature.
- For every honest colluder $u^{(k)}$ in SC_1, given $\{\tilde{D}_j^{i_{2,n}}(k,l)\}_{k,l\in SC_1}$ received from the assistant $u^{(i_{2,n})}$ in $\mathbf{A}_j(SC_2)$, $u^{(k)}$ follows Step 4 in Section IV-A2, examines the histogram of $\{\tilde{D}_j^{i_{2,n}}(k,l)\}_{k,l\in SC_1}$, and uses Algorithm 1 to detect and identify selfish colluders. For every $l \in SC_1$ and for each $i_{2,n} \in \mathbf{A}_j(SC_2)$, $u^{(k)}$ sets $\upsilon_j^{(k)}(n,l) = 1$ if Algorithm 1 identifies $u^{(l\in SC_1)}$ as a potential selfish colluder from $\{\tilde{D}_j^{i_{2,n}}(k,l)\}_{k,l\in SC_1}$, and $\upsilon_j^{(k)}(n,l) = 0$ otherwise. Then $u^{(k)}$ combines the m detection results $\{\upsilon_j^{(k)}(n,l)\}_{n=1,\ldots,m}$ and uses majority vote to determine whether $u^{(l)}$ is a selfish colluder. If $\sum_{n=1}^{m}\upsilon_j^{(k)}(n,l) \geq \lceil m/2\rceil$, $u^{(k)}$ believes that $u^{(l)}$ processed his/her copy before collusion and sets $\Upsilon_j^{(k)}(l) = 1$. $\Upsilon_j^{(k)}(l) = 0$ otherwise.

The same procedure is used to identify selfish colluders in SC_2.

To further improve the performance of the selfish colluder detection and identification algorithm, colluders in SC_h should jointly consider the detection results from all frames in the video sequence when making the final decision on the identities of the selfish colluders.

For each frame j in the video sequence, in Step 1 of the autonomous selfish colluder detection and identification, define

$$I_j(k,l) \triangleq \begin{cases} 1 & \text{if } k \text{ and } l \text{ are in the same subgroup,} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

For every pair of colluders $(u^{(k)}, u^{(l)})$, we further define $F(k,l) \triangleq \{j : I_j(k,l) = 1\}$, which contains the indices of all frames where $u^{(k)}$ and $u^{(l)}$ are assigned to the same subgroup.

For an honest colluder $u^{(k\in SC_h)}$, to determine whether $\tilde{\mathbf{X}}^{(l)}$ is the original copy that $u^{(l)}$ received from the content owner, $u^{(k)}$ jointly considers all the detection results $\{\Upsilon_j^{(k)}(l)\}_{j\in F(k,l)}$ that he/she has, and considers $u^{(l)}$ as a selfish colluder if the average of $\{\Upsilon_j^{(k)}(l)\}_{j\in F(k,l)}$ is above a predetermined threshold α. $u^{(k)}$ then outputs the estimated selfish colluder set

$$\widehat{SC}_s^{(k)} = \left\{ l : \frac{\sum_{j\in F(k,l)} \Upsilon_j^{(k)}(l)}{|F(k,l)|} > \alpha \right\}. \tag{10}$$

A larger α helps lower the false alarm rate at the cost of a higher miss detection rate, and the selection of the parameter α should address the tradeoff between the false alarm and the miss detection rates.

3) Performance Analysis: In this section, to address the unique issues in autonomous selfish colluder identification, we investigate how such a group of selfish colluders can manipulate the detection results and how it affects the performance of the autonomous selfish colluder detection and identification. For each frame in the video sequence, define P_{cs} as the probability that the group of selfish colluders can successfully manipulate the detection results and intentionally let others make errors when detecting selfish behavior. In this section, we first analyze P_{cs}, and we then study how it affects the detection error rates.

a) Terminology Definition: Assume that there are a total of KC colluders. For each frame j, during the autonomous selfish colluder detection and identification process, assume that the number of colluders in subgroup SC_1 and SC_2 are KC_1 and KC_2, respectively, with $KC_1 + KC_2 = KC$. $\mathbf{A}_j(SC_1)$ is the set with the indices of the m assistants in SC_1 selected to help

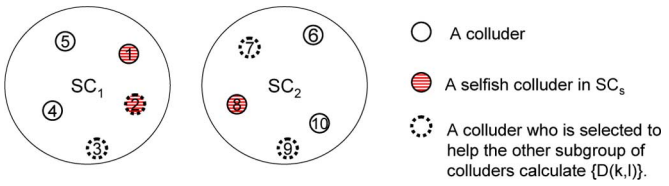

Fig. 7. An example to illustrate the terms defined in Section IV-B3a. In this example, there are ten colluders and $SC = \{1,2,\ldots,10\}$. $SC_1 = \{1,2,3,4,5\}$ and $SC_2 = \{6,7,8,9,10\}$. $SC_s = \{1,2,8\}$ contains the indices of the selfish colluders and $KC_s = 3$. Among these three selfish colluders, colluders 1 and 2 are in SC_1 and colluder 8 is in SC_2. Therefore, $KC_s(SC_1) = 2$ and $KC_s(SC_2) = 1$. In SC_1, colluders 2 and 3 are selected as assistants to help SC_2 calculate $\{D_j(k,l)\}$ and $\mathbf{A}_j(SC_1) = \{2,3\}$. $\mathbf{A}_j(SC_2) = \{7,9\}$, and colluders 7 and 9 are selected to help SC_1 calculate $\{D_j\}$. In this example, $\mathbf{C}_s \cap \mathbf{A}_j(SC_1) = \{2\}$ and $KC_{\rm as}(SC_1) = 1$. $\mathbf{C}_s \cap \mathbf{A}_j(SC_2) = \emptyset$ and $KC_{\rm as}(SC_2) = 0$.

colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$; and $\mathbf{A}_j(SC_2)$ contains the indices of the m assistants selected to help colluders in SC_1 calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_1}$.

