ABSTRACT

Title of dissertation: Behavior Modeling and Forensics for Multimedia Social Networks

Wan-Yi Lin, Doctor of Philosophy, 2009

Dissertation directed by: Professor K. J. Ray Liu
Department of Electrical and Computer Engineering

Within the past decades, the explosive combination of multimedia signal processing, communications and networking technologies has facilitated the sharing of digital multimedia data and enabled pervasive digital media distribution over all kinds of networks. People involved in the sharing and distribution of multimedia contents form *multimedia social networks* in which users share and exchange multimedia content, as well as other resources. Users in a multimedia social network have different objectives and influence each other's decision and performance. It is of ample importance to understand how users interact with and respond to each other and analyze the impact of human factors on multimedia systems. This thesis illustrates various aspects of issues and problems in multimedia social networks via two case studies of human behavior in multimedia fingerprinting and peer-to-peer live streaming.

Since media security and content protection is a major issue in current multimedia systems, this thesis first studies the user dynamics of multimedia fingerprinting social networks. We investigate the side information which improves the traitor-tracing performance and provide the optimal strategies for both users (fingerprint detector and the colluders) in the multimedia fingerprinting social network. Furthermore, before a collu-

sion being successfully mounted, the colluders must be stimulated to cooperate with each other and all colluders have to agree on the attack strategy. Therefore, not all types of collusion are possible. We reduce the possible collusion set by analyzing the incentives and bargaining behavior among colluders. We show that the optimal strategies designed based on human behavior can provide more information to the fingerprint detector and effectively improve the collusion resistance.

The second part of this thesis focuses on understanding modelling and analyzing user dynamics for users in various types of peer-to-peer live streaming social networks. We stimulate user cooperation by designing the optimal, cheat-proof, and attack-resistant strategies for peer-to-peer live streaming social networks over Internet as well as wireless networks. Also, as more and more smart-phone users subscribe to the live-streaming service, a reasonable market price has to be set to prevent the users from reselling the live video. We start from analyzing the equilibrium between the users who want to resell the video and the potential buyers to provide the optimal price for the content owner.

Behavior Modeling and Forensics for Multimedia Social Networks

by

Wan-Yi Lin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:
Professor Professor K. J. Ray Liu, Chair/Advisor
Professor Rama Chellappa
Professor Min Wu
Professor Yu Chen
Professor Lawrence C. Washington

Dedication

To my family.

Acknowledgments

Completing my Ph.D. thesis is a milestone in my career that would not have been possible without the help and support of special individuals to whom I dedicate the following few words that can not capture my appreciation for them.

First, I would like to express my sincere gratitude to my advisor, Prof. K. J. Ray Liu, who provided me the opportunity to be the researcher I am today. for his guidance and support during my study in University of Maryland. Dr. Liu has been my inspiration for doing exciting world-class research over the past few years. He was always available whenever I needed his guidance through problems in my research or even in my life. He has played a significant role in both my professional and personal development in Maryland, and his vision, energy and desire for excellence have influenced me with lifetime benefits.

I also want to thank Prof. Rama Chellappa, Prof. Min Wu, Prof. Yu Chen, and Prof. Lawrence Washington for agreeing to serve in my thesis committee.

I would like to take this chance to thank members in the Signal and Information group for their friendship, encouragement and help. I always feel lucky to be in such an energetic and excellent group, and their accompanying during my stay in Maryland has helped me to survive my Ph.D study. I also want to thank all my friends, it is because of their help that my life in the past years is much more joyful and colorful.

Finally, I give my heartfelt gratitude to my parents and my brother, my role models and the three most important persons in my life. Without their love, unconditional support and countless sacrifices, I could never accomplish so much and reach this milestone in my life. I dedicate this thesis to them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

A social network is a structure of nodes (including individuals and organizations) that are connected with each other via certain types of relations, for examples, values, friendship, conflict, financial exchange, trade, etc. People have been studying methodologies to formulate the relationships between members at all scales, from interpersonal to international, and social network analysis has become a popular topic in sociology, economics, information science and many other disciplines.

In a multimedia social network community, a group of users form a dynamically changing network infrastructure to share and exchange data, often multimedia content, as well as other resources [1]. In the past decades, we have witnessed the emergence of large-scale multimedia social network communities, for instance, Napster, flickr and YouTube, and the Internet traffic has shifted dramatically from HTML text pages to multimedia file sharing [2]. These multimedia social networks have millions of users worldwide, and a crucial issue there is to understand the user dynamics that influence human's behavior. As an example, a study showed that in a campus network, peer-to-peer file sharing can consume 43% of the overall bandwidth, which is about three times of the WWW traffic

[3]. This poses new challenges to the efficient, scalable and robust sharing of multimedia over large and heterogeneous networks.

By participating in multimedia social networks, users receive rewards by being able to access extra resources from their peers, and they also contribute their own resources. Users aim to maximize their own payoff by participating in multimedia social networks, and different users have different (and often conflicting) objectives and full cooperation cannot be enforced since users will try all their means to increase their own profit. For example, in a peer-to-peer file-sharing system, users pool together the resources and cooperate with each other to provide an inexpensive, highly scalable and robust platform for distributed data sharing [4, 5]. However, due to the voluntary participation nature in many multimedia social networks and the limited resources available to each user, users' full cooperation cannot be guaranteed unless there exist powerful central authorities who mandate and enforce user cooperation. A recent study of Napster and Gnutella showed that many users are free riders and 25% of the users in Gnutella share no files at all [6]. Thus, as demonstrated in Figure 1.1, an important issue in multimedia social networks is to understand the optimal strategies that users will play when negotiating with each other and achieve fairness. Game theory [7, 8] provides a fundamental tool to study the fairness dynamics among users, and the Nash Equilibrium analysis gives the optial strategies from which no user has incentives to deviate.

The above discussion focuses on analyzing the behavior of *rational users* who are willing to contribute their own resources if cooperation with others can help improve their payoff. They are honest when exchanging information and negotiating with other users. There are also *selfish users* who wish to consume others' resources with little or

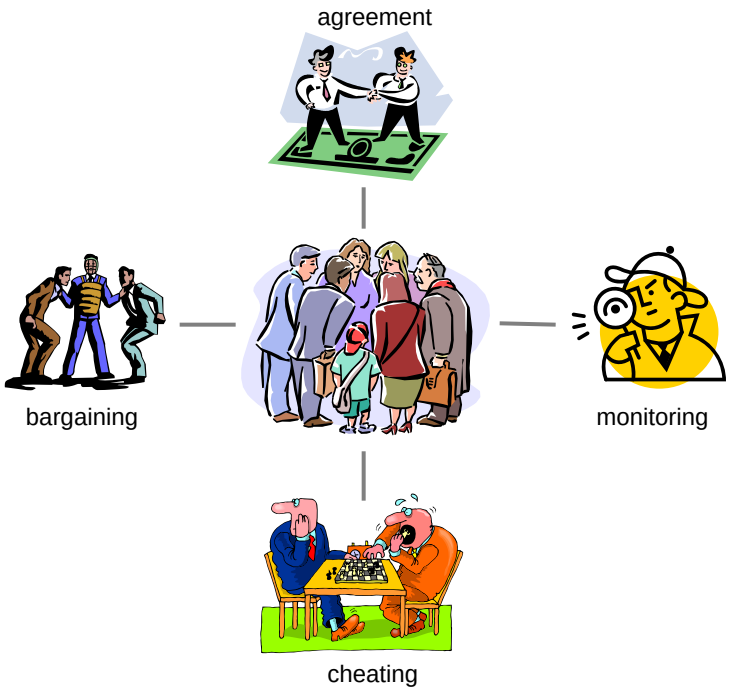


Figure 1.1: User dynamics in social networks.

no contribution of their own. If necessary, these selfish users might even cheat during the negotiation process in order to maximize their own payoff, as shown in Figure 1.1. Furthermore, there might exist *malicious users*, whose goal is to attack and sabotage the system. For example, in peer-to-peer file-sharing systems, they tamper the media files with the intention of making the content useless (the so-called "pollution" attack) [3]. They can also launch the Denial of Service (DoS) attack to exhaust other users' resources and make the system unavailable [9]. It is possible that a few malicious users collude with each other to effectively attack the system, for example, the flooding Distributed Denial of Service (DDoS) attack in peer-to-peer file-sharing systems. Therefore, cheat free and attack resistance are fundamental requirements in order to achieve user cooperation and provide reliable services in multimedia social networks.

To model and analyze human dynamics in multimedia social networks when there

exist selfish and malicious users, the first step is to study the strategies that selfish colluders can use to cheat and those that malicious users adopt to attack the system. The next issue is to implement monitoring mechanisms to detect and identify misbehaving users, as illustrated in Figure 1.1. A challenging issue here is that the monitoring mechanisms should be able to distinguish "intentional" misbehavior (for example, intentional manipulation of multimedia content) from "innocent" ones (for example, transmission error and packet loss in erroneous and congested networks). The above investigation will facilitate the design of cheat-proof and attack-resistant strategies, which make non-cooperation non-profitable, thus unattractive to selfish users, and minimize the damage to the system caused by malicious users.

In a nutshell, multimedia social networks involve a large number of users of different types with different objectives, and before multimedia social network communities become successful, they must provide a predictable and satisfactory level of service. It is of ample importance to understand how users interact with and respond to each other and analyze the impact of human factors on multimedia systems. Such an understanding provides fundamental guidelines to better design of multimedia systems and networking, and to offer more secure and personalized services. All these are essential factors to maximize the overall system performance and minimize the damage caused by malicious users. In addition, for different multimedia social networks, different structures will result in different mechanisms to monitor user behavior and achieve cheat proof and attack resistance. The goals of this thesis are to illustrate why human behavior factors are important and emerging issues strongly related to signal processing, and to demonstrate that signal processing can be effectively used to model, analyze and perform behavior

forensics for multimedia social networks.

1.2 Dissertation Outline and Contributions

From the discussion above, behavior modelling for multimedia social networks is a new paradigm that provides guidelines for both the system designer and the users in multimedia systems. This thesis develops and analyzes methodologies to model the user behavior dynamics and investigate the optimal strategies for multimedia fingerprinting social networks and peer-to-peer live streaming social networks over Internet and wireless networks. We envision that insights from a wide range of disciplines, such as game theory, networking, and economics, will help improve the understanding of human dynamics and its impact on signal processing and ultimately lead to systems with more secure, efficient and personalized services. The rest of the thesis is organized as follows.

1.2.1 Equilibriums of Multimedia Fingerprinting Social Networks with Side Information (Chapter 2)

Multimedia fingerprinting is an emerging technology that offers proactive post-delivery protection of multimedia content [10–14]. It labels each distributed copy with the corresponding user's identification information, called *fingerprint*, which can be used to track the distribution of multimedia data and to identify the source of illicit copies. Multiuser collusion is a cost-effective attack against multimedia fingerprinting, where a group of attackers work collectively to remove or attenuate the embedded fingerprints [15, 16].

