AUTONOMOUS IDENTIFICATION OF SELFISH COLLUDERS IN TRAITOR-WITHIN-TRAITOR BEHAVIOR FORENSICS

H. Vicky Zhao

ECE Dept., University of Alberta
Edmonton, AB T6G 2V4 Canada

K. J. Ray Liu

ECE Dept., University of Maryland
College Park, MD 20742 USA

ABSTRACT

During multi-user collusion attacks against digital fingerprinting, an important issue that colluders have to address is to distribute the risk evenly among all colluders and achieve fairness of the attack. Although they might agree so, some selfish colluders may break their agreement and process their fingerprinted copies before collusion in order to further reduce their own risk. To protect their own interest, other colluders have to detect these selfish colluders and exclude them from multi-user collusion. This paper studies this problem of traitors within traitors. We propose an autonomous selfish colluder detection and identification algorithm, in which colluders help each other detect selfish behavior. We show that the proposed algorithm can correctly identify all selfish colluders without falsely accusing any others, even when a small group of selfish colluders collaborate with each other to change the detection results.

Index Terms— security, multimedia systems, video signal processing

1. INTRODUCTION

The popularity of multimedia applications in government operations and commercial markets has raised the critical issue of protecting multimedia content and enforcing digital rights. Multimedia security systems involve a lot of users with conflicting objectives and they influence each other's decisions. An important issue in media security is to formulate the dynamics among users and investigate how they interact with and respond with each other. Such analysis helps the digital rights enforcer have a better understanding of multimedia security and offer stronger protection of multimedia.

During multi-user collusion, a powerful attack against digital fingerprinting, a group of attackers collectively undermine the traitor tracing capability of multimedia fingerprints. One important issue during collusion is to distribute the risk evenly among colluders and achieve fairness of the attack. To ensure equal risk of all colluders, colluders must give each other correct information about their fingerprinted copies, and adjust the collusion attacks accordingly.

Most prior work on multi-user collusion assumed that colluders keep their fair-play agreement and focused on the analysis of collusion strategies and effectiveness [1–3]. However, there might exist some selfish colluders who wish to further lower their probability of being detected. They may process their fingerprinted copies before collusion to minimize their risk [4]. In some scenarios, pre-collusion processing may even increase other attackers' probability of being detected, and thus is not only selfish but also malicious. To protect their own interest, other attackers must detect selfish colluders and exclude them from collusion. The existence of selfish colluders introduces complex dynamics among colluders, and it is important to study this problem of *traitors within traitors*.

The authors can be reached at vzhao@ece.ualberta.ca and kjrliu@eng.umd.edu.

The work in [5] investigated selfish colluder detection and identification in traitors within traitors, assuming that there is a ringleader whom all colluders trust. This paper focuses on autonomous selfish colluder identification, where there does not exist a trusted ringleader and colluders help each other detect selfish behavior.

The rest of the paper is organized as follows. We introduce the problem of traitors within traitors and formulate the dynamics among colluders in Section 2. Section 3 reviews selfish colluder identification with a trusted ringleader [5]. Section 4 proposes an autonomous selfish colluder detection algorithm, and we show simulation results in Section 5. Conclusions are drawn in Section 6.

2. SYSTEM MODEL AND PROBLEM FORMULATION

2.1. Multimedia Fingerprinting

Proven to be robust against many single-copy attacks as well as common signal processing, spread spectrum embedding [6, 7] is widely used in multimedia fingerprinting. It additively embeds fingerprints into the host signal, and uses human visual models [7] to achieve the imperceptibility of the embedded fingerprints.

During multi-user collusion, a group of attackers who receive differently fingerprinted copies of the same content collectively mount attacks, and generate a new copy where 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 lowers all colluders' risk of being detected [1, 6].

When identifying colluders, the fingerprint detector first removes the host signal from the test copy and extracts the fingerprint. Then, the detector measures the similarity between the extracted fingerprint and each of the original fingerprints, compares with a pre-determined threshold and outputs the estimated identities of the colluders.