Let $\mathbf{C}_s$ denote the set with the indices of the selfish colluders who collaborate with each other to avoid being detected by their fellow colluders, and $KC_s = |\mathbf{C}_s|$ is its size.[3] Among the KC_s selfish colluders, $KC_s(SC_1) = |\mathbf{C}_s \cap SC_1|$ of them are in subgroup SC_1, and the other $KC_s(SC_2) = |\mathbf{C}_s \cap SC_2|$ selfish colluders are in SC_2. We have $KC_s(SC_1) + KC_s(SC_2) = KC_s$ and $0 \le KC_s(SC_1), KC_s(SC_2) \le KC_s$. For frame j, we further define $KC_{\rm as}(SC_1) = |\mathbf{C}_s \cap \mathbf{A}_j(SC_1)|$ as the number of selfish colluders in SC_1 that are selected as assistants to help calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$, and $KC_{\rm as}(SC_2) = |\mathbf{C}_s \cap \mathbf{A}_j(SC_2)|$ is the number of selfish colluders in SC_2 that are selected to help calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_1}$. Fig. 7 gives an example of the above defined terms.

b) Analysis of P_{cs}: In this paper, we consider the scenario where $KC_s \ll KC_1$ and $KC_s \ll KC_2$. For subgroup SC_1, among the m selected assistants in $\mathbf{A}_j(SC_2)$, if more than half of them are from $\mathbf{C}_s$ (i.e., $KC_{\rm as}(SC_2) \ge \lceil m/2 \rceil$), even if colluders in SC_1 apply majority vote as in Section IV-B2, the selfish colluders can still change the values of $\{\tilde{D}_j(k,l)\}_{k,l\in SC_1}$ and successfully cause others make detection errors when identifying selfish colluders using frame j; the same for subgroup SC_2. Therefore, for each frame in the video sequence, the selfish colluders can change the detection results if and only if either $KC_{\rm as}(SC_1) \ge \lceil m/2 \rceil$ or $KC_{\rm as}(SC_2) \ge \lceil m/2 \rceil$. Define $G_{SC_1}(p) \triangleq \{KC_s(SC_1) = p\}$ as the event that in subgroup SC_1, there are p selfish colluders from $\mathbf{C}_s$, i.e., $KC_s(SC_1) = p$. We have

$$P_{cs} = \sum_{p=0}^{KC_s} P\Big[(KC_{\rm as}(SC_1) \ge \lceil m/2 \rceil) \bigcup (KC_{\rm as}(SC_2) \ge \lceil m/2 \rceil) \,|\, G_{SC_1}(p)\Big] \times P[G_{SC_1}(p)]$$

$$= \sum_{p=0}^{KC_s} \left\{ 1 - \left(\sum_{p_1=0}^{\min\{\lceil m/2 \rceil - 1, p\}} P[KC_{\rm as}(SC_1) = p_1 \,|\, G_{SC_1}(p)] \right) \times \left(\sum_{p_2=0}^{\min\{\lceil m/2 \rceil - 1, KC_s - p\}} P[KC_{\rm as}(SC_2) = p_2 \,|\, G_{SC_1}(p)] \right) \right\} \times P[G_{SC_1}(p)]. \quad (11)$$

In (11), for $0 \le p \le KC_s$, $0 \le p_1 \le \min\{m, p\}$ and $0 \le p_2 \le \min\{m, KC_s - p\}$, we have

$$P[KC_{\rm as}(SC_1) = p_1 \mid G_{SC_1}(p)] = \binom{p}{p_1}\binom{KC_1 - p}{m - p_1} \Big/ \binom{KC_1}{m},$$

$$P[KC_{\rm as}(SC_2) = p_2 \mid G_{SC_1}(p)] = \binom{KC_s - p}{p_2}\binom{KC_2 - (KC_s - p)}{m - p_2} \Big/ \binom{KC_2}{m},$$

and

$$P[G_{SC_1}(p)] = \binom{KC_s}{p}\binom{KC - KC_s}{KC_1 - p} \Big/ \binom{KC}{KC_1}. \quad (12)$$

Fig. 8(a) and (b) plots the simulation results of P_{cs} with a total of $KC = 50$ and $KC = 150$ colluders, respectively. In our simulations, we let SC_1 and SC_2 be of the same size and $KC_1 = KC_2 = KC/2$. From Fig. 8, selecting multiple assistants in each subgroup significantly reduces P_{cs}. For example, when 10% of the colluders are selfish colluders in $\mathbf{C}_s$, choosing $m = 3$ assistants from each subgroup helps lower P_{cs} from 0.2 to 0.05 when compared with the scenario with $m = 1$. In addition, P_{cs} is larger when there are more selfish colluders in $\mathbf{C}_s$.

c) Simulation Results of $P_{\rm fa}$ and $P_{\rm md}$: The above analysis considers one frame in the video sequence. This section studies the performance of the proposed algorithm when the detection results from all frames are jointly considered to identify selfish colluders.

We test on the first 300 frames of sequence carphone, and our simulation setup is the same as that in Section III-C2. Human visual model based spread spectrum embedding [18] is used to embed fingerprints into the host signal, and orthogonal fingerprints are assigned to different users. During precollusion processing, selfish colluders select λ_j in (1) such that the newly generated frames have PSNR of 40 dB when compared with the originally received ones. Each selfish colluder processes his/her copy independently.

For each frame in the video sequence, each subgroup selects $m = 3$ assistants to help the other subgroup calculate $\{\tilde{D}_j(k,l)\}$, and they apply majority vote to identify the selfish colluders. We assume that if selected as assistants to help calculate $\{\tilde{D}_j(k,l)\}$, honest colluders tell other colluders correct values of $\{\tilde{D}_j(k,l)\}$. We further assume that $\mathbf{C}_s = SC_s$, and all selfish colluders who apply precollusion processing collaborate

[3]Note that SC_s contains all selfish colluders who apply precollusion processing to further lower their own risk of being detected by the fingerprint detector; while $\mathbf{C}_s$ includes those who work together during the selfish colluder identification process to avoid being detected by their fellow colluders. $\mathbf{C}_s \subseteq SC_s$.

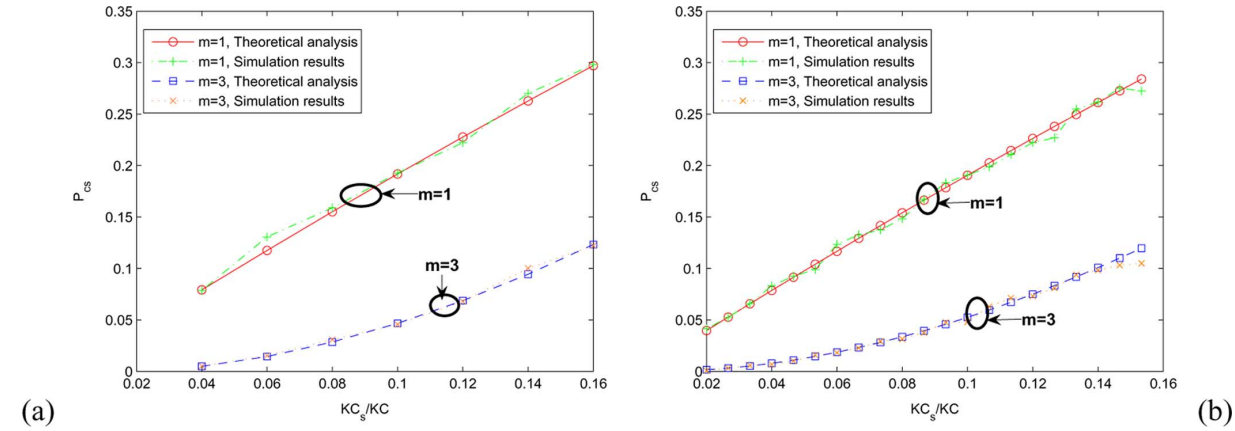

Fig. 8. Probability that a group of selfish colluders modify the values of $\{\bar{D}_j(k,l)\}$ and colluders in SC_h make errors when detecting selfish behavior. (a) The total number of colluders is $KC = 50$. (b) $KC = 150$. We let $KC_1 = KC_2 = KC/2$. In the x axis, KC_s is the number of selfish colluders in $\mathbf{C}_s$. The results are based on 4000 simulation runs.