In multimedia fingerprinting, colluders and the fingerprint detector form a multi-

media social network: colluders who apply multiuser collusion attempt to remove the identifying fingerprints in their copies, and the digital rights enforcer detects the embedded fingerprints in the suspicious copy to capture colluders. It is obvious that the colluders and the fingerprint detector influence each other's performance and decision: given a colluded copy, the detector always wants to adjust his/her detection strategy to achieve the best possible traitor-tracing performance. Meanwhile, during collusion, the colluders try the best to minimize their risk based on the available information about the detection procedure. There are many collusion strategies that the colluders can use to remove the identifying fingerprints. Also, the detector can apply different detection strategies to identify the colluders. Thus, the dynamics between the colluders and the fingerprint detector is complicated.

Side information is the information other than the colluded multimedia content that can help increase the probability of detection. We propose a game-theoretical framework to model and analyze the complex dynamics between the colluders and fingerprint detector. In this thesis, we model the colluder-detector behavior dynamics as a two-stage game, where the fingerprint detector tries to maximize the detection performance while the colluders adjust the collusion parameters to minimize their risk under the fairness constraint. We first study the impact of side information in multimedia fingerprinting and show that the statistical means of the detection statistics can help the fingerprint detector significantly improve the collusion resistance. We then investigate how to probe the side information and model the dynamics between the fingerprint detector and the colluders as a two-stage extensive game with perfect information. We find the equilibrium of the colluder-detector game using backward induction and show that the min-max solution is

a Nash equilibrium, which gives no incentive for everyone in the multimedia fingerprint social network to deviate. This thesis demonstrates that the proposed side information can help improve the system performance, and the self-probing fingerprint detector has almost the same performance as the optimal correlation-based detector. Also, we provide the solutions to how to reach optimal collusion strategy and the corresponding detection, thus lead to a better collusion resistance [17].

1.2.2 Behavior Analysis in Colluder Social Networks (Chapter 3)

During collusion, a group of attackers collectively and effectively attack multimedia fingerprinting system and use multimedia content illegally. Before the collusion being successful, the colluders have to make agreement on how to distribute the risk and reward by redistributing the colluded multimedia signal. Hence, the colluders in a multimedia fingerprinting system also form a social network.

To have a better understanding of the attackers' behavior during collusion to achieve fairness, we first model the dynamics among colluders as a non-cooperative game, propose a general model of utility functions and study four different bargaining solution of this game. Our framework considers both the colluders' risk of being detected by the digital rights enforcer and the reward received from illegal usage of multimedia content. Moreover, the market value of the redistributed multimedia content is time sensitive. The earlier the colluded copy being released, the more the people who are willing to pay for it. Thus the colluders have to reach agreement on how to distribute reward and the probability being detected among themselves as soon as possible. We further incorporate the

time-sensitiveness of the colluders' reward, and study the time-restricted bargaining equilibrium. We also investigate how do the colluders select fellow attackers to maximize colluders' payoffs. We provide the solution to the equilibrium that all the colluders have no inventive to disagree in order to maximize their own payoff.

Our analysis shows that colluding with more attackers does not always increase an attacker's utility, and attackers may not always want to cooperate with each other. We first examine the necessary conditions for attackers to collude together, and study how they select the collusion parameters such that cooperation benefits all colluders. We then study how the number of colluders affects each attacker's utility, and investigate the optimum strategy that an attacker should use to select fellow attackers in order to maximize his or her own payoff [18].

1.2.3 Incentive Cooperation Strategies for Peer-to-Peer Live Multimedia Streaming Social Networks (Chapter 4)

With recent advance in networking, multimedia signal processing, and communication technologies, we witness the emergence of large-scale multimedia social networks, where millions of users form a distributed and dynamically changing infrastructure to share media content. Peer-to-Peer (P2P) live streaming using the mesh-pull architecture [19] is one of the biggest multimedia social networks on the Internet and has enjoyed many successful deployments to date, for example, CoolStreaming, pplive and SopCast [20–27]. Users in a P2P live-streaming system watch live broadcasting TV programs over networks simultaneously. The system relies on voluntary contributions of resources from

individual users to achieve high scalability and robustness and to provide satisfactory performance.

In peer-to-peer live-streaming social networks, users cooperate with each other to provide a distributed, highly scalable and robust platform for live streaming applications. A an essential issue to be resolved first is to stimulate user cooperation. In addition, users in P2P live streaming systems are strategic and rational, in that they are likely to manipulate any incentive system (for example, by cheating) to maximize their payoff. As such, in large-scale social networks, users influence each other's decisions and performance, and there exist complicated dynamics among users. It is of ample importance to investigate user behavior and analyze the impact of human factors on multimedia social networks. We propose a game-theoretic framework to model user behavior and designs incentive-based strategies to stimulate user cooperation in peer-to-peer live streaming. We first analyze the Nash equilibrium and the Pareto optimality of 2-person game and then extend to multiuser case. We also take into consideration selfish users' cheating behavior and malicious users' attacking behavior. Both our analytical and simulation results show that the proposed strategies can effectively stimulate user cooperation, achieve cheat free, attack resistance and help to provide reliable services [28, 29].

1.2.4 Cooperation Stimulation Strategies for Peer-To-Peer Wireless Live Video-Sharing Social Networks (Chapter 5)

Recent development on wireless local area network (WLAN) enable users to utilize WLAN with low cost and high quality of service [30–33]. Users watching live streaming in the

same wireless network share the same limited bandwidth of backbone connection to the Internet, thus they might want to cooperate with each other to obtain better video quality. These users form a wireless live-streaming social network. Every user wishes to watch video with high quality while paying as little as possible cost to help others. Given the unstable wireless channel and less user in the wireless network, the attackers can cause more damage to the wireless live streaming social network than in the Internet phenomenon. Therefore, the malicious-user identification mechanism has to be faster and more reliable.

This thesis focuses on providing incentives for user cooperation. We propose a game-theoretic framework to model user behavior and to analyze the optimal strategies for user cooperation simulation in wireless live streaming. We first analyze the Pareto optimality and the time-sensitive bargaining equilibrium of the two-person game. We then extend the solution to the multiuser scenario. We also consider potential selfish users' cheating behavior and malicious users' attacking behavior and analyze the performance of the proposed strategies with the existence of cheating users and malicious attackers. We introduce the concept of trust to further bound the damage caused by malicious attack. Both our analytical and simulation results show that the proposed strategies can effectively stimulate user cooperation, achieve cheat free and attack resistance, and help provide reliable services for wireless live streaming applications [34].

1.2.5 Optimal Price Setting for Mobile Live Video Service (Chapter 6)

The mobile phone is becoming the most popular consumer device over all kinds of electronic products. Recently, the development of smart phones enables users to watch live TV program by subscribing data plans from cellphone service providers. Nowadays, the price of data plans are set only to compete with other service providers. However, due to the high popularity and the mobility of the mobile phones, the subscribers can form a network to re-sell the live video to the non-subscribers. Such re-selling mechanism is a potential competitor for the service provider. The service provider has to set a reasonable price that can prevent such re-selling behavior to protect the provider's profit.

In this thesis, we analyze the optimal price setting for the service provider by investigating the equilibrium between the re-sellers and the non-subscribers. We model the behavior between the re-sellers and the non-subscribers as a hybrid Stackelburg auction game and find the optimal price for both groups of users. Such analysis can help design a reasonable price for the less-competitor mobile live-streaming market to improve the quality of service for end users.

Chapter 2

Behavior Analysis in Colluder Social Networks

During collusion, the colluders share the reward from the illegal usage of multimedia as well as the risk of being captured by the digital rights enforcer. Before collusion relationship can be established, an agreement must be reached regarding how to distribute the risk and the reward. Therefore, the colluders in the digital fingerprinting system also form a social network. In the *colluder social network*, users collaborate with each other to reduce their chance of being caught by the digital right enforcer and share the reward of redistributing the colluded multimedia signal. However, each colluder prefers the collusion that favors his/her payoff the most (lowest risk and highest reward), and different colluders have different preferences. To address such a conflict, a critical issue for the colluders is to decide how to fairly distribute the risk and the reward. It is of ample importance to understand how colluders negotiate with each other to achieve fairness of the attack.

To analyze the dynamics among the members in colluder social network, we model the user behavior as a non-cooperative game where each colluder tries to maximize his/her individual payoff under the fairness constraint. First, the attackers have to decide whether

to collude with people who have the high resolution copy, or is it better to cooperate with those who have the base layer only, and how many people should them collude with. To minimize the risk, colluders are always willing to cooperate with each other since it reduces all attackers' risk, and a colluder should find as many fellow attackers as possible. However, colluding with more attackers also means sharing with more people the reward from illegal usage of multimedia and, therefore, colluders may not always want to cooperate with each other. In addition, when colluders receive copies of different resolutions, an attacker also needs to decide with whom to collude, which has been seldom addressed in the literature.

After finding the best partners, the colluders will bargain to reach the agreement of fairly distributing the probability of being detected and the reward of redistributing the multimedia content. In this chapter, we consider different definitions of fairness, investigate how the colluders would like to share the risk and the reward, and study different bargaining solutions: Nash-Bargaining, Max-Min, and Max-Sum solutions. Also, users in the colluder social network may have different social positions, thus some users may be willing to take higher risk and higher reward at the same time, while other users may be more concerned about risk and want to take lower risk and lower reward. We also take this phenomenon into consideration and study the proportional fairness collusion.

In addition, the other side of the fingerprinting system, the fingerprint detector, also has to choose its optimal strategy according to various types of collusion. The colluders will agree on the bargaining solutions if and only if the bargaining solutions are the best strategies they can choose under the fairness criteria. Therefore, it is crucial for both the colluders and the digital rights enforcer to investigate the optimal strategies for

each other's choices and reach the equilibrium for the multimedia fingerprinting social network.

The rest of the chapter is as follows. We define the general utility functions for the colluders and investigate under what conditions will the attackers collaborate with each other in Section 2.1. We then analyze the fair collusion in Section 2.2, including bargaining solutions with and without time constraint. The analysis of equilibriums for the colluder-detector game in the multimedia fingerprinting social networks is studied in Section 2.3 and conclusions are drawn in Section 2.4.

2.1 Game-theoretic Formulation and Incentives for Multi-user Collusion

In this section, we will first introduce the multimedia fingerprinting system with side information as discussed in Chapter 2, and then define the utility function of every user in the colluder social network. Based on each colluder's utility, we will discuss the criteria that stimulates collusion.

2.1.1 Multimedia Fingerprinting with Side Information

To improve the detection performance, in Chapter 2 we investigated techniques for the digital rights enforcer to explore the special characteristics of the colluded copy, probe side information about multiuser collusion, and select the optimum detection strategy.