2.2. Traitor-within-Traitor Behavior Dynamics

During collusion, colluders not only share the profit from the illegal usage of multimedia, they also share the risk of being detected. An important issue that colluders have to address during collusion is to balance the profit and the risk that each attacker takes and ensure fairness of the attack. Absolute fairness is widely adopted in the literature, where all colluders have the same probability of being detected. To ensure fairness of collusion, colluders are required to provide one another correct information of their fingerprinted copies, and then adjust the collusion attacks accordingly.

Most prior work assumed that all colluders keep their fair collusion agreement. However, the assumption of fair play may not always hold. There might exist some selfish colluders who wish to take no risk of being detected while still profiting from collusion. It was shown in [4] that temporal filtering of their fingerprinted copies before collusion can help selfish colluders further reduce their own risk. In some scenarios, such selfish pre-collusion processing may also increase other attackers' probability of being detected by the digital rights enforcer, and it is not only selfish but also malicious.

The existence of selfish colluders introduces complicated dynamics among attackers during collusion. No colluders know what others might have done to their fingerprinted copies and how it might affect their own risk. They do not trust each other, and this distrust forbids them to collude with each other. To continue collusion, colluders have to build trust among themselves first, force everyone to keep their fair collusion agreement, and guarantee that all colluders share the profit and the risk as agreed. This requires each colluder to examine the fingerprinted copies from others before collusion, identify selfish colluders, and exclude them from collusion. This paper addresses this issue of selfish colluder detection and identification in *traitors within traitors*.

2.3. Problem Formulation

With the existence of selfish colluders, to continue collusion, colluders have to share something in common that enables them to detect selfish behavior and build trust among themselves first. The work in [5] considered the scenario where there is a ringleader whom all colluders trust and investigated how the trust ringleader can help detect and identify selfish colluders. This paper focuses on autonomous selfish colluder identification, where there does not exist a trusted ringleader and colluders help each other detect selfish behavior. We consider the scenario where there are only a few selfish colluders and most attackers keep their fair collusion agreement. In this paper, we explore strategies for attackers who keep their fair-play agreement to collaborate with each other and identify selfish colluders.

Note that before a colluder decides with whom to collude, he/she is unwilling to give others his/her received copy that contains his/her identification information. Thus, selfish colluder detection and identification must prevent attackers from accessing fingerprinted coefficients in others' copies.

2.4. Performance Criteria

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

To evaluate the performance of the proposed algorithm, we consider two types of detection errors: the probability that there exists at least one colluder in SC_h who misses a selfish colluder in SC_s during detection (P_{md}); and the probability that at least one colluder in SC_h falsely accuses another one in SC_h as selfish (P_{fa}).

3. SELFISH COLLUDER IDENTIFICATION WITH A TRUSTED RINGLEADER

For selfish colluders to further reduce their probability of being detected, one possible solution is to attenuate the energy of the embedded fingerprints even before multi-user collusion, e.g., by temporally filtering adjacent frames of similar content in a video sequence [4].

Assume that colluder $\mathbf{u}^{(i)}$ tells other attackers that $\widetilde{\mathbf{X}}_j^{(i)}$ is the fingerprinted frame j that he/she received from the content owner. For two colluders $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$, define $D_j(k,l) = ||\widetilde{\mathbf{X}}_j^{(k)} - \widetilde{\mathbf{X}}_j^{(l)}||^2$. We further define $\mathfrak{D}_j(SC_h, SC_h) = \{D_j(k,l) : k,l \in SC_h, k \neq l\}$ and $\mathfrak{D}_j(SC_h, SC_s) = \{D_j(k,l) : k \in SC_h, l \in SC_s\}$. It was shown in [5] that, temporal filtering not only averages the embedded fingerprints and attenuates their energies, it also filters adjacent host frames and introduces extra distortion into the host signal. Consequently, if there are no selfish colluders, then $\{D_j(k,l)\}_{k,l \in SC}$ follow the same distribution with a single mean. When there are some colluders who temporally filter their copies before collusion, $\{D_j(k,l)\}$ are from two or more distributions with distinct means, and $\mathfrak{D}_j(SC_h, SC_s)$ has a much larger mean than $\mathfrak{D}_j(SC_h, SC_h)$.