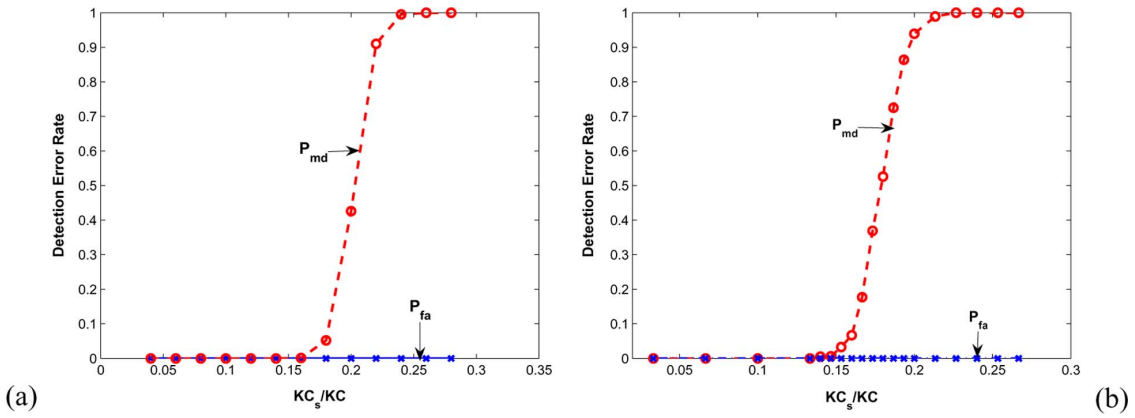

Fig. 9. Simulation results of P_{fa} and P_{md} on the first 300 frames of sequence carphone. (a) $KC = 50$ and (b) $KC = 150$. We let $KC_1 = KC_2 = KC/2$, and KC_s is the number of selfish colluders in $\mathbf{C}_s$. $m = 3$ assistants are selected from each subgroup to help calculate $\{\bar{D}_j(k,l)\}$, and colluders apply majority vote when identifying selfish colluders. $\alpha = 0.85$. The results are based on 4000 simulation runs.

with each other to prevent being detected by other fellow colluders. If a selfish colluder i in subgroup SC_1 is selected to help colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$, we assume that $u^{(i)}$ changes the histogram of $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$ such that none of the selfish colluders in SC_2 can be detected. In addition, $u^{(i)}$ randomly selects an honest colluder $k \in SC_2$, and change the values of $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$ so that Algorithm 1 falsely identifies $u^{(k)}$ as selfish. This is similar to that in Fig. 6(c). Same for selfish colluders in SC_2. The threshold α in (10) is set to 0.85.

Based on 4000 simulation runs, Fig. 9(a) and (b) shows the simulation results with $KC = 50$ and $KC = 150$ colluders, respectively. From Fig. 9, if less than 15% of the colluders are selfish, i.e., $KC_s/KC \leq 15\%$, the proposed autonomous selfish colluder identification algorithm can correctly identify all selfish colluders. When $KC_s/KC > 15\%$ and when P_{cs} is larger than $1-\alpha$, P_{md} increases quickly as the total number of selfish colluders grows. In addition, Figs. 8 and 9 show that the performance of our proposed algorithm depends on the percentage of selfish colluders KC_s/KC, but not the total number of colluders KC.

The false alarm probability (P_{fa}) depends on how selfish colluders change $\{\tilde{D}_j\}$. In our simulations, when the selfish colluders are selected to help calculate $\{\tilde{D}_j\}$, they randomly choose one honest colluder and accuse him/her as selfish. In all our 4000 simulation runs, as shown in Fig. 9, the proposed algorithm does not falsely accuse any honest colluders, even when there are a large number of selfish colluders who cooperate with each other to manipulate the detection results. This is because, majority vote and joint consideration of the detection results from all video frames help honest colluders easily correct this false alarm detection error. In another scenario where selfish colluders continuously compromise the same colluder in SC_h whenever possible, the false alarm rate P_{fa} will be similar to the miss detection rate P_{md}.

C. Resistance to Framing Attacks

In addition to actively manipulate the detection results, colluders can also *passively* attack the autonomous selfish colluder identification algorithm. The purpose of this passive attack is not to change the detection results but to access fingerprinted coefficients in others' copies and frame other colluders. This section analyzes the resistance of the autonomous selfish detection and identification scheme to such framing attacks. We use the term "framing colluders" to denote colluders who try to access fingerprinted coefficients in others' copies and frame other colluders. Note that framing colluders can be selfish colluders who process their copies before collusion, and it is also possible that framing

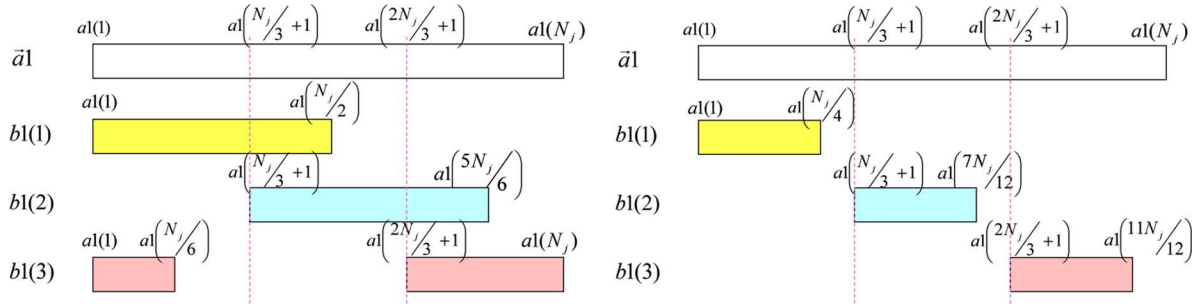

Fig. 10. Examples of $\{\mathbf{b}1(n)\}$ for each $i_{2,n} \in \mathrm{Ass}_j(SC_2)$, $n = 1, \ldots, m$. $m = 3$. (Left): $L = (1/2)N_j$. (Right): $L = (1/4)N_j$.