In the two-layer scalable multimedia fingerprinting system in Section 3.1, for user

$u^{(i)}$ who receives a high resolution fingerprinted copy, let $\mathbf{W}_b^{(i)}$ and $\mathbf{W}_e^{(i)}$ denote $u^{(i)}$'s fingerprints that are embedded in the base layer and the enhancement layer, respectively. Let $\mathbf{Y}_b$ and $\mathbf{Y}_e$ be the fingerprints extracted from the base layer and the enhancement layer of the test copy, respectively.

In such a system, there are many different ways to determine if $u^{(i)}$ participates in collusion. For example, the fingerprint detector can use $\mathbf{Y}_b$ and $\mathbf{Y}_e$ collectively to determine whether $u^{(i)}$ is a colluder, and the fingerprint detector uses the collective detection statistic

$$TN_c^{(i)} = \frac{\langle \mathbf{Y}_b, \mathbf{W}_b^{(i)} \rangle + \langle \mathbf{Y}_e, \mathbf{W}_e^{(i)} \rangle}{\sqrt{||\mathbf{W}_b^{(i)}||^2 + ||\mathbf{W}_e^{(i)}||^2}} \tag{2.1}$$

to measure the similarity between $\mathbf{Y}$ and $\mathbf{W}^{(i)}$. From the analysis in Chapter 2, with orthogonal fingerprint modulation, if the detection noise is i.i.d. Gaussian $\mathcal{N}(0, \sigma_n^2)$, then $TN_c^{(i)}$ follows the Gaussian distribution

$$TN_c^{(i)} \sim \begin{cases} \mathcal{N}\left(\mu_c^{be} = \frac{(1-\beta)N_b + N_e}{K^{be}\sqrt{N_b+N_e}}\sigma_w, \sigma_n^2\right), & \text{if} \quad i \in SC^{be}, \\ \mathcal{N}(0, \sigma_n^2), & \text{if} \quad i \notin SC^{be}. \end{cases} \tag{2.2}$$

In (2.2), N_b and N_e are the lengths of the fingerprints embedded in the base layer and the enhancement layer, respectively, and σ_w^2 is the variance of the fingerprint $\mathbf{W}^{(i)}$.

The fingerprint detector can also use the fingerprint extracted from each individual layer to determine whether $u^{(i)}$ participates in collusion, and uses

$$TN_t^{(i)} = \frac{\langle \mathbf{Y}_t, \mathbf{W}_t^{(i)} \rangle}{||\mathbf{W}_t^{(i)}||} \sim \begin{cases} \mathcal{N}\left(\mu_t^{be}, \sigma_n^2\right), & \text{if} \quad i \in SC^{be}, \\ \mathcal{N}\left(0, \sigma_n^2\right), & \text{if} \quad i \notin SC^{be}, \end{cases} \tag{2.3}$$

to calculate the similarity between the extracted fingerprint and the original fingerprint,

Here, the subscript 't' is the layer index and is either 'b' (the base layer) or 'e' (the enhancement layer). In (2.3), $\mu_b^{be} = \frac{(1-\beta)\sqrt{N_b}\sigma_w}{K^{be}}$ and $\mu_e^{be} = \frac{\sqrt{N_e}\sigma_w}{K^{be}}$.

Comparing (2.2) and (2.3), $TN_c^{(i)}$, $TN_b^{(i)}$ and $TN_e^{(i)}$ have the same variance but different statistical means, and the one with the largest mean gives the best detection performance. From the analysis in Chapter 2, μ_b^{be} is always smaller than $\max(\mu_c^{be}, \mu_e^{be})$, and TN_b^{be} should not be used to identify colluders. When $\beta < \beta^+ \triangleq \frac{\sqrt{N_b+N_e}\left(\sqrt{N_b+N_e}-\sqrt{N_e}\right)}{N_b}$, $\mu_c^{(i)} > \mu_e^{(i)}$ and $TN_c^{(i)}$ is more robust against collusion attacks. If $\beta > \beta^+$, then $\mu_e^{(i)} > \mu_c^{(i)}$ and $TN_e^{(i)}$ provides a better performance.

2.1.2 Game Model

During collusion, every user in the colluder social network wants to minimize his/her own risk and maximizes his/her own reward.

For colluder $\mathbf{u}^{(i)}$, his/her payoff function $\pi^{(i)}$ should be composed of two terms: colluder i's loss if being detected plus his/her reward as follows:

$$\pi^{(i)} = -P_d^{(i)} * L^{(i)} + \left(1 - P_d^{(i)}\right) R^{(i)}. \tag{2.4}$$

In (2.4), $P_d^{(i)}$ and L stand for colluder $\mathbf{u}^{(i)}$'s probability and loss of being detected, and $R^{(i)}$ is the reward that $\mathbf{u}^{(i)}$ gets after redistributing the colluded multimedia content and sharing with other colluders. Since the total reward that will be shared by all the colluders by redistributing the colluded copy is proportional to the video quality. For instance, the pirated video with DVD quality would have higher value than the video with VCD quality. In temporal scalable video coding scenario, video quality is an increasing function of the number of frames. Here we aim to show how do different factors during collusion affect

the behavior of the colluders, thus we introduce a simple model that the video quality is proportional to the number of frames. As a result, $R^{(i)}$ can be modelled as:

$$R^{(i)} = \frac{F^c/F^{max}}{\left[\sum_{j=1}^{K}\left(F^{(j)}\right)^{\gamma} D\left(P_d^{(j)}\right)\right]/M}\left(F^{(i)}\right)^{\gamma} D\left(P_d^{(i)}\right). \tag{2.5}$$

Where F^c is the number of frames in the final colluded copy, and F^{max} is the largest number of frames among all the subscribers' copies, hence F^c/F^{max} illustrates higher quality of colluded copy gives higher reward to the colluders. For instance, in our system in Section 2.1.1, if all the colluders only received the lower quality copy, then $F^c = N_b$, $F^{max} = N_b + N_e$, F^c/F^{max} = 1- $N_e/(N_b + Ne) < 1$, which implies the colluders cannot get the full market value of the video. $F^{(i)}$ is the number of frames in $\mathbf{u}^{(i)}$'s copy; K is the total number of colluders, M is the total number of subscribers, and $D(\bullet)$ is a non-decreasing function. $\left(F^{(i)}\right)^{\gamma}$ illustrates colluders with higher-quality copies would have more reward since they already paid more money to subscribe to higher resolution copies , and γ is the factor to control how much extra reward the colluders with higher-resolution copies should get. For example, if γ = 0, then the reward is equally distributed among the colluders with the same quality copies, and larger γ indicates the reward distribution favors the colluders with higher-quality copies more. Different colluders have different evaluation of their risk. Therefore, some colluders might want to take higher risk, and in return, they would ask for more reward. $D\left(P_d^{(i)}\right)$ allows the colluders who take risk would have higher reward.

In the following sections, to simplify the analysis, we assume the colluders who receive the same quality copies agree to share the same probability of being detected as in Section 2.1.1. Hence, the bargaining process during collusion can be modelled as the

following game:

- **Players**: There are two players in the game. Colluders who receive the low-resolution copies act as a single player in the game and they have the same utility π^b, while while colluders who have the high-resolution copies act as a single player during the bargaining process and they have the same utility $\pi^{b,e}$. Denote all the colluder in SC^b as sc^b, and all the colluders in $SC^{b,e}$ be $sc^{b,e}$ in this game.

- **Strategies**: Based on the two-step collusion model in Chapter 2, the collusion parameter β controls the risk for both sc^b and $sc^{b,e}$. The control factors of the reward distribution (γ and $D\left(P_d^{(i)}\right)$) is declared before the game. Therefore, the players' possible strategies are all the possible values of β.

- **Cost**: For each player, joining the collusion and redistribute the colluded copy incur the probability of being detected by the fingerprint detector. Being accused by the detector causes a consequence of cost. Thus we model the cost of each player as the probability of being detected times its loss. The loss is the private information of each colluders, and a reasonable setting is the loss should be bounded by a maximal value L_{max}.

- **Reward**: The players gain reward by redistributing the colluded copy, and the reward is distributed among all the colluders. Here we assume the value of the colluded copy remains the same no matter how long does the collusion process take.

- **Utility function**: The utility function is considered as the reward minus the expected cost as in (2.4).

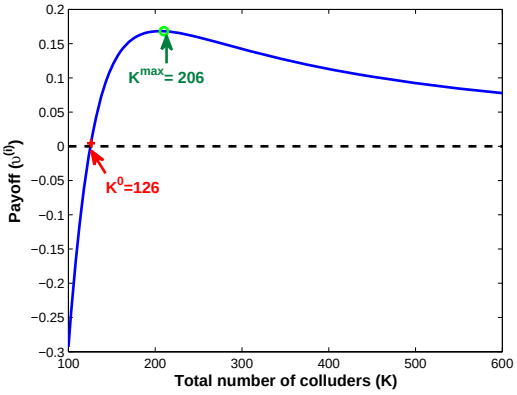


Figure 2.1: $\pi^{(i)}$ when all colluders receive fingerprinted copies of high resolution. $N_b = N_e = 50000$ and $\theta = 50$. The probability of falsely accusing an innocent user is $P_{fa} = 10^{-3}$.

2.1.3 Necessary Condition for Single-Resolution Collusion

We first discuss the situations that the attackers will collaborate with each other for multi-user collusion. As an example, we assume that all attackers receive high resolution copies with base layer and enhancement layer, and they generate a colluded copy of high resolution, that is, $K = K^{be}$ and $f_c = 1$. The analysis is similar for the scenario where all fingerprinted copies have the base layer only and thus omitted. In such a scenario, since all copies have the same resolution, there is no bargaining in collusion, and attackers simply average all copies that they have with equal weights. Based on (2.4), colluder $u^{(i)}$'s utility function can be simplified to

$$\begin{aligned} \pi^{(i)} &= -P_d^{(i)} + \left(1 - P_d^{(i)}\right)\frac{\theta}{K}, \\ \text{where} \quad P_d^{(i)} &= Q\left(\frac{h - \sqrt{N_b + N_e}\sigma_w/K}{\sigma_n}\right). \end{aligned} \tag{2.6}$$

Figure 2.1 shows an example of $\pi^{(i)}$ versus the total number of colluders K. In

Figure 2.1, the lengths of the fingerprints embedded in the base layer and the enhancement layer are $N_b = 50000$ and $N_e = 50000$, respectively. In Figure 2.1, we use $\theta = 50$ as an example to illustrate colluders' payoffs, and we observe similar trends with other values of θ. $\sigma_w^2 = \sigma_n^2 = 1$ and h is selected so that the probability of falsely accusing an innocent is 10^{-3}. From Figure 2.1, when $K < 126$, $\pi^{(i)} < 0$ due to $u^{(i)}$'s large probability of being detected. In this scenario, colluders may not want to use multimedia illegally since it is too risky. Furthermore, from Figure 2.1, colluding with more attackers does not always increase $u^{(i)}$'s payoff, and $\pi^{(i)}$ becomes a decreasing function of K when there are more than 206 attackers.