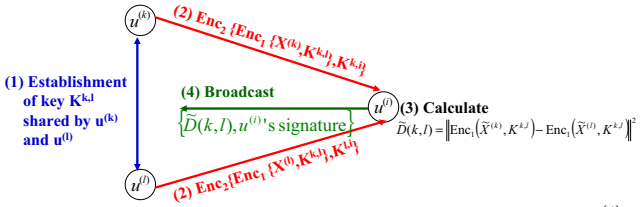

Fig. 1. Calculation of $D(k,l)$ without a trusted ringleader. $\mathbf{u}^{(i)}$ is selected to help $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ calculate $D(k,l)$.

Based on this observation, the work in [5] proposed an algorithm to detect and identify selfish colluders. First, $D_j(k,l)$ is calculated for every pair of colluders $(\mathbf{u}^{(k)}, \mathbf{u}^{(l)})$. Then, the histogram of $\{D_j(k,l)\}$ is examined to determine the existence of pre-collusion processing. If all $D_j(k,l)$ are from the same distribution with a single mean, there are no selfish colluders. Otherwise, there is at least one selfish colluder. Finally, colluders in SC_h examine every $D_j(k,l)$ in $\mathfrak{D}_j(SC_h, SC_s)$, separate SC into two subgroups and identify selfish colluders. Readers are referred to [4] for details of the pre-collusion processing detection and selfish colluder identification algorithm. From [5], the above algorithm can accurately identify all selfish colluders in SC_s without falsely accusing any others in SC_h, and $P_{fa} = 0$ and $P_{md} = 0$.

To prevent colluders from framing each others, it is required that no colluder can access fingerprinted coefficients in others' copies during this selfish colluder detection process, especially when calculating $\{D_j(k,l)\}$. To achieve this goal, the work in [5] considered the scenario where there exists a trusted ringleader. Colluders believe that the trusted ringleader will not give their fingerprinted copies to any other attackers; the ringleader himself/herself will not frame any colluders; and the ringleader will give them the exact output of the selfish colluder detection and identification algorithm and will not modify the results. In [5], the trusted ringleader helps colluders calculate $\{D_j(k,l)\}$: first, each colluder encrypts his/her fingerprinted frame with a secret key shared with the ringleader only, and transmits the encrypted bit stream to the ringleader. This encryption prevents others from accessing the fingerprinted coefficients without the decryption key. Then, the trusted ringleader decrypts the received bit stream, and calculates $D_j(k,l)$ for every pair $(\mathbf{u}^{(k)}, \mathbf{u}^{(l)})$.

4. AUTONOMOUS SELFISH COLLUDER DETECTION

4.1. Calculation of $D_j(k,l)$

Without a trusted ringleader, the challenging issue in autonomous selfish colluder identification is how colluders calculate $\{D_j(k,l)\}$ without knowing the fingerprinted coefficients in others' copies. Assume that $\widetilde{\mathbf{X}}_j^{(k)}$ is the copy that colluder $\mathbf{u}^{(k)}$ uses during collusion. Without a trusted ringleader, for each pair of colluders $(\mathbf{u}^{(k)}, \mathbf{u}^{(l)})$, they have to find a third colluder $\mathbf{u}^{(i)}$ to help them calculate $D_j(k,l)$. To prevent $\mathbf{u}^{(i)}$ from accessing the fingerprinted coefficients in their copies, $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ must encrypt $\widetilde{\mathbf{X}}_j^{(k)}$ and $\widetilde{\mathbf{X}}_j^{(l)}$ with a key $K^{k,l}$ that is known to $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ only, and let $\mathbf{u}^{(i)}$ calculate $D_j(k,l)$ from the encrypted $\widetilde{\mathbf{X}}_j^{(k)}$ and $\widetilde{\mathbf{X}}_j^{(l)}$. In addition, to prevent $\mathbf{u}^{(l)}$ from accessing the fingerprinted coefficients in $\widetilde{\mathbf{X}}_j^{(k)}$, $\mathbf{u}^{(k)}$ should also encrypt $\widetilde{\mathbf{X}}_j^{(k)}$ with a key $K^{k,i}$ that is shared by $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(i)}$ only. By doing so, accessing fingerprinted coefficients in $\widetilde{\mathbf{X}}_j^{(k)}$ requires the knowledge of both key $K^{k,l}$ and $K^{k,i}$; while any other single colluder has at most one decryption key and thus, cannot frame $\mathbf{u}^{(k)}$.