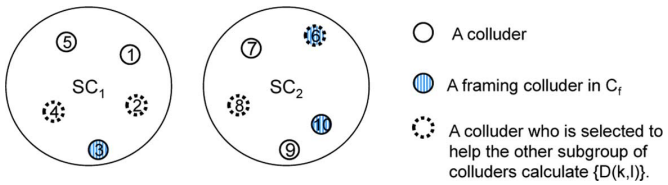

Fig. 11. Example to illustrate the terms defined in Section IV-C3a. In this example, there are ten colluders and $SC = \{1, 2, \ldots, 10\}$. $SC_1 = \{1, 2, 3, 4, 5\}$ and $SC_2 = \{6, 7, 8, 9, 10\}$. $\mathbf{C}_f = \{3, 6, 10\}$ includes all the framing colluders and $KC_f = 3$. Among these three framing colluders, colluder 3 is in SC_1 and colluders 6 and 10 are in SC_2. In this example, $KC_f(SC_1) = 1$ and $KC_f(SC_2) = 2$. In SC_1, colluders 2 and 4 are selected as the assistants and $\mathbf{A}_j(SC_1) = \{2, 4\}$. $\mathbf{A}_j(SC_2) = \{6, 8\}$, and colluders 6 and 8 are selected as assistances. $\mathbf{C}_f \cap \mathbf{A}_j(SC_1) = \emptyset$ and $KC_{af}(SC_1) = 0$. $\mathbf{C}_f \cap \mathbf{A}_j(SC_2) = \{6\}$ and $KC_{af}(SC_2) = 1$.

colluders honestly report their received fingerprinted copies but want to access fingerprinted coefficients in others' copies.

1) A Group of Framing Colluders: During the autonomous selfish colluder detection and identification in Section IV-A2, every colluder processes and encrypts his/her fingerprinted copy using two different keys. Any other single colluder has at most one key. Therefore, it prohibits a single framing colluder from accessing others' copies, and $\gamma_j = 0$ with one single framing colluder. However, it is possible that a group of framing colluders work collaboratively to access others' fingerprinted copies. For example, in Fig. 5, $u^{(l)}$ knows $K^{k,l}$ and $u^{(i)}$ has key $K^{k,i}$. If $u^{(l)}$ and $u^{(i)}$ collaborate, they can decrypt $\mathrm{Enc}\{f(\tilde{\mathbf{X}}_j^{(k)}, K^{k,l}), K^{k,i}\}$ and access the fingerprinted coefficients in $\tilde{\mathbf{X}}_j^{(k)}$. Let $\mathbf{C}_f$ denote the set containing the indices of framing colluders working together to access others' copies. In this paper, we consider the scenario where there are only a limited number of framing colluders and the size of $\mathbf{C}_f$ is small.

2) Non-Overlapping Content to Each Assistant: To lower γ_j and minimize the framing colluders' chance of successfully accessing others' copies, one possible solution is, for each selected assistant in SC_2, every colluder in SC_1 transmits only part of his/her fingerprinted frame, instead of the entire one. Thus, if only one of the framing colluders in $\mathbf{C}_f$ is selected to help calculate the MSE between different copies, they can only decrypt part of the fingerprinted copies; and decrypting the entire fingerprinted frames requires that multiple framing colluders are selected as assistants.

Assume that the jth fingerprinted frame from $u^{(i)}$ is $\tilde{\mathbf{X}}_j^{(i)} = [\tilde{X}_j^{(i)}(1), \tilde{X}_j^{(i)}(2), \ldots, \tilde{X}_j^{(i)}(N_j)]$. Same as in Section IV-B, for each frame j, the colluders first divide themselves into two non-overlapping subgroups SC_1 and SC_2. Then, m colluders in SC_2 are selected as assistants, and $\mathbf{A}_j(SC_2) = \{i_{2,1}, \ldots, i_{2,m}\} \subset SC_2$ is the set containing their indices. For colluders in SC_1:

- they first randomly shuffle the vector $[1, 2, \ldots, N_j]$, and $\mathbf{a}1 = [a1(1), a1(2), \ldots, a1(N_j)]$ is the returned shuffled vector. Here, $a1(l) \in \{1, 2, \ldots, N_j\}$ for $l = 1, \ldots, N_j$, and $a1(l_1) \neq a1(l_2)$ if $l_1 \neq l_2$.
- For each $i_{2,n} \in \mathbf{A}_j(SC_2)$, let $\mathbf{b}1(n) \triangleq \{a1(\mathrm{mod}(l, N_j)) : (n-1)/(m)N_j + 1 \leq l \leq (n-1)/(m)N_j + L\}$, where $1 \leq L \leq N_j$. Fig. 10 shows examples of $\{\mathbf{b}1(n)\}$ for each $i_{2,n} \in \mathbf{A}_j(SC_2)$ with $m = 3$. For $i_{2,p}, i_{2,q} \in \mathbf{A}_j(SC_2)$ where $p \neq q$, $\mathbf{b}1(p)$ and $\mathbf{b}1(q)$ are of the same length L.
- For each $i_{2,n} \in \mathbf{A}_j(SC_2)$, every colluder $u^{(k)}$ in SC_1 selects $\tilde{\mathbf{X}}_{j,n}^{(k)} \triangleq \{\tilde{X}_j^{(k)}(l) : l \in \mathbf{b}1(n)\}$, processes and encrypts $\tilde{\mathbf{X}}_{j,n}^{(k)}$ in the same way as in Section IV-B2, and then transmits it to $u^{(i_{2,n})}$.

Note that $L = N_j/m$ corresponds to a random partitioning. Colluders in SC_2 repeat the same process: generate a shuffled vector $\mathbf{a}2$, select $\mathbf{b}2(n)$ for each assistant $i_{1,n} \in \mathbf{A}_j(SC_1)$, and transmits the encrypted version of $\tilde{\mathbf{X}}_{j,n}^{(k \in SC_2)} \triangleq \{\tilde{X}_j^{(k)}(l) : l \in \mathbf{b}2(n)\}$ to $u^{(i_{1,n})}$. Finally, colluders follow the same procedure as in Section IV-B2 to detect and identify selfish colluders.