Let $K_0 \stackrel{\triangle}{=} \left\{ K : \pi^{(i)}(K-1) < 0, \pi^{(i)}(K) \geq 0 \right\}$ be the smallest K that gives $u^{(i)}$ a non-negative payoff. Attackers will collude with each other if and only if there are more than K_0 colluders and when they receive positive payoffs from collusion. Also, we define $K_{max} \stackrel{\triangle}{=} \arg_{K \geq K_0} \max \pi^{(i)}$ as the optimum K that maximizes colluder $u^{(i)}$'s utility when all attackers receive copies of the same resolution. A colluder should find a total of K_{max} attackers if possible to maximize his/her payoff. In the example in Figure 1, $K_0 = 126$ and $K_{max} = 206$.

2.1.3.1 Analysis of K_0

Given N_b, N_e and θ, to find the minimal number of colluders K_0, we solve the equation

$$\pi^{(i)}(K) = -Q\left(\frac{h - \sqrt{N_b + N_e}\sigma_w/K}{\sigma_n}\right) + \left[1 - Q\left(\frac{h - \sqrt{N_b + N_e}\sigma_w/K}{\sigma_n}\right)\right]\frac{\theta}{K} = 0,$$

$$\text{or equivalently,} \quad Q\left(\frac{h - \sqrt{N_b + N_e}\sigma_w/K}{\sigma_n}\right) = \frac{\theta}{K+\theta}. \tag{2.7}$$

Define $x = 1/K$ as the inverse of K. Then (2.7) can be rewritten as

$$Q\left(\frac{h-\sqrt{N_b+N_e}\sigma_w x}{\sigma_n}\right) = Q\left(a-b_s x\right) = 1-\frac{1}{1+\theta x}, \tag{2.8}$$

where $a = h/\sigma_n$ and $b_s = \sqrt{N_b+N_e}\sigma_w/\sigma_n$. It is difficult to find the exact analytical solution to (2.8) due to the existence of the Gauss tail function. To analyze K_0, we use the following polynomial approximation of $Q(t)$ for $t > 0$ [35]

$$Q(t) \approx \begin{cases} 0.5 - 0.1t(4.4-t) & \text{for} \quad 0 \le t \le 2.2, \\ 0.01 & \text{for} \quad 2.2 < t \le 2.6, \\ 0 & \text{for} \quad t > 2.6, \end{cases} \tag{2.9}$$

and find an approximated solution to (2.8). In this paper, we only consider the scenario where $a - b_s x > 0$, that is, $P_d^{(i)} < 0.5$ and a colluder's chance of being detected is smaller than 0.5. Assuming that $a - b_s x \le 2.2$, with the $Q(\cdot)$ function approximation in (2.9), we have

$$0.5 - 0.1(a-b_s x)(4.4-a+b_s x) \approx 1 - \frac{1}{1+\theta x}. \tag{2.10}$$

After rearranging both sides, (2.10) becomes a cubic equation of x

$$\begin{aligned} f_s(x) &= a_0 x^3 + a_1 x^2 + a_2 x + a_3 \approx 0, \\ \text{where} \quad a_0 &= b_s^2\theta, a_1 = -\left(2ab_s\theta - 4.4b_s\theta - b_s^2\right), \\ a_2 &= -\left(4.4a\theta - a^2\theta + 5\theta + 2ab_s - 4.4b_s\right), \text{ and } a_3 = -a(4.4-a) + 2. \end{aligned} \tag{2.11}$$

The cubic equation $f_s(x) = 0$ has three roots, which can be found using the Cardan's method [36]. Let $q = (3a_0a_2 - a_1^2)/(9a_0^2)$ and $r = (9a_0a_1a_2 - 27a_0^2a_3 - 2a_1^2)/(54a_0^3)$. Furthermore, let $s_1 = \sqrt[3]{r+\sqrt{q^3+r^2}}$ and $s_2 = \sqrt[3]{r-\sqrt{q^3+r^2}}$. Then, the three roots of

(2.11) are

$$\begin{aligned} x_1 &= s_1+s_2-\frac{a_1}{3a_0}, \\ x_2 &= -0.5(s_1+s_2)-\frac{a_1}{3a_0}+\frac{\sqrt{3}}{2}(s_1-s_2)j, \\ \text{and} \quad x_3 &= -0.5(s_1+s_2)-\frac{a_1}{3a_0}-\frac{\sqrt{3}}{2}(s_1-s_2)j, \end{aligned} \tag{2.12}$$

where j in (2.12) is the imaginary unit. Therefore, $K_{0,1} = 1/x_1$, $K_{0.2} = 1/x_2$ and $K_{0,3} = 1/x_3$ are the approximated roots of (2.7). We need to examine each of the three approximated roots to find K_0. Note that K_0 is a positive integer, and from Figure 2.1, $\frac{d\pi^{(i)}}{dK}|_{K=K_0} > 0$. Therefore, given $\{K_{0,1}, K_{0,2}, K_{0,3}\}$, we find the positive real root that satisfies $\frac{df_s(x)}{dx}|_{x=1/K} > 0$. We then use the selected root as an approximation of K_0. That is,

$$\hat{K}_0 = \lceil K_{0,i} \rceil \quad \text{where} \quad K_{0,i} \in \mathbb{R}^+ \quad \text{and} \quad \left.\frac{df_s(x)}{dx}\right|_{x=1/K_{0,i}} > 0. \tag{2.13}$$

To demonstrate the process to find the approximated K_0, we consider the example in Figure 2.1 where $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, $\theta = 50$, and $P_{fa} = 10^{-3}$. Plugging these numbers into (2.11), the cubic function is $f_s(x) = 5000000x^3 + 71848.38x^2 - 1015.41x + 0.9525$, whose three roots are $x_1 = 0.008011$, $x_2 = -0.023397$ and $x_3 = 0.001016$. Therefore, $K_{0,1} = 1/x_1 = 124.82$, $K_{0,2} = 1/x_2 = -42.73$ and $K_{0,3} = 1/x_3 = 983.95$. We first eliminate $K_{0,2}$ since it is a negative number. We then calculate the first order derivatives of the cubic function $f_s(x)$ at x_1 and x_3, which are $\left.\frac{df_s(x)}{dx}\right|_{x=x_1} = 1098.52$ and $\left.\frac{df_s(x)}{dx}\right|_{x=x_3} = -853.87$. Then, we can eliminate $K_{0,3}$ since $\left.\frac{df_s(x)}{dx}\right|_{x=x_3} < 0$, and select $\hat{K}_0 = \lceil K_{0,1} \rceil = 125$, which is very close to the true value $K_0 = 126$ we found using exhaustive search in the previous section.

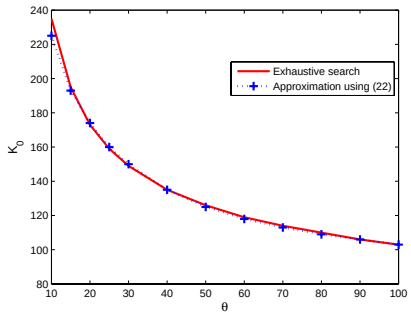
When θ takes different values, Figure 2.2 plots the approximated $\hat{K}_0$ calculated

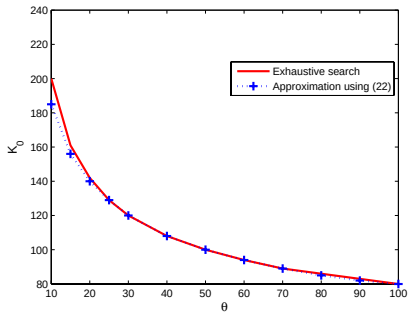
using (2.13) and the true values of K_0 that are found using exhaustive search. From Figure 2.2, K_0 is a decreasing function of θ. As an example, when $f_c = 1$, K_0 drops from 235 to 103 when θ increases from 10 to 100. In addition, if we compare Figure 2.2a with 2.2b, K_0 takes a smaller value if colluders generate a colluded copy of low resolution. For example, when $\theta = 50$, $K_0 = 126$ when $f_c = 1$ and $K_0 = 100$ when $f_c = 0.5$.

Furthermore, in the example in Figure 2.2, when $\theta \geq 20$, $\hat{K}_0$ in (2.13) gives a very good approximation of K_0. In the example in Figure 2.2a, when $f_c = 1$ and $\theta \geq 20$, the approximation error is no larger than 1, that is $|\hat{K}_0 - K_0| \leq 1$. In such cases, to improve the accuracy, given $\hat{K}_0$, we can verify whether it satisfies $\pi^{(i)}(\hat{K}_0 - 1) < 0$ and $\pi^{(i)}(\hat{K}_0) \geq 0$. If so, then $K_0 = \hat{K}_0$. Otherwise, we decrease $\hat{K}_0$ by one if $\pi^{(i)}(\hat{K}_0 - 1) \geq 0$, and we increase $\hat{K}_0$ by one if $\pi^{(i)}(\hat{K}_0) < 0$. Then, we verify again whether the new $\hat{K}_0$ satisfies $\pi^{(i)}(\hat{K}_0 - 1) < 0$ and $\pi^{(i)}(\hat{K}_0) \geq 0$. By doing so, we will find the exact solution of K_0.

When $\theta < 20$, there is a difference between the approximated $\hat{K}_0$ and the true value of K_0. In Figure 2.2a, when $f_c = 1$ and $\theta < 20$, the largest approximation error is 10, which happens when $\theta = 10$. This is because the above analysis of $\hat{K}_0$ uses the approximation $Q(t) \approx 0.5 - 0.1t(4.4 - t)$, which gives a good approximation of the Gauss tail function for $0 \leq t \leq 2.2$. When $a - b_s/K$ is larger than 2.2, the polynomial approximation of $Q(t)$ cannot be used. Therefore, as the last step, we need to verify that the selected root x_i in (2.13) satisfies $0 \leq a - b_s x_i \leq 2.2$. If not, numerical methods can be used to find the root of (2.8).

(a)





(b)

Figure 2.2: The approximation $\hat{K}_0$ in (2.13) and the true value of K_0. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, and $P_{fa} = 10^{-3}$. (a): the colluded copy includes both layers with $f_c = 1$. (b): $f_c = 0.5$.