Let $\mathrm{Enc}(X,K)$ denote the encryption of message X with key K. To calculate $D_j(k,l)$, as shown in Figure 1,

- $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ first generate a secret key $K^{k,l}$. $K^{k,i}$ is a secret key shared by $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(i)}$.
- $\mathbf{u}^{(k)}$ first encrypts $\widetilde{\mathbf{X}}_j^{(k)}$ with key $K^{k,l}$, then encrypts it again with key $K^{k,i}$. Then, $\mathbf{u}^{(k)}$ transmits the encrypted copy, $\text{Enc}_2\left(\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{k,l}), K^{k,i}\right)$, to $\mathbf{u}^{(i)}$. $\mathbf{u}^{(l)}$ repeats the same process.
- $\mathbf{u}^{(i)}$ calculates and broadcasts $\widetilde{D}_j(k,l) \triangleq ||\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{k,l}) - \text{Enc}_1(\widetilde{\mathbf{X}}_j^{(l)}, K^{k,l})||^2$, together with his/her digital signature to enable other colluders to authenticate the sender and verify the integrity of the transmitted data.

In Figure 1, the two encryptions have different requirements and, therefore, should use different methods. The second encryption $\text{Enc}_2(X, K)$ aims to prevent $\mathbf{u}^{(l)}$ from accessing the fingerprinted coefficients in $\widetilde{\mathbf{X}}_j^{(k)}$, and $\mathbf{u}^{(k)}$ can use any methods in the literature [8] that provide the desired security. The first encryption $\text{Enc}_1(X, K)$ must enable $\mathbf{u}^{(i)}$ to calculate $D_j(k,l)$ from the encrypted copies of $\widetilde{\mathbf{X}}_j^{(k)}$ and $\widetilde{\mathbf{X}}_j^{(l)}$. Thus, $\text{Enc}_1(X, K)$ has to preserve the MSE between these two copies, i.e.,

$$\begin{aligned} \widetilde{D}_j(k,l) &= ||\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{k,l}) - \text{Enc}_1(\widetilde{\mathbf{X}}_j^{(l)}, K^{k,l})||^2 \\ &= ||\widetilde{\mathbf{X}}_j^{(k)} - \widetilde{\mathbf{X}}_j^{(l)}||^2 = D_j(k,l). \end{aligned} \quad (1)$$

In this paper, for $\text{Enc}_1(X, K)$, we use a simple component-wise addition-based encryption method. Other methods that protect the fingerprinted coefficients and satisfy (1) can also be applied. Assume that $\widetilde{\mathbf{X}}_j^{(k)}$ and $\widetilde{\mathbf{X}}_j^{(l)}$ are of length N_j. $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ use key $K^{k,l}$ 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 i.i.d. and uniformly distributed in $[-\mathcal{U}, \mathcal{U}]$. During the first encryption, $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(l)}$ add $\mathbf{v}_j^{(k,l)}$ to their fingerprinted copies component by component, and calculate $\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{k,l}) = \widetilde{\mathbf{X}}_j^{(k)} + \mathbf{v}_j^{(k,l)}$ and $\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(l)}, K^{k,l}) = \widetilde{\mathbf{X}}_j^{(l)} + \mathbf{v}_j^{(k,l)}$, respectively. Thus, $||\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{k,l}) - \text{Enc}_1(\widetilde{\mathbf{X}}_j^{(l)}, K^{k,l})||^2 = ||\widetilde{\mathbf{X}}_j^{(k)} + \mathbf{v}_j^{(k,l)} - \widetilde{\mathbf{X}}_j^{(l)} - \mathbf{v}_j^{(k,l)}||^2 = ||\widetilde{\mathbf{X}}_j^{(k)} - \widetilde{\mathbf{X}}_j^{(l)}||^2$, and (1) 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.