3) Performance Analysis: In this section, we first calculate γ_j defined in (5) for the autonomous selfish colluder detection and identification algorithm. We then quantify its robustness against framing attacks and evaluate the maximum number of framing colluders that the autonomous selfish colluder detection and identification algorithm can withstand.

a) Terminology Definition: Assume that there are KC_1 and KC_2 colluders in subgroup SC_1 and SC_2, respectively, and $KC_f = |\mathbf{C}_f|$ is the number of framing colluders. Among the KC_f framing colluders, $KC_f(SC_1) = |\mathbf{C}_f \cap SC_1|$ of them are in subgroup SC_1 and the other $KC_f(SC_2) = |\mathbf{C}_f \cap SC_2|$ are in SC_2. We have $KC_f(SC_1) + KC_f(SC_2) = KC_f$ and $0 \leq KC_f(SC_1), KC_f(SC_2) \leq KC_f$. We further define $KC_{af}(SC_1) \triangleq |\mathbf{C}_f \cap \mathbf{A}_j(SC_1)|$ as the number of framing colluders that are selected to help colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}_{k,l \in SC_2}$, and $KC_{af}(SC_2) \triangleq |\mathbf{C}_f \cap \mathbf{A}_j(SC_2)|$ is the number of framing colluders that are selected to help colluders in SC_1 calculate the MSE between different copies. Fig. 11 gives an example of the above defined terms.

b) Analysis of γ_j: In this paper, we consider the scenario where $KC_f \ll KC_1$ and $KC_f \ll KC_2$. We consider two

scenarios: $L = N_j$ where every colluder in SC_1 transmits his/her entire fingerprinted frame to all the selected assistants in $\mathbf{A}_j(SC_2)$, and $L < N_j$ where colluders in SC_1 only gives part of his/her copy to each assistant in $\mathbf{A}_j(SC_2)$.

i) $\mathbf{L = N_j}$: In this scenario, for each frame j in the video sequence, an assistant in $\mathbf{A}_j(SC_1)$ receives the entire fingerprinted frame from each colluder in SC_2. If both SC_1 and SC_2 contain framing colluders in $\mathbf{C}_f$ (i.e., $KC_f(SC_1) > 0$ and $KC_f(SC_2) > 0$), and if at least one framing colluder is selected as the assistant (i.e., $K_{af}(SC_1) > 0$ or $KC_{af}(SC_2) > 0$), then the framing colluders are able to obtain both keys and access others' fingerprinted coefficients. They can generate a new frame of high quality that does not contain any information of their own fingerprints. Recall that $\mathrm{Ind}_j^{(k,i)}$ is the set including all the indices of the fingerprinted coefficients in $\mathbf{X}_j^{(i)}$ that $u^{(k)}$ could access. Define $\mathrm{Ind}_j^{(k)} = \bigcup_{i\in SC, i\neq k} \mathrm{Ind}_j^{(k,i)}$, and $\mathrm{Ind}_j^{(\mathbf{C}_f)} = \bigcup_{k\in\mathbf{C}_f} \mathrm{Ind}_j^{(k)}$. Therefore, we have (13), shown at the bottom of the page. For $0 \le p \le KC_f$, let $G_f(p) \triangleq \{KC_f(SC_1) = p\}$ denote the event that p of the framing colluders are in SC_1. For $0 \le p \le KC_f$, $0 \le m_1 \le \min\{m, p\}$ and $0 \le m_2 \le \min\{m, KC_f - p\}$, we have

$$\begin{aligned}\gamma_j = &\sum_{p=1}^{KC_f-1} P\left[(KC_{af}(SC_1)\ge 1)\right.\\ &\left.\bigcup (KC_{af}(SC_2)\ge 1)\,|\,G_f(p)\right] \times P[G_f(p)]\\ = &\sum_{p=1}^{KC_s-1} \{1 - P[KC_{af}(SC_1) = 0\,|\,G_f(p)]\\ &\cdot P[KC_{af}(SC_2) = 0\,|\,G_f(p)]\}\\ &\times P[G_f(p)]\end{aligned}$$

where

$$\begin{aligned}&P[KC_{af}(SC_1) = m_1\,|\,G_f(p)]\\ &\quad = \binom{KC_1 - p}{m - m_1}\binom{p}{m_1}\Big/\binom{KC_1}{m}\\ &P[KC_{af}(SC_2) = m_2\,|\,G_f(p)]\\ &\quad = \binom{KC_2 - (KC_f - p)}{m - m_2}\\ &\qquad \times \binom{KC_f - p}{m_2}\Big/\binom{KC_2}{m},\end{aligned}$$

and

$$\begin{aligned}P[G_f(p)] = &\binom{KC_f}{p}\\ &\times \binom{KC - KC_f}{KC_1 - p}\Big/\binom{KC}{KC_1}.\end{aligned} \tag{14}$$

ii) $\mathbf{L < N_j}$: We use $L \le N_j/m$ as an example to analyze the performance of the selfish colluder detection and identification algorithm. The analysis for $N_j/m < L < N_j$ is similar and omitted here. From Fig. 10(b), if $L \le N_j/m$, $\mathbf{b}1(p) \bigcap \mathbf{b}1(q) = \emptyset$ for any $i_{2,p}, i_{2,q} \in \mathbf{A}_j(SC_2)$ where $p \neq q$. Similarly, $\mathbf{b}2(p) \bigcap \mathbf{b}2(q) = \emptyset$ for any $i_{1,p}, i_{1,q} \in \mathbf{A}_j(SC_1)$ where $p \neq q$.

For each frame j, among all the K_f framing colluders in $\mathbf{C}_f$, assume that $KC_{af}(SC_1) = m_1 \le m$ of them are selected to help colluders in SC_2 calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$, and $KC_{af}(SC_2) = m_2 \le m$ of the framing colluders in $\mathbf{C}_f$ are selected to help colluders in SC_1 calculate $\tilde{D}_j(k,l)\}_{k,l\in SC_1}$. By combining all the decrypted fingerprinted coefficients that they have, we can show that the following holds:

$$\begin{aligned}E\Big[\Big|\mathrm{Ind}_j^{(\mathbf{C}_f)}\Big|\,\Big|\,&KC_{af}(SC_1) = m_1,\\ &KC_{af}(SC_2) = m_2, 0 < K_f(SC_1),\\ &KC_f(SC_2) < KC_f\Big]\\ &= \min\left\{m_1 L + m_2 L - m_1 m_2 \frac{L^2}{N_j}, N_j\right\}.\end{aligned} \tag{15}$$

Therefore, we have (16), shown at the bottom of the page, where $P[KC_{af}(SC_1) = m_1|G_f(p)]$, $P[KC_{af}(SC_2) = m_2|G_f(p)]$ and $P[G_f(p)]$ are the same as in (14).