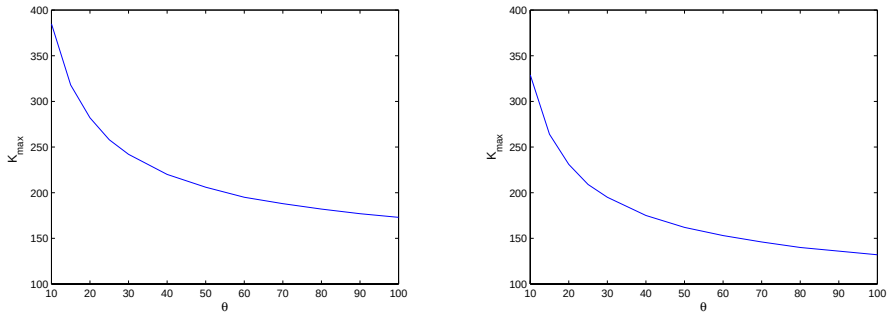
2.1.3.2 Analysis of K_{max}

Given N_b, N_e and θ, to find the maximal number of colluders K_{max}, we solve $\frac{\partial \pi^{(i)}}{\partial K} = 0$, or equivalently, find the root of $\frac{\partial \pi^{(i)}}{\partial x} = 0$, where $\pi^{(i)}$ is in (2.6) and $x = 1/K$ is the inverse of K. Same as in the previous section, we use $f_c = 1$ as an example, and the analysis for other values of f_c is similar and omitted. From (2.6), to find K_{max}, we solve

$$\begin{aligned} \frac{\partial \pi^{(i)}}{\partial x} &= -\frac{\partial P_d^{(i)}}{\partial x}(1+\theta x) + \left(1 - P_d^{(i)}\right)\theta = 0, \\ \text{where} \quad P_d^{(i)} &= Q(a - b_s x) \quad \text{and} \quad \frac{\partial P_d^{(i)}}{\partial x} = \frac{b_s}{\sqrt{2\pi}} \exp\left\{-\frac{(a-b_s x)^2}{2}\right\}. \end{aligned} \tag{2.14}$$

Due to the existence of both the Gauss tail function and the exponential function, it is difficult to find the analytical solution to (2.14), and we use numerical methods to solve (2.14) and find K_{max}.

Figure 2.3a and 2.3b show K_{max} as a function of θ when the colluded copy has high and low resolutions, respectively. The system setup is the same as in Figure 2.2. If we compare Figure 2.3a with 2.3b, K_{max} takes a smaller value when the colluded copy has a lower resolution. For example, in Figure 2.3, with $\theta = 50$, $K_{max} = 206$ when the colluded



(a) (b)

Figure 2.3: K_{max} versus θ. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, and $P_{fa} = 10^{-3}$. (a): all colluders receive fingerprinted copies of high resolution and $f_c = 1$. (b): all attackers receive the base layer only and $f_c = 0.5$.
copy has high resolution, and $K_{max} = 162$ when $f_c = 0.5$.

Furthermore, K_{max} is a decreasing function of θ. For example, when $f_c = 1$, $K_{max} = 385$ when $\theta = 10$ and $K_{max} = 173$ when $\theta = 100$. This is because, when θ takes a smaller value and when attackers emphasize more on risk minimization, they prefer to collude with more people to lower their risk. Mathematically, it can be proved as follows. After rearranging both sides of (2.14), we have

$$\frac{1 - P_d^{(i)} - \frac{\partial P_d^{(i)}}{\partial x}}{\frac{\partial P_d^{(i)}}{\partial x}} = [1 - Q(a - b_s x)] \frac{\sqrt{2\pi}}{b_s} \exp\left\{\frac{(a - b_s x)^2}{2}\right\} - 1 = \theta^{-1}. \tag{2.15}$$

Assume that $\theta_1 \geq \theta_2$, and x_1 and x_2 are the solutions to (2.15) when $\theta = \theta_1$ and $\theta = \theta_2$, respectively. Note that the left hand side of (2.15) is a constant of θ. Consequently, $[1 - Q(a - b_s x_1)] \frac{\sqrt{2\pi}}{b_s} \exp\left\{\frac{(a-b_s x_1)^2}{2}\right\} - 1 = \theta_1^{-1} \leq \theta_2^{-1} = [1 - Q(a - b_s x_2)] \frac{\sqrt{2\pi}}{b_s} \exp\left\{\frac{(a-b_s x_2)^2}{2}\right\} - 1$. In this paper, we consider the scenario where colluders' probability of being detected is smaller than 0.5, that is, $P_d^{(i)} = Q(a - b_s x) < 0.5$ and $a - b_s x > 0$. In such a scenario, both $[1 - Q(a - b_s x)]$ and $\exp\left\{\frac{(a-b_s x)^2}{2}\right\}$ are decreasing functions of x, thus $x_1 \geq x_2$ and $K_{max}^1 = 1/x_1 \leq K_{max}^2 = 1/x_2$.

To summarize, when all attackers receive copies of the same resolution, they collude

with each other if and only if the total number of colluders is larger than K_0 and when all attackers receive positive payoffs. In addition, an attacker should try to find a total of K_{max} colluders if possible to maximize his/her payoff.

2.1.4 Necessary Condition for Multi-Resolution Collusion

In this subsection, we consider the scenario where colluders receive fingerprinted copies of different resolutions and analyze when attackers will collude with each other.

One possible outcome of the bargaining between SC^b and SC^{be} is that they do not reach an agreement. In such a scenario, attackers only collude with their fellow attackers in the same subgroup, and SC^b and SC^{be} do not cooperate with each other. Given N_b, N_e, K^b and K^{be}, if an attacker in SC^b colludes with those in SC^b only, his or her utility is

$$\begin{aligned} \pi^b_{nc} &= -P^b_{d,nc} + \left(1 - P^b_{d,nc}\right) R^b_{nc} \\ \text{where} \quad P^b_{d,nc} &= Q\left(\frac{h}{\sigma_n} - \frac{\sqrt{N_b}\sigma_w}{K^b\sigma_n}\right) = Q\left(a - b_b/K^b\right) \quad \text{and} \quad R^b_{nc} = \frac{\theta f_b}{K^b}. \end{aligned} \tag{2.16}$$

In (2.16), $a = h/\sigma_n$ and $b_b = \sqrt{N_b}\sigma_w/\sigma_n$. Similarly, if an attacker in SC^{be} colludes with those in SC^{be} only, his or her payoff is

$$\begin{aligned} \pi^{be}_{nc} &= -P^{be}_{d,nc} + \left(1 - P^{be}_{d,nc}\right) R^{be}_{nc} \\ \text{where} \quad P^{be}_{d,nc} &= Q\left(\frac{h}{\sigma_n} - \frac{\sqrt{N_b + N_e}\sigma_w}{K^{be}\sigma_n}\right) = Q\left(a - b_s/K^{be}\right) \quad \text{and} \quad R^{be}_{nc} = \frac{\theta}{K^{be}}. \end{aligned} \tag{2.17}$$

In (2.17), $b_s = \sqrt{N_b + N_e}\sigma_w/\sigma_n$.

If SC^b and SC^{be} collaborate with each other and select the collusion parameter β, for an attacker $i \in SC^b$, his or her utility is

$$\pi^b = -P^b_{d,c} + \left(1 - P^b_{d,c}\right) R^b_c,$$

$$\text{where} \quad P^b_{d,c} = Q\left(\frac{h}{\sigma_n} - \frac{\beta\sqrt{N_b}}{K^b}\cdot\frac{\sigma_w}{\sigma_n}\right) = Q\left(a - \beta\frac{b_b}{K^b}\right) \quad \text{and} \quad R^b_c = \frac{(f^c)^\gamma\theta}{K^b(f^c)^\gamma + K^{be}} \tag{2.18}$$

Similarly, for $0 \le \beta \le \beta^+$, an attacker $i \in SC^{be}$'s payoff is

$$\begin{aligned}
\pi^{be} &= -P^{be}_{d,c} + \left(1 - P^{be}_{d,c}\right) R^{be}_c, \text{ where} \\
P^{be}_{d,c} &= Q\left(\frac{h}{\sigma_n} - \frac{(1-\beta)N_b + N_e}{K^{be}\sqrt{N_b+N_e}}\cdot\frac{\sigma_w}{\sigma_n}\right) = Q\left(a - \frac{b_s}{K^{be}} + \beta\frac{b_{be}}{K^{be}}\right) \text{ and } R^{be}_c = \frac{\theta}{K^b(f^c)^\gamma + K^{be}}
\end{aligned} \tag{2.19}$$

In (2.19), $b_{be} = \frac{N_b\sigma_w}{\sqrt{N_b+N_e}\sigma_n}$.

Therefore, among all the possible solutions $\left\{(\pi^b, \pi^{be})\right\}$ in the feasible set $\mathbb{S}$, colluders are only interested in those in $\mathbb{S}_p = \left\{(\pi^b, \pi^{be}) \in \mathbb{S} : \pi^b \ge \underline{\pi}^b = \max(0, \pi^b_{nc}), \pi^{be} \ge \underline{\pi}^{be} = \max(0, \pi^b_{nc}), 0\right.$ where cooperation helps both SC^b and SC^{be} increase their payoffs.

- From (2.18), $P^b_{d,c}$ is an increasing function of β and, therefore, π^b is a decreasing function of β. Let $\bar{\beta}$ be the β that makes π^b equal to $\underline{\pi}^b$, that is, $\pi^b(\bar{\beta}) = \underline{\pi}^b$. Then, the constraint $\pi^b \ge \underline{\pi}^b$ is equivalent to let $\beta \le \bar{\beta}$.

- Similarly, from (2.19), $P^{(be)}_{d,c}$ is a decreasing function of β, and thus π^{be} is an increasing function of β. Let $\underline{\beta}$ be the β that makes π^{be} equal to $\underline{\pi}^{be}$, that is, $\pi^{be}(\underline{\beta}) = \underline{\pi}^{be}$. Therefore, the constraint $\pi^{be} \ge \underline{\pi}^{be}$ is equivalent to select $\beta \ge \underline{\beta}$.

- Furthermore, if we compare (2.17) and (2.19), $P^{be}_{d,c} = P^{be}_{d,nc}$ when $\beta = 0$. That is, if colluders select $\beta = 0$, then collaborating with SC^b does not help SC^{be} further reduce their risk of being detected. Meanwhile, $R^{be}_c < R^{be}_{nc}$ and colluders in SC^{be} receive less reward if they cooperate with SC^b. Consequently, $\pi^{be}(0) < \pi^{be}_{nc} \le \underline{\pi}^{be} = \max\left(\pi^{be}_{nc}, 0\right) = \pi^{be}(\underline{\beta})$. Thus, $\underline{\beta} > 0$ since π^{be} is an increasing function of β.

From the above analysis, we can rewrite $\mathbb{S}_p$ as $\mathbb{S}_p = \left\{(\pi^b, \pi^{be}) \in \mathbb{S} : \underline{\beta} \le \beta \le \min(\bar{\beta}, \beta^+)\right\}$. When attackers receive fingerprinted copies of different resolutions, the two subgroups of

colluders SC^b and SC^{be} will collude with each other if and only if there exists at least one β such that $\underline{\beta} \leq \beta \leq \min(\bar{\beta}, \beta^+)$, or equivalently, when $\mathbb{S}_p$ is not empty.