4.2. Autonomous Selfish Colluder Identification

The key steps in the autonomous selfish colluder detection and identification are: for each frame j in the video sequence,

Step 1 Grouping: Colluders randomly divide themselves into two subgroups SC_1 and SC_2 where $SC_1 \cup SC_2 = SC$ and $SC_1 \cap SC_2 = \emptyset$. Colluders in SC_1 randomly select an *assistant* $\mathbf{u}^{(i_1 \in SC_1)}$ to help colluders in SC_2 calculate $\{D_j(k,l)\}_{k,l \in SC_2}$. Similarly, $\mathbf{u}^{(i_2 \in SC_2)}$ is randomly selected to help colluders in SC_1 calculate $\{D_j(k,l)\}_{k,l \in SC_1}$.

Step 2 Encryption: Assume that K^{SC_1} is a key that is shared by colluders in SC_1. Each colluder $\mathbf{u}^{(k)}$ in SC_1 generates a secret key K^{k,i_2} shared with the selected assistant $\mathbf{u}^{(i_2 \in SC_2)}$. $\mathbf{u}^{(k)}$ encrypts his/her fingerprinted copy $\widetilde{\mathbf{X}}_j^{(k)}$ with key K^{SC_1} and K^{k,i_2} in the same way as in Section 4.1. Then, $\mathbf{u}^{(k)}$ transmits the encrypted fingerprinted copy, $\text{Enc}_2\left(\text{Enc}_1(\widetilde{\mathbf{X}}_j^{(k)}, K^{SC_1}), K^{k,i_2}\right)$, to $\mathbf{u}^{(i_2)}$. Colluders in SC_2 follow the same procedure.

Step 3 Calculation of $\{D_j\}$: After decrypting all the received bit streams, for each pair of colluders $(\mathbf{u}^{(k)}, \mathbf{u}^{(l)})$ in subgroup SC_1, the selected assistant $\mathbf{u}^{(i_2 \in SC_2)}$ calculates $\widetilde{D}_j(k,l)$, and then broadcasts $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_1}$ to colluders in SC_1, together with his/her digital signature. $\mathbf{u}^{(i_1)}$ in SC_1 repeats the same process.

Step 4 Selfish Colluder Identification: Given $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_1}$, colluders in SC_1 apply the same method as in [5] to detect and identify selfish colluders in SC_1. Similarly, attackers in SC_2 examine $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$ and identify selfish colluders in SC_2.

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

The performance of the proposed autonomous selfish colluder identification algorithm depends on the correctness of $\{\widetilde{D}_j(k,l)\}$. If all the selected assistants give the other attackers correct values of $\{\widetilde{D}_j(k,l)\}$, the autonomous scheme has the same performance as that with a trusted ringleader in [5], and $P_{md} = 0$ and $P_{fa} = 0$. Here, we assume that if colluders in SC_h are selected to help calculate $\{\widetilde{D}_j\}$, they give other attackers correct values of $\{\widetilde{D}_j(k,l)\}$.

A unique issue in autonomous selfish colluder identification is that, two or more selfish colluders can collaborate with each other to change the detection results and prevent their fellow colluders from detecting their selfish behavior. For example, assume that there are two selfish colluders $\mathbf{u}^{(k_1)}$ and $\mathbf{u}^{(k_2)}$, and they are in different subgroups. Without loss of generality, assume that $k_1 \in SC_1$ and $k_2 \in SC_2$. If $\mathbf{u}^{(k_1)}$ is selected as the assistant to help colluders in SC_2 calculate $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$, $\mathbf{u}^{(k_1)}$ can modify the values of $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$ and let them follow the same distribution. Then, from [5], other colluders in SC_2 can not identify $\mathbf{u}^{(k_2)}$ as a selfish colluder and they make a miss-detection error. $\mathbf{u}^{(k_1)}$ can also change the values of $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$ to let attackers in SC_2 falsely accuse other attackers in SC_h as selfish colluders. Therefore, by collaborating with each other, a group of selfish colluders can change the values of $\{\widetilde{D}_j(k,l)\}$ and cause detection errors during autonomous selfish colluder detection and identification.