Fig. 12 shows the simulation results of γ_j when L takes different values. There are a total of $KC = 150$ colluders, and $KC_1 = KC_2 = KC/2$ with $m = 3$. From Fig. 12, transmitting only part of the fingerprinted frames to each selected assistant can significantly reduce γ_j and help improve the robustness against framing attacks. For example, with $KC_f/KC = 0.1$, γ_j equals to 50% when $L = N_j$ and is reduced to 15% if

$$\left|\mathrm{Ind}_j^{(\mathbf{C}_f)}\right| = \begin{cases} N_j, & \text{if } (0 < KC_f(SC_1), KC_f(SC_2) < K_f) \bigcap (\{KC_{af}(SC_1)\ge 1\} \bigcup \{KC_{af}(SC_2)\ge 1\})\\ 0, & \text{otherwise.}\end{cases} \tag{13}$$

$$\begin{aligned}\gamma_j = &\sum_{p=1}^{KC_f-1}\sum_{m_1=0}^{\min\{m,p\}}\sum_{m_2=0}^{\min\{m,KC_f-p\}} E\left[\frac{\left|\mathrm{Ind}_j^{(\mathbf{C}_f)}\right|}{N_j}\,\middle|\,KC_{af}(SC_1) = m_1, KC_{af}(SC_2) = m_2, G_f(p)\right]\\ &\times P[KC_{af}(SC_1) = m_1, KC_{af}(SC_2) = m_2, G_f(p)]\\ = &\sum_{p=1}^{KC_f-1}\sum_{m_1=0}^{\min\{m,p\}}\sum_{m_2=0}^{\min\{m,KC_f-p\}} \min\left\{m_1\frac{L}{N_j} + m_2\frac{L}{N_j} - m_1 m_2\left(\frac{L}{N_j}\right)^2, 1\right\}\\ &\times P[KC_{af}(SC_1) = m_1|G_f(p)] \times P[KC_{af}(SC_2) = m_2|G_f(p)] \times P[G_f(p)]\end{aligned} \tag{16}$$

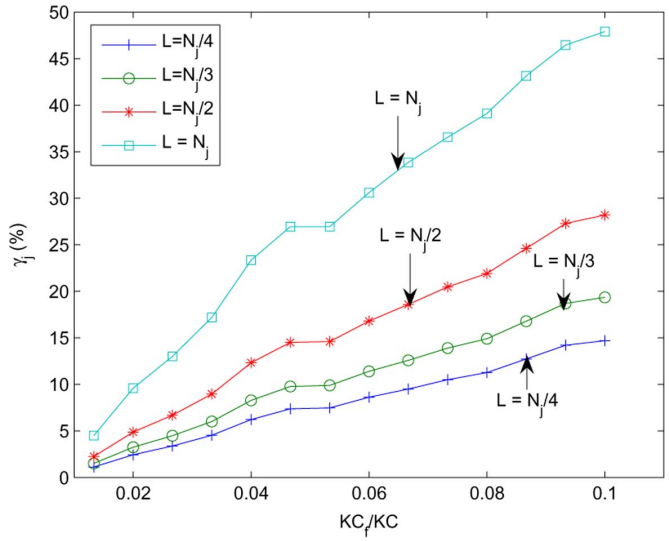

Fig. 12. γ_j when L takes different values. There are a total of $KC = 150$ colluders, $KC_1 = KC_2 = KC/2$, and $m = 3$. Simulation results are based on 4000 simulation runs.

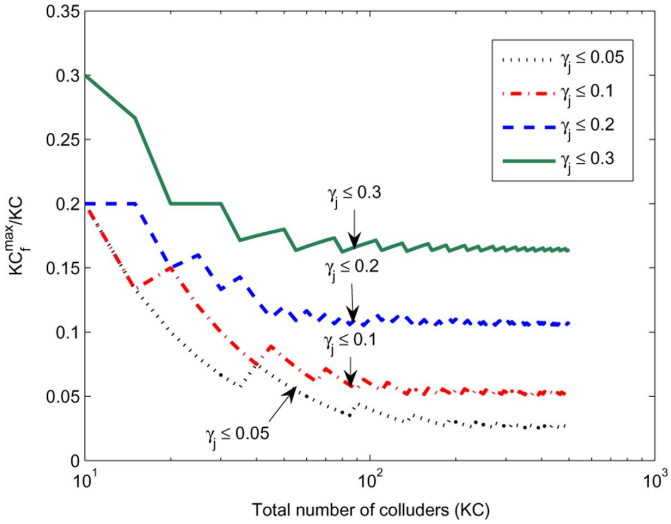

Fig. 13. Maximum number of framing colluders in $\mathbf{C}_f$ that the autonomous selfish colluder detection and identification process can resist. We let $KC_1 = KC_2 = KC/2$, $m = 3$ and $L = N_j/3$.

$L = N_j/4$. In addition, γ_j has a smaller value when there are fewer framing colluders in $\mathbf{C}_f$.

From Fig. 12, γ_j has a smaller value when L decreases and, therefore, a smaller L is preferred to minimize γ_j and resist framing attacks. On the other hand, for each assistant $i_{2,n} \in \mathbf{A}_j(SC_2)$, $\{\tilde{\mathbf{X}}_{j,n}^{(k)}\}_{k\in SC_1}$ has to be long enough such that given $\{\tilde{d}_j^{(i_{2,n})}(k,l) \triangleq \|f(\tilde{\mathbf{X}}_{j,n}^{(k)}, K^{SC_1}) - f(\tilde{\mathbf{X}}_{j,n}^{(l)}, K^{SC_1})\|^2\}_{k,l\in SC_1}$ that are received from $u^{(i_{2,n})}$, Algorithm 1 can correctly detect and identify selfish colluders in SC_1. We use the second frame in the carphone sequence an example, and assume that 10 out 150 colluders are selfish who process their fingerprinted copies before collusion. They select the parameter λ_j to generate new frames with PSNR 45 dB. When $L = N_j$, $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ do not overlap, and Algorithm 1 can accurately identify all ten selfish colluders. With $L = N_j/4$ and $L = N_j/8$, the overlap ratio c defined in (6) are 14.54% and 34.24%, respectively, and Algorithm 1 starts to make detection errors. Thus, a larger L should be used to ensure the accuracy of Algorithm 1. To address this tradeoff, for the example in Fig. 12, $L \approx N_j/m$ with $m \approx 3$ is often preferred, that is, $L = N_j/3$ or $L = N_j/4$.

c) Resistance to Framing Attacks: In this section, we quantify the robustness of the selfish colluder identification algorithms against framing attacks. For any fingerprinted copy, given the requirement that framing colluders can access no more than θ percent of the fingerprinted coefficients, i.e., $\gamma \le \theta$, we define $KC_f^{\max} \triangleq \arg\max_{KC_f}\{\gamma_j \le \theta\}$, which is the maximum number of framing colluders that it can resist.