2.1.4.1 Lower and Upper Bounds of β

To further understand under what conditions SC^b and SC^{be} will cooperate with each other, we will first analyze $\underline{\beta}$ and $\bar{\beta}$.

From the previous discussion, given N_b, N_e, K^b and K^{be}, colluders should select β such that

$$\pi^b(\beta) = -P^b_{d,c}(\beta) + \left[1 - P^b_{d,c}(\beta)\right] R^b_c \geq \underline{\pi}^b = \max\left(0, \pi^b_{nc}\right), \tag{2.20}$$

where $P^b_{d,c}(\beta)$ and R^b_c are in (2.18). Consequently, we have

$$P^b_{d,c}(\beta) = Q\left(a - \frac{\beta b_b}{K^b}\right) \leq \frac{R^b_c - \underline{\pi}^b}{R^b_c + 1}. \tag{2.21}$$

Since $Q(x)$ is a decreasing function of x, therefore, we have

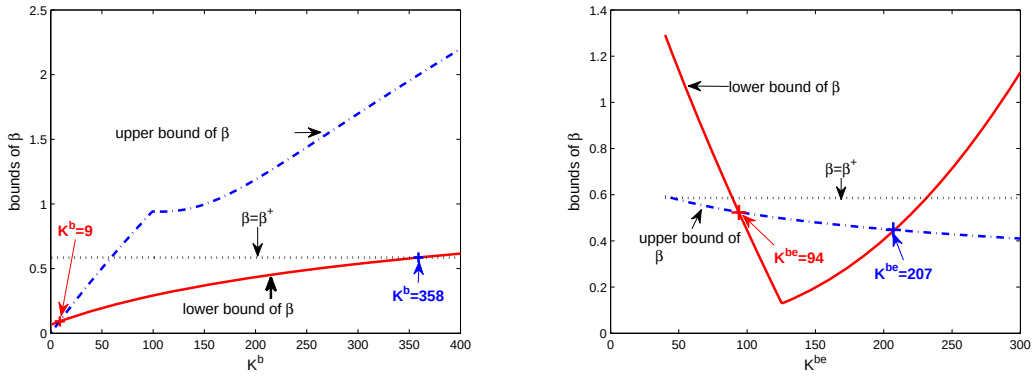
$$\begin{aligned} a - \beta\frac{b_b}{K^b} &\geq Q^{-1}\left(\frac{R^b_c - \underline{\pi}^b}{R^b_c + 1}\right), \\ \text{or equivalently,} \quad \beta &\leq \bar{\beta} = \left[a - Q^{-1}\left(\frac{R^b_c - \underline{\pi}^b}{R^b_c + 1}\right)\right] K^b / b_b. \end{aligned} \tag{2.22}$$

Similarly, given N_b, N_e, K^b and K^{be}, colluders should select β such that

$$\pi^{be}(\beta) = -P^{be}_{d,c}(\beta) + \left[1 - P^{be}_{d,c}(\beta)\right] R^{be}_c \geq \underline{\pi}^{be} = \max\left(0, \pi^{be}_{nc}\right), \tag{2.23}$$

where $P^{be}_{d,c}(\beta)$ and R^{be}_c are in (2.19). Therefore, we have

$$\begin{aligned} P^{be}_{d,c}(\beta) &= Q\left(a - \frac{\sqrt{N_b + N_e}\sigma_w}{K^{be}\sigma_n} + \frac{\beta b_{be}}{K^{be}}\right) \geq \frac{R^{be}_c - \underline{\pi}^{be}}{R^{be}_c + 1}, \\ \text{or equivalently,} \quad \beta \geq \underline{\beta} &= \left[Q^{-1}\left(\frac{R^{be}_c - \underline{\pi}^{be}}{R^{be}_c + 1}\right) - a + \frac{\sqrt{N_b + N_e}\sigma_w}{K^{be}\sigma_n}\right] K^{be} / b_{be} \\ &= \frac{N_b + N_e}{N_b} + \left[Q^{-1}\left(\frac{R^{be}_c - \underline{\pi}^{be}}{R^{be}_c + 1}\right) - a\right] K^{be} / b_{be}. \end{aligned} \tag{2.24}$$



(a) (b)

Figure 2.4: The upper bound and lower bound of β.(a): $K^{be} = 120$, (b): $K^b = 50$.

Figure 2.1.4.1 shows examples of $\underline{\beta}$ and $\bar{\beta}$. $K^{be} = 120$ in Figure 2.1.4.1a, and $K^b = 50$ in Figure 2.1.4.1b. From Figure 2.1.4.1(a), $\bar{\beta} < \underline{\beta}$ when $K^b < 9$, and $\underline{\beta} > \beta^+$ when $K^b > 358$. Therefore, in this example where K^{be} is fixed as 120, $\mathbb{S}_p \neq \emptyset$ if and only if $9 \leq K^b \leq 358$. Similarly, from Figure 2.1.4.1(b), $\underline{\beta} > \bar{\beta}$ if $K^{be} < 94$ or $K^{be} > 207$. Thus, when $K^b = 50$ is fixed, SC^{be} and SC^b will collude with each other if and only if $94 \leq K^{be} \leq 207$.

2.1.4.2 Analysis of $\mathbb{K}_p$

From Figure 2.1.4.1, given N_b, N_e and θ, for some pairs of (K^b, K^{be}), $\mathbb{S}_p$ may be empty and thus, SC^b and SC^{be} will not cooperate. Define $\mathbb{K}_p \triangleq \{(K^b, K^{be}) : \mathbb{S}_p \neq \emptyset\}$ as the set including all pairs of (K^b, K^{be}) where $\mathbb{S}_p$ is not empty and where SC^b and SC^{be} will collude with each other.

Given N_b, N_e and θ, SC^b and SC^{be} will collude with each other if and only if $\mathbb{S}_p \neq \emptyset$, that is, when $\underline{\beta} \leq \beta^+$ and $\underline{\beta} \leq \bar{\beta}$. Since π^{be} in (2.19) is an increasing function of β, if

$\underline{\beta} \leq \beta^+$, then we have

$$\begin{aligned} \pi^{be}(\beta^+) &= -P^{be}_{d,c}(\beta^+) + \left[1 - P^{be}_{d,c}(\beta^+)\right] R^{be}_c \geq \pi^{be}(\underline{\beta}) = \underline{\pi}^{be}, \\ \text{or equivalently,} \quad R^{be}_c &= \frac{\theta}{K^b (f_c)^\gamma + K^{be}} \geq \frac{\underline{\pi}^{be} + P^{be}_{d,c}(\beta^+)}{1 - P^{be}_{d,c}(\beta^+)}. \end{aligned} \tag{2.25}$$

Consequently, to ensure $\underline{\beta} \leq \beta^+$, (K^b, K^{be}) must satisfy

$$K^b \leq K^{b'}(K^{be}) \triangleq \frac{\theta\left(1 - P^{be}_{d,c}(\beta^+)\right)}{\left(\underline{\pi}^{be} + P^{be}_{d,c}(\beta^+)\right)(f_b)^\gamma} - \frac{K^{be}}{(f_b)^\gamma}. \tag{2.26}$$

From (2.22) and (2.24), to ensure $\underline{\beta} \leq \bar{\beta}$, (K^b, K^{be}) must satisfy

$$\underline{\beta} = \frac{N_b + N_e}{N_b} + \left[Q^{-1}\left(\frac{R^{be}_c - \underline{\pi}^{be}}{R^{be}_c + 1}\right) - a\right] \frac{K^{be}}{b_{be}} \leq \bar{\beta} = \left[a - Q^{-1}\left(\frac{R^b_c - \underline{\pi}^b}{R^b_c + 1}\right)\right] \frac{K^b}{b_b}. \tag{2.27}$$

Combining (2.26) and (2.27), we have

$$\begin{aligned} \mathbb{K}_p = \Bigg\{ (K^b, K^{be}) \quad : \quad & K^b \leq \frac{\theta\left(1 - P^{be}_{d,c}(\beta^+)\right)}{\left(\underline{\pi}^{be} + P^{be}_{d,c}(\beta^+)\right)(f_b)^\gamma} - \frac{K^{be}}{(f_b)^\gamma}, \\ & \frac{N_b + N_e}{N_b} + \left[Q^{-1}\left(\frac{R^{be}_c - \underline{\pi}^{be}}{R^{be}_c + 1}\right) - a\right] \frac{K^{be}}{b_{be}} \leq \left[a - Q^{-1}\left(\frac{R^b_c - \underline{\pi}^b}{R^b_c + 1}\right)\right] \frac{K^b}{b_b} \Bigg\} \end{aligned} \tag{2.28}$$

The shaded area in Figure 2.5 shows an example of $\mathbb{K}_p$. At point 'A' in Figure 2.5, when $K^{be} < 91$, no matter which value K^b takes, $\mathbb{S}_p$ is always empty and attackers will not collude with each other. Similarly, when $K^{be} > 226$ (point 'B' in Figure 2.5), no matter how many attackers are in SC^b and how they select β, cooperation between SC^b and SC^{be} cannot improve all colluders' payoffs. Furthermore, when $K^b > 431$ (point 'C' in Figure 2.5), SC^b and SC^{be} will not collude with each other. To quantify the above boundary points of $\mathbb{K}_p$, we define

$$\begin{aligned} \underline{K}^{be} &\triangleq \min\left\{K^{be} : \exists K^b \text{ s.t. } (K^b, K^{be}) \in \mathbb{K}_p\right\}, \\ \bar{K}^{be} &\triangleq \max\left\{K^{be} : \exists K^b \text{ s.t. } (K^b, K^{be}) \in \mathbb{K}_p\right\}, \end{aligned}$$

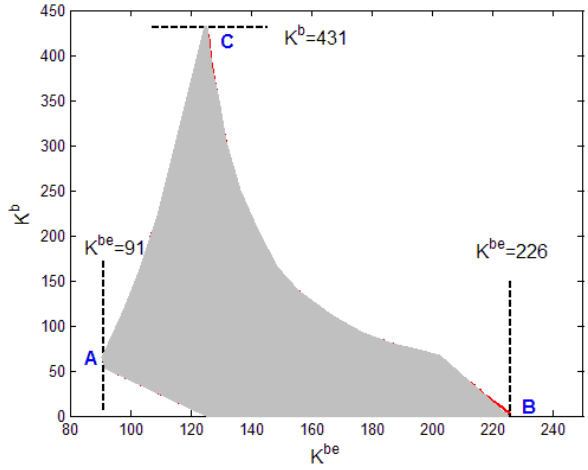


Figure 2.5: An example of $\mathbb{K}_p$. $N_b = N_e = 50000$ and $\theta = 50$. The probability of falsely accusing an innocent user is $P_{fa} = 10^{-3}$.