4.3. Multiple Assistants Selected from Each Subgroup

In order to manipulate the detection results, at least one of the selfish colluders has to be selected to help calculate $\{\widetilde{D}_j\}$. To reduce the chance that these selfish colluders can successfully change the detection results, colluders can select multiple assistants from each subgroup to calculate $\{\widetilde{D}_j\}$ and use majority vote when identifying selfish colluders.

For each frame j, to detect and identify selfish colluders in SC_1,

- m attackers are randomly selected from SC_2 to help calculate $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_1}$, and $\mathbf{A}_j(SC_2) = \{i_{2,1}, i_{2,2}, \cdots, i_{2,m}\}$ 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 4.2, encrypt their fingerprinted copies twice and transmit them to $\mathbf{u}^{(i_{2,n})}$.
- Each selected assistant $i_{2,n}$ in $\mathbf{A}_j(SC_2)$ calculates $\widetilde{D}_j^{i_{2,n}}(k,l)$ for all $k, l \in SC_1$, and broadcasts the results to attackers in SC_1 together with his/her digital signature.
- For every colluder $\mathbf{u}^{(k)}$ in SC_1 who does not process his/her copy before collusion, given $\{\widetilde{D}_j^{i_{2,n}}(k,l)\}_{k,l \in SC_1}$ received from $i_{2,n} \in \mathbf{A}_j(SC_2)$, $\mathbf{u}^{(k)}$ examines $\{\widetilde{D}_j^{i_{2,n}}(k,l)\}_{k,l \in SC_1}$, and sets $v_j^{(k)}(n,l) = 1$ if colluder $l \in SC_1$ is identified as a suspicious selfish colluder. Otherwise, $v_j^{(k)}(n,l) = 0$. Then, $\mathbf{u}^{(k)}$ applies majority vote to the m detection results $\{v_j^{(k)}(n,l)\}_{n=1,\cdots,m}$. If $\sum_{n=1}^{m} v_j^{(k)}(n,l) \geq \lceil m/2 \rceil$, then $\Upsilon_j^{(k)}(l) = 1$. $\Upsilon_j^{(k)}(l) = 0$ otherwise.

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

When more than half of the selected m assistants in $\mathbf{A}_j(SC_2)$ are selfish colluders, they can still cause detection errors. Thus, when estimating the identities of the selfish colluders, colluders in SC_h should jointly consider the detection results from all frames.

For each frame j in the video sequence, define

$$I_j(k,l) \triangleq \begin{cases} 1 & \text{if } k \in SC_1 \text{ and } l \in SC_1, \\ 1 & \text{if } k \in SC_2 \text{ and } l \in SC_2, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

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

For colluder $\mathbf{u}^{(k)}$ who does not apply pre-collusion processing, to determine whether $\widetilde{\mathbf{X}}^{(l)}$ is the original copy that $\mathbf{u}^{(l)}$ received from the content owner, $\mathbf{u}^{(k)}$ jointly considers all the detection results $\{\Upsilon_j^{(k)}(l)\}_{j \in F(k,l)}$ that he/she has, and identifies $\mathbf{u}^{(l)}$ as a selfish colluder if the average of $\{\Upsilon_j^{(k)}(l)\}_{j \in F(k,l)}$ is above a threshold α. $\mathbf{u}^{(k)}$ then outputs the estimated selfish colluder set $\widehat{SC}_s^{(k)} = \left\{ l : \sum_{j \in F(k,l)} \Upsilon_j^{(k)}(l) / |F(k,l)| > \alpha \right\}$. Detailed analysis of the parameter selection for m and α are in [9].

5. SIMULATION RESULTS

We test on the first 300 frames of sequence carphone to evaluate the performance of the autonomous selfish colluder detection and identification algorithm. Human visual model based spread spectrum embedding [7] is used to embed fingerprints into the host signal, and orthogonal fingerprints are assigned to different users. During pre-collusion processing, the selfish colluders apply temporal filtering in [4], and the newly generated frames have PSNR of $40dB$ when compared with the originally received ones. Each selfish colluder processes his/her copy independently before collusion.