Fig. 13 plots the ratio $KC_f^{\max}/KC$ versus KC when θ takes different values. In Fig. 13, the two subgroups SC_1 and SC_2 are of the same size $KC/2$, and there are $m = 3$ assistants selected in each subgroup. We let $L = N_j/3$. From Fig. 13, with hundreds of colluders, if no more than 5% of them are framing colluders, then others can be sure that the framing colluders can access no more than 10% of the fingerprinted coefficients in their copies. If KC_f does not exceed 10% of the total number of colluders, then the framing colluders can access less than 20% of the fingerprinted coefficients in others' copies.

D. Autonomous Selfish Colluder Detection and Identification Process

To summarize, in peer-structured colluder social networks, the key steps in the autonomous selfish colluder detection and identification process are: for each frame j in the video sequence:

Step 1) **Grouping:** Colluders randomly divide themselves into two subgroups SC_1 and SC_2 with $SC_1 \cup SC_2 = SC$ and $SC_1 \cap SC_2 = \emptyset$. A total of m colluders in SC_1, $\mathbf{A}_j(SC_1) = \{i_{1,1}, \ldots, i_{1,m}\}$, are randomly selected to calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_2}$ for colluders in SC_2. Similarly, colluders in SC_2 randomly select m assistants $\mathbf{A}_j(SC_2) = \{i_{2,1}, \ldots, i_{2,m}\}$ to help colluders in SC_1 calculate $\{\tilde{D}_j(k,l)\}_{k,l\in SC_1}$.

Step 2) **Encryption:** Assume that K^{SC_1} is a key that is shared by colluders in SC_1. For each selected assistant $i_{2,n} \in \mathbf{A}_j(SC_2)$, every colluder $u^{(k)}$ in SC_1 generates a secret key $K^{k,i_{2,n}}$ shared with $u^{(i_{2,n})}$. For each $i_{2,n} \in \mathbf{A}_j(SC_2)$, colluders in SC_1 follow the same procedure as in Section IV.C.2 to generate $\mathbf{b1}(n)$. Then, every colluder $u^{(k)}$ in SC_1 selects $\tilde{\mathbf{X}}_{j,n}^{(k)} = \{\tilde{X}_j^{(k)}(l) : l \in \mathbf{b1}(n)\}$, processes and encrypts it with Key K^{SC_1} and $K^{k,i_{2,n}}$, respectively, in the same way as in Section IV.A.2. Finally, $u^{(k)}$ transmits the encrypted $\tilde{\mathbf{X}}_{j,n}^{(k)}$ to $u^{(i_{2,n})}$. Colluders in SC_2 follow the same procedure, process and encrypt their fingerprinted copies, and transmit them to the corresponding assistants in $\mathbf{A}_j(SC_1)$.

Step 3) **Calculation of** $\{D_j\}$: After decrypting the bit streams received from all colluders in SC_1, each selected assistant $i_{2,n} \in \mathbf{A}_j(SC_2)$ follows the same procedure in Section IV-A1 to calculate $\{\tilde{d}_j^{i_{2,n}}(k,l) = \|f(\tilde{\mathbf{X}}_{j,n}^{(k)}, K^{SC_1}) -$

$f(\tilde{\mathbf{X}}_{j,n}^{(l)}, K^{SC_1})\|^2\}_{k,l\in SC_1}$, and broadcasts the results to colluders in SC_1 together with his/her digital signature. Each selected assistant $i_{1,n}$ in $\mathbf{A}_j(SC_1)$ repeats the same process to help colluders in SC_2 calculate $\{\tilde{d}_j^{i_{1,n}}(k,l)\}_{k,l\in SC_2}$.

Step 4) **Selfish Colluder Detection and Identification:** For every honest colluder $u^{(k)}$ in SC_1, given $\{\tilde{d}_j^{i_{2,n}}(k,l)\}_{k,l\in SC_1}$ received from the selected assistant $i_{2,n} \in \mathbf{A}_j(SC_2)$, $u^{(k)}$ follows Step 4 in Section IV.A.2 and sets $\upsilon_j^{(k)}(n,l) = 1$ if Algorithm 1 outputs $u^{(l\in SC_1)}$ as a selfish colluder. Otherwise, $\upsilon_j^{(k)}(n,l) = 0$. Then for every colluder $l \neq k$ in SC_1, $u^{(k)}$ combines the m detection results $\{\upsilon_j^{(k)}(n,l)\}_{n=1,\ldots,m}$, considers $u^{(l)}$ as a potential selfish colluder and sets $\Upsilon_j^{(k)}(l) = 1$ if $\sum_{n=1}^{m} \upsilon_j^{(k)}(n,l) > \lceil m/2 \rceil$. Otherwise, $\Upsilon_j^{(k)}(l) = 0$.

Finally, each colluder $u^{(k\in SC_h)}$ combines the detection results from all frames in the video sequence and outputs the estimated selfish colluder set $\widehat{SC}_s^{(k)} = \{l : (\sum_{j\in F(k,l)} \Upsilon_j^{(k)}(l))/(|F(k,l)|) > \alpha\}$ where α is a predetermined threshold. Then, colluders in SC_h exclude those identified selfish colluders from collusion.

From the analysis in Section IV-B3, the above autonomous selfish colluder detection and identification process can accurately identify selfish colluders without false accusing others when there are limited number of selfish colluders. From Section IV-C3, the above algorithm also helps resist framing attacks and prevent colluders from accessing the fingerprinted coefficients in others' copies when the number of framing colluders is small.

V. Conclusion

This paper studies human dynamics in video sharing social networks and provides a case study of misbehavior detection and forensics in multiuser collusion attacks against video fingerprinting. When there exist selfish colluders who process their fingerprinted copies before collusion to further lower their own risk, this paper investigates possible strategies to detect and identify these selfish colluders, and analyzes the impact of network structures on misbehavior detection and identification. We also evaluate the performance of the proposed algorithms and analyze their resistance against framing attacks.

We first consider the centralized colluder social networks where there exists a ringleader whom all colluders can trust, and we propose an algorithm where the trusted ringleader helps detect and identify selfish colluders. The trusted ringleader calculates the difference between fingerprinted copies from different colluders, and the colluders analyze the histogram of this difference to detect precollusion processing and identify selfish colluders. We show that the proposed scheme can accurately identify selfish colluders without falsely accusing others even if $\mathfrak{D}_j(SC_h, SC_s)$ and $\mathfrak{D}_j(SC_h, SC_h)$ overlap. The proposed algorithm also protects the fingerprinted coefficients in all copies and prevents colluders from framing each other.