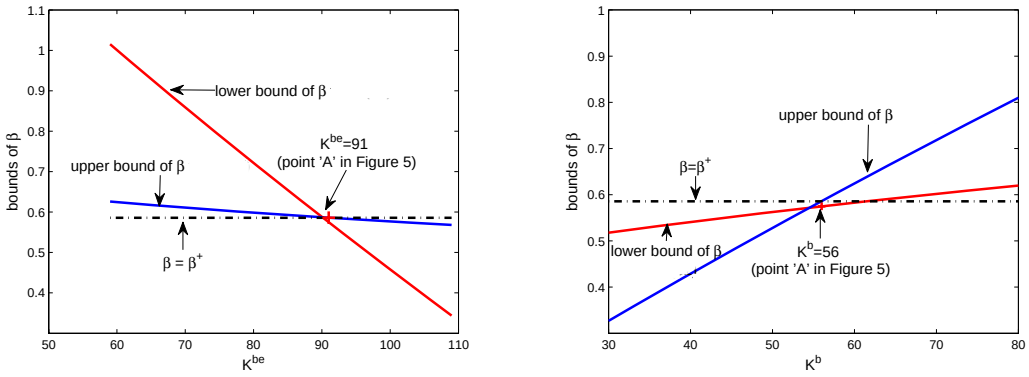
$$\text{and} \quad \bar{K}^b \triangleq \max\left\{K^b : \exists K^{be} \text{ s.t. } (K^b, K^{be}) \in \mathbb{K}_p\right\}. \tag{2.29}$$

In the example in Figure 2.5, $\underline{K}^{be} = 91$, $\bar{K}^{be} = 226$ and $\bar{K}^b = 431$.

2.1.4.3 Analysis of the Bounds of Number of Colluders

From Figure 2.5 and (2.29), if $K^b > \bar{K}^b$, $K^{be} < \underline{K}^{be}$, or $K^{be} > \bar{K}^{be}$, then it is impossible to find a β that increases all colluders' payoffs, and SC^b and SC^{be} will not cooperate with each other. Therefore, during collusion, as a preliminary step, colluders should first check that $K^b \leq \bar{K}^b$ and $\underline{K}^{be} \leq K^{be} \leq \bar{K}^{be}$. Then, they should ensure that (K^b, K^{be}) is in the set $\mathbb{K}_p$ defined in (2.29), and guarantee that there exists at least one β that increases both SC^b and SC^{be}'s payoffs. In the following section, we will analyze the boundary points of $\mathbb{K}_p$ ($\underline{K}^{be}$, $\bar{K}^{be}$ and $\bar{K}^b$) in details.

$\underline{K}^{be}$ Using exhaustive search, we find that at point 'A' in Figure 2.5, $K^b = 56$ and $K^{be} = 91$. Since $K^b < K_0(f_c = f_b) = 100$ and $K^{be} < K_0(f_c = 1) = 126$, we have $\pi^b_{nc} < 0$, $\pi^{be}_{nc} < 0$,



(a) (b)

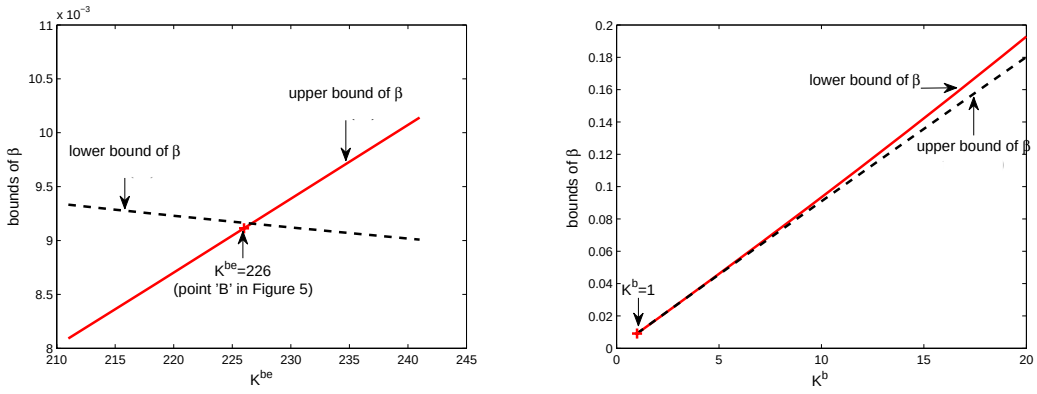
Figure 2.6: Upper and lower bound of β at point 'A' in Figure 2.5. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, $\gamma = 1/3$, $P_{fa} = 10^{-3}$, and $\theta = 50$. (a): $K^b = 56$, (b): $K^{be} = 91$.

and $\underline{\pi}^{be} = \underline{\pi}^{b} = 0$. To have a better understanding of $\underline{K}^{be}$, Figure 2.5 plots $\underline{\beta}$ and $\bar{\beta}$ around the point $(K^b = 56, K^{be} = 91)$. As can be seen from Figure 2.5, at point 'A', $\underline{\beta} = \bar{\beta} = \beta^+$, and $\mathbb{S}_p$ has only one item, which is $\mathbb{S}_p = \{(\pi^b, \pi^{be}) : \beta = \beta^+\}$. Since $\pi^{be} = \underline{\pi}^{be}$ when $\beta = \underline{\beta}$, and $\pi^b = \underline{\pi}^b$ when $\beta = \bar{\beta}$. Therefore, at the boundary point 'A', (K^b, K^{be}) satisfies

$$\begin{cases} \pi^b(K^b, K^{be}, \beta^+) = \underline{\pi}^b = 0, \\ \pi^{be}(K^b, K^{be}, \beta^+) = \underline{\pi}^{be} = 0. \end{cases} \tag{2.30}$$

To find $\underline{K}^b$, we should first find the solution (K^b, K^{be}) to the above equation (2.30) and then select $\underline{K}^{be} = \lceil K^{be} \rceil$. Using Figure 2.5 as an example, given the parameters $N_b = N_e = 50000$, $\gamma = 1/3$, $\theta = 50$ and $P_{fa} = 10^{-3}$, we first find the solution to (2.30), which is $(K^b = 55.88, K^{be} = 90.15)$. We then calculate $\underline{K}^{be} = \lceil K^{be} \rceil = 91$, which is consistent with the result we find using exhaustive search.

$\bar{K}^{be}$ To analyze $\bar{K}^{be}$, using exhaustive search, we find that at point 'B' in Figure 2.5, $K^b = 1 < K_0(f_c = f_b) = 100$ and $K^{be} = 226 > K_0(f_c = 1) = 126$. Therefore, at this



(a) (b)

Figure 2.7: Upper and lower bound of β at point 'B' in Figure 2.5. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, $\gamma = 1/3$, $P_{fa} = 10^{-3}$, and $\theta = 50$. (a): $K^b = 1$. (b): $K^{be} = 226$.

point, $\underline{\pi}^b = 0$ and $\underline{\pi}^{be} = \pi_{nc}^{be} > 0$. As shown in Figure 2.7, at this point, $\underline{\beta} = \bar{\beta}$, and $\mathbb{S}_p = \left\{(\pi^b, \pi^{be}) : \beta = \underline{\beta} = \bar{\beta}\right\}$ has only one entry. Also, from Figure 2.7b, when $K^{be} = \bar{K}^{be}$, if SC^b has more than one attacker (that is, $K^b \geq 2$), there is no β that can improve both SC^{be} and SC^b's payoffs. Therefore, point 'B' corresponds to the scenario where $K^b = 1$, $K^{be} = \bar{K}^{be}$ and $\underline{\beta} = \bar{\beta}$. Thus, from (2.22) and (2.24), to find $\bar{K}^{be}$, we first solve

$$\underline{\beta} = \frac{N_b + N_e}{N_b} + \left[Q^{-1}\left(\frac{R_c^{be} - \pi_{nc}^{be}}{R_c^{be} + 1}\right) - a\right]\frac{K^{be}}{b_{be}} = \bar{\beta} = \left[a - Q^{-1}\left(\frac{R_c^b}{R_c^b + 1}\right)\right]\frac{1}{b_b}, \quad (2.31)$$

and then let $\bar{K}^{be} = \lfloor K^{be} \rfloor$. In (2.31), $R_c^{be} = \theta/[(f_b)^\gamma + K^{be}]$, $R^b = \theta(f_b)^\gamma/[(f_b)^\gamma + K^{be}]$, and π_{nc}^{be} is in (2.17). As an example, given the system setup in Figure 2.5, the solution to (2.31) is $K^{be} = 226.64$ and thus $\bar{K}^{be} = \lfloor 226.64 \rfloor = 226$. It is consistent with the result we found using exhaustive search.

$\bar{K}^b$ At point 'C' in Figure 2.5, we find $K^b = 431$ and $K^{be} = 125$ using exhaustive search and $\underline{\beta} = \beta^+$, as shown in Figure 2.8. From the analysis in Section 2.1.4.2, for a given K^{be}, to satisfy the constraint $\underline{\beta} \leq \beta^+$, it is required that $K^b \leq K^{b'}$, where $K^{b'}$ is defined

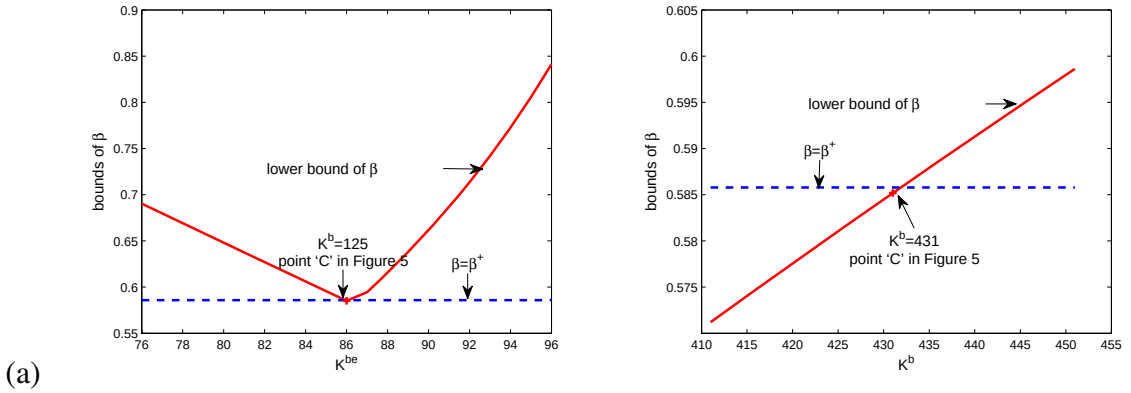


Figure 2.8: Upper and lower bound of β at point 'C' in Figure 2.5. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, $\gamma = 1/3$, $P_{fa} = 10^{-3}$, and $\theta = 50$. (a): $K^b = 125$. (b): $K^{be} = 431$.