For each frame in the video sequence, each subgroup selects $m = 3$ colluders to help the other subgroup calculate $\{\widetilde{D}_j(k,l)\}$, and they follow Section 4.3 to identify selfish colluders. We assume that if selected as assistants, colluders in SC_h tell other attackers correct values of $\{\widetilde{D}_j(k,l)\}$. We further assume that all selfish colluders who apply pre-collusion processing collaborate with each other to prevent being detected by their fellow attackers. If a selfish colluder $i \in SC_1$ is selected to help attackers in SC_2 calculate $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$, we assume that $\mathbf{u}^{(i)}$ changes the histogram of $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$ such that none of the selfish colluders in SC_2 can be detected. We also assume that $\mathbf{u}^{(i)}$ randomly selects another colluder $k \in SC_2$ who does not apply pre-collusion processing, and change the values of $\{\widetilde{D}_j(k,l)\}_{k,l \in SC_2}$ so that other colluders falsely identify $\mathbf{u}^{(k)}$ as selfish. Same for selfish colluders in SC_2.

From Figure 2, the proposed autonomous selfish colluder detection and identification algorithm can achieve error-free performance even when a small number of selfish colluders collaborate with each other to manipulate the detection results. If less than 15% of the colluders are selfish, the proposed algorithm can always correctly identify all selfish colluders without falsely accusing any others; while P_{md} increases quickly as the number of selfish colluders is above 15% of the total number of colluders. Note that in Figure 2, even when there are a large number of selfish colluders, the proposed algorithm never falsely accuses any colluders in SC_h as selfish. This is because in our simulations, when selected as assistants to help calculate $\{\widetilde{D}_j\}$, selfish colluders randomly choose another attacker in SC_h and accuse him/her as selfish. Colluders in SC_h can easily correct this false-alarm error by using majority vote and jointly considering detection results from all video frames. Therefore, $P_{fa} = 0$ even if the total number of selfish colluders is large.

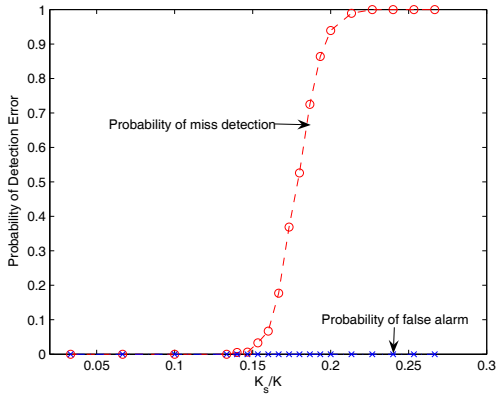

Fig. 2. Simulation results of P_{fa} and P_{md} on the first 300 frames of sequence carphone. There are a total of $K = 150$ colluders. SC_1 and SC_2 are of the same size, and each has 75 colluders. K_s is the number of selfish colluders. $m = 3$ and $\alpha = 0.85$.

6. CONCLUSIONS

This paper studies the traitor-within-traitor behavior forensics and investigates how attackers detect selfish pre-collusion processing and identify selfish colluders to protect their own interest. We propose an autonomous selfish colluder detection algorithm, in which colluders help each other identify selfish colluders, and analyze its performance. The proposed algorithm protects the secrecy of the fingerprinted coefficients in all copies and prevents colluders from framing each other. Our simulation results show that the proposed algorithm can accurately identify all selfish colluders without falsely accusing any others, even when a small group of selfish colluders collaborate with each other to manipulate the detection results.

7. REFERENCES

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

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

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

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

[5] H. V. Zhao and K. J. R. Liu, "Selfish colluder detection and identification in traitors within traitors," *IEEE Int. Conf. on Image Proc.*, Oct. 2006.

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

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

[8] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] H. V. Zhao and K. J. R. Liu, "Multi-user collusion behavior forensics in traitors within traitors: Selfish colluder detection and identification," *submitted to IEEE Trans. on Information Forensics and Security*.