We then consider the peer structure where there does not exist such a trusted ringleader, and we propose an autonomous algorithm where colluders help each other detect precollusion processing and identify selfish colluders. In this scenario, all the fingerprinted copies have to be processed and encrypted appropriately during selfish colluder detection to prevent framing attacks. From our analytical and simulation results, when detecting selfish behavior, the proposed algorithm can accurately identify selfish colluders even if a small group of selfish colluders collaborate with each other to change the detection results. We also evaluate its antiframing performance, and quantify the maximum number of framing colluders that it can resist. Our results show that framing colluders can access no more than 10% of the fingerprinted coefficients in others' copies, if the number of framing colluders does not exceed 5% of the total number of colluders.

References

[1] X. Hei, Y. Liu, and K. W. Ross, "IPTV over P2P streaming networks: The mesh-pull approach," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 86–92, Feb. 2008.

[2] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Commun. Surveys Tutorial*, vol. 7, no. 2, pp. 72–93, Mar. 2004.

[3] X. Cheng, C. Dale, and J. Liu, "Statistics and social networking of YouTube videos," in *Proc. IEEE Int. Workshp Quality of Service (IWQoS)*, Jun. 2008, pp. 229–238.

[4] C. Buragohain, D. Agrawal, and S. Sur, "A game theoretic framework for incentives in P2P systems," in *Proc. 3rd Intl. Conf. Peer-to-Peer Comput.*, Sep. 2003, pp. 48–56.

[5] G. Tan and S. A. Jarvis, "A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast," in *Proc. Int. Workshop Quality of Service (IWQoS)*, Jun. 2006.

[6] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free riding in BitTorrent is cheap," in *Proc. 5th Workshop Hot Topics in Netw. (HotNets)*, Irvine, CA, Nov. 2006.

[7] N. E. Baughman, M. Liberatore, and B. N. Levine, "Cheat-proof playout for centralized and peer-to-peer gaming," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 1–13, Feb. 2007.

[8] G. Horng, T. Chen, and D. Tsai, "Cheating in visual cryptography," *Designs, Codes, Cryptography*, vol. 38, no. 2, pp. 219–236, Feb. 2006.

[9] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *SPIE J. Electron. Imag.*, vol. 9, no. 4, pp. 456–467, Oct. 2000.

[10] W. Trappe, M. Wu, Z. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process., Special Iss. Signal Process. for Data Hiding in Digital Media*, vol. 51, no. 4, pp. 1069–1087, Apr. 2003.

[11] Z. J. Wang, M. Wu, W. Trappe, and K. J. R. Liu, "Group-oriented fingerprinting for multimedia forensics," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 14, pp. 2142–2162, Nov. 2004.

[12] S. He and M. Wu, "Joint coding and embedding techniques for multimedia fingerprinting," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 2, pp. 231–247, Jun. 2006.

[13] H. Stone, "Analysis of attacks on image watermarks with randomized coefficients," NEC Research Inst., Tech. Rep. 96-045, 1996.

[14] F. Hartung, J. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counterattacks," in *Proc. SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging*, Jan. 1999, pp. 147–158.

[15] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in *proc. 2nd Workshop Inf. Hiding, Lecture Notes in Compute. Sci.*, Apr. 1998, pp. 218–238.

[16] H. V. Zhao and K. J. R. Liu, "Traitor-within-traitor behavior forensics: Strategy and risk minimization," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 4, pp. 440–456, Dec. 2006.

[17] W. S. Lin, H. V. Zhao, and K. J. R. Liu, "Scalable multimedia fingerprinting forensics with side information," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 2293–2296.

[18] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 525–540, May 1998.
[19] Z. J. Wang, M. Wu, H. V. Zhao, W. Trappe, and K. J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 804–821, Jun. 2005.
[20] G. Doerr, J. L. Dugelay, and L. Grange, "Exploiting self-similarities to defeat digital watermarking systems: A case study on still images," in *Proc. ACM Multimedia and Security Workshop*, 2004.
[21] D. Kirovski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1045–1053, Apr. 2003.
[22] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 43–51, Feb. 2005.
[23] S. Baker, R. Gross, I. Matthews, and T. Ishikawa, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb.–Mar. 2004.
[24] S. Yoon and N. Ahuja, "Frame interpolation using transmitted block-based motion vectors," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2001, vol. 3, pp. 856–859.
[25] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC, 1996.
[26] A. Tosun and W. Feng, "On error preserving encryption algorithms for wireless video transmission," in *Proc. ACM Multimedia Conf.*, 2001, vol. 9, pp. 302–308.
[27] H. V. Zhao, W. S. Lin, and K. J. R. Liu, *Behavior Dynamics in Media-Sharing Social Networks*. Cambridge, MA: Cambridge, Univ. Press, 2010, to be published.

H. Vicky Zhao (M'05) received the B.S. and M.S. degree from Tsinghua University, Beijing, China, in 1997 and 1999, respectively, and the Ph.D. degree from University of Maryland, College Park, in 2004, all in electrical engineering.

She was a Research Associate with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, from January 2005 to July 2006. Since August 2006, she has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Her research interests include information security and forensics, multimedia social networks, digital communications, and signal processing. She coauthored the book *Multimedia Fingerprinting Forensics for Traitor Tracing* (Hindawi, 2005).

Dr. Zhao received the IEEE Signal Processing Society (SPS) 2008 Young Author Best Paper Award. She is an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS and *Elsevier Journal of Visual Communication and Image Representation*.

K. J. Ray Liu (F'03) was named a Distinguished Scholar-Teacher of University of Maryland, College Park, in 2007. He is Associate Chair of Graduate Studies and Research of Electrical and Computer Engineering Department and leads the Maryland Signals and Information Group conducting research encompassing broad aspects of wireless communications and networking, information forensics and security, multimedia signal processing, and biomedical engineering. His recent books include *Cognitive Radio Networking and Security: A Game Theoretical View* (Cambridge University Press, 2010), *Cooperative Communications and Networking* (Cambridge University Press, 2008), *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications* (Cambridge University Press, 2008), *Ultra-Wideband Communication Systems: The Multiband OFDM Approach* (IEEE-Wiley, 2007), *Network-Aware Security for Group Communications* (Springer, 2007), *Multimedia Fingerprinting Forensics for Traitor Tracing* (Hindawi, 2005), and *Handbook on Array Processing and Sensor Networks* (IEEE-Wiley, 2009).

Dr. Liu is the recipient of numerous honors and awards including the IEEE Signal Processing Society Technical Achievement Award and Distinguished Lecturer. He also received various teaching and research recognitions from the University of Maryland including a university-level Invention of the Year Award and the Poole and Kent Senior Faculty Teaching Award and Outstanding Faculty Research Award, both from the A. James Clark School of Engineering. He is a Fellow of the AAS. He is President-Elect and was Vice President—Publications of the IEEE Signal Processing Society. He was the Editor-in-Chief of the *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of *EURASIP Journal on Advances in Signal Processing*.