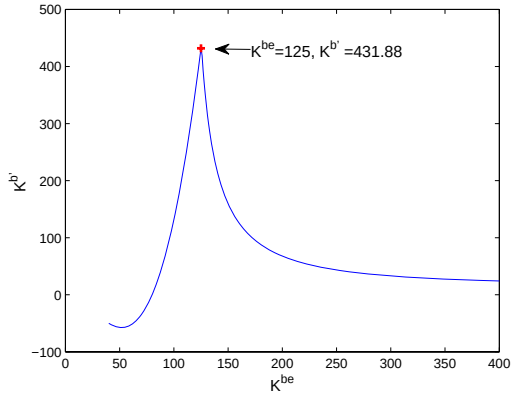


Figure 2.9: $K^{b'}$ versus K^{be}. $N_b = N_e = 50000$, $\sigma_w^2 = \sigma_n^2 = 1$, $\gamma = 1/3$, $P_{fa} = 10^{-3}$, and $\theta = 50$.

in (2.26). Therefore, we have $\bar{K}^b = \lfloor \max_{K^{be}} K^{b'} \rfloor$. Using the system setup in Figure 2.5 as an example, Figure 2.9 plots $K^{b'}$ versus K^{be}, and $K^{b'}$ achieves a maximum of 431.88 when $K^{be} = 125$. Consequently, $\bar{K}^b = \lfloor 431.88 \rfloor = 431$, which agrees with the result we found using exhaustive search.

To summarize, given N_b, N_e, and other parameters including θ and γ, in order to ensure that cooperation can help both SC^b and SC^{be} improve their payoffs, colluders should first follow the analysis in Section 2.1.4.3 and ensure that $\underline{K}^{be} \leq K^{be} \leq \bar{K}^{be}$ and $K^b \leq \bar{K}^b$.

Then, attackers should further check whether (K^b, K^{be}) satisfies the constraints in (2.28) and whether $\mathbb{S}_p$ is not empty. If $(K^b, K^{be}) \in \mathbb{K}_p$, colluders should use (2.22) and (2.24) to calculate $\bar{\beta}$ and $\underline{\beta}$, respectively, and find $\mathbb{S}_p = \left\{(\pi^b, \pi^{be}) : \underline{\beta} \leq \beta \leq \min(\bar{\beta}, \beta^+)\right\}$. By doing so, no matter which pair (π^b, π^{be}) that colluders select in $\mathbb{S}_p$, it is a Pareto optimal solution and all colluders increase their payoffs by cooperating with each other.

2.2 Fair Bargaining Solutions in Colluders Social Network

Based on the discussion in Section 2.1, we know the situations that the attackers will mount multi-user collusion. The next question to ask is which collusion strategy is fair and all colluders will agree with it.

2.2.1 Fairness Criteria

Depending on the definition of fairness and the objectives of collusion, colluders select different collusion strategies and aim to reach agreement under different fairness criteria. In this section, we demonstrate the behavior analysis of colluder social network by four commonly used fairness criteria during bargaining.

Absolute Fairness: The most straight-forward fairness criteria is the absolute fairness, which means the utility of every user in the colluder social network is equal, where

$$\pi_{Absolute} = \pi^{(i)} = \pi^{(j)} \quad \forall i, j \in SC. \tag{2.32}$$

Moreover, since we have assumed colluders who receive the same quality copies have

equal utility, (2.32) can be simplified to

$$\pi_{Absolute} = \pi^b = \pi^{b,e}. \tag{2.33}$$

Properties: Although absolute fairness solution is the simplest and seemed most fair criteria, depending on the parameter $L^{(i)}$, $|SC^b|$, and $|SC^{b,e}|$, absolute fairness solution does not always exist. Therefore, other fairness criteria have to be taken into account.

MaxMin Fairness: To guarantee the utility of every one who participate the colluder social network, colluders can reach the agreement that the collusion parameters maximize the minimum utility over all the users in the social network, that is,

$$\pi_{maxmin} = \max_{\beta} \min_{i} \{\pi^{(i)} : i \in SC\}, \tag{2.34}$$

which can also be simplified to

$$\pi_{maxmin} = \max_{\beta} \min\{\pi^b, \pi^{b,e}\}. \tag{2.35}$$

Max Sum Fairness: Under some circumstances, all the users in the colluder social network have the same goal so that they are willing to maximize the total utility over the whole social network. Mathematically, the max-sum fairness solution can be formulated as follows:

$$\pi_{maxsum} = \max_{\beta} \sum_{i \in SC} \pi^{(i)}. \tag{2.36}$$

Properties: Max sum solution has a desired property that if it is feasible, it is Pareto-Optimal. Pareto optimality means no player can increase his/her payoff without decreasing others'. In a bargaining situation, players would always like to settle at a Pareto-optimal outcome. This is because if they select a point that is not Pareto-optimal,

then there exists another solution where at least one player can have larger payoff without hurting the interest of the other players.

Proof: If $\pi_{maxsum} = K^b \pi^b_{maxsum} + K^{b,e} \pi^{b,e}_{maxsum}$ is feasible but not Pareto-Optimal, then there exists $(\pi^b_{maxsum}, \pi^{b,e'})$ or $(\pi^{b'}, \pi^{b,e}_{maxsum})$ in feasible set where $\pi^{b'} > \pi^b_{maxsum}, \pi^{b,e'} > \pi^{b,e}_{maxsum}$ by the definition of Pareto-Optimal. Thus there exists a feasible $\pi' > \pi_{maxsum}$, which contradict the definition in (2.36).

Nash-Bargaining Solution: Nash-Bargaining solution, which is also Pareto-Optimal [7], is a famous bargaining solution in game theory, in which the basic idea being proportional fairness. Definition of general Nash-Bargaining solution is as follows:

$$\begin{aligned} g(\pi^b, \pi^{be}) &= \left(\pi^b - \pi^{b*}\right)^{B_b} \left(\pi^{be} - \pi^{be*}\right)^{B_{b,e}}, \\ \text{where} \quad \pi^{b*} &= \min_{\beta}\{\pi^b\} \quad , \quad \pi^{be*} = \min_{\beta}\{\pi^{be}\}, \end{aligned} \tag{2.37}$$

and $B_b, B_{b,e}$ are the bargaining power of SC^b, $SC^{b,e}$, respectively. When $B_b = B_{b,e} = 1$, Nash-Bargaining solution divides the additional utility between the two players in a ratio that is equal to the rate at which this utility can be transferred. If $B_b \neq B_{b,e}$, then the bargaining solution deviate from the proportional fairness solution and favors the player with higher bargain power.

2.2.2 Case Study and Simulation Results

In this section, we take two different utility functions as examples to illustrate the human behavior dynamics of colluder social network.

2.2.2.1 Case one: Reward is not proportional to risk

To have a clear picture of the agreement that the four fairness criteria will achieve, we first use a simple utility function as follows:

$$\pi^{(i)} = -P_d^{(i)} * L^{(i)} + \left(1 - P_d^{(i)}\right) \frac{F^c/F^{max}}{K/M}, \tag{2.38}$$

which is a special case of (2.5) where γ=0 and $D(P_d^{(i)})$=1, meaning the reward of redistributing the colluded copy is equally distributed to all the colluders. Therefore, the utility functions of the two players, sc^b and $sc^{b,e}$ can be written as

$$\pi^b = (R - L^b)P_d^b + R \qquad \text{and } \pi^{b,e} = (R - L^{b,e})P_d^{b,e} + R, \quad \text{where } R = \frac{F^c/F^{max}}{K/M}. \tag{2.39}$$

In the following, we will analyze the feasible region, the Pareto-Optimal set, and the bargaining solutions based on different fairness criteria. Moreover, since the loss term $L^{(i)}$ is a private information and is declared by the colluders themselves, we will also discuss which value of $L^{(i)}$ would be optimal for each player.

1. **Feasible Set**

   Given a N-person general-sum game, there is a certain subset S of R_N, called the feasible set. It is feasible in the sense that, given any $(\pi_1, \pi_2, ..., \pi_N) \in S$, it is possible for the players $u_1, u_2, ..., u_N$, acting together, to obtain the utilities $\pi_1, \pi_2, ..., \pi_N$, respectively.

   The self-probing fingerprint detector has approximately the same performance as the optimal detector. Therefore, colluders should consider the worse-case scenario

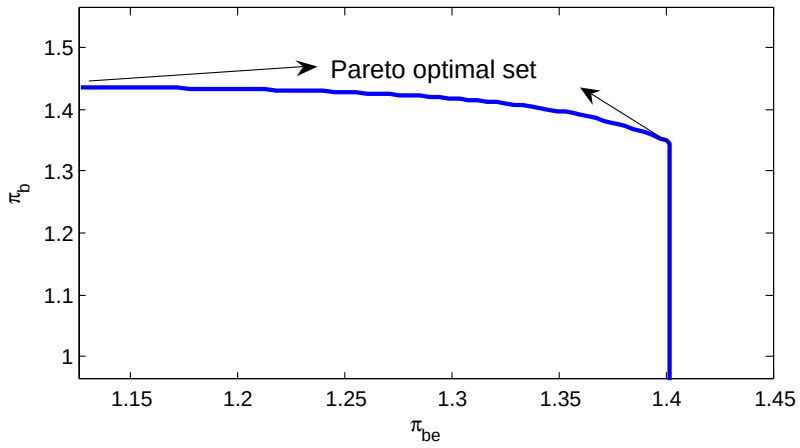


Figure 2.10: An example of Pareto-optimal set for the bargaining problem in case one

and assume that the fingerprint detector can always select the detection statistics with the largest mean. Following the analysis in [37], under the assumption that the detection noise are i.i.d. Gaussian $\mathcal{N}(0,\sigma_n^2)$,

$$\begin{aligned} P_d^{(i)} &= Q\left(\frac{h-\mu_{max}^{(i)}}{\sigma_n}\right), \\ \mu_{max}^{(i)} &= \mu_b \stackrel{\triangle}{=} \frac{\beta\sqrt{N_b}}{K^b}\sigma_w \quad \text{for } i \in SC^b, \\ \text{and} \quad \mu_{max}^{(i)} &= \mu_{b,e} \stackrel{\triangle}{=} \max\{\mu_{b,e}^b, \mu_{b,e}^e, \mu_{b,e}^c\} \quad \text{for } i \in SC^{b,e}, \\ \text{where} \quad \mu_{b,e}^b &= \frac{(1-\beta)\sqrt{N_b}}{K^{b,e}}\sigma_w,\ \mu_{b,e}^e = \frac{\sqrt{N_e}}{K^{b,e}}, \\ \text{and} \quad \mu_{b,e}^c &= \frac{(1-\beta)N_b+N_e}{K^{b,e}\sqrt{N_b+N_e}}\sigma_w. \end{aligned} \tag{2.40}$$

$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{\frac{-t^2}{2}}dt$ is the Gaussian tail function.

From (2.40), for a given β, μ_b is fixed while $\mu_{b,e1}$ may take three different values. To find the feasible set of the game, we need to find the relationship between β and $\mu_{b,e}$ first.

- **Scenario 1** $\mu_{b,e} = \mu_{b,e}^b$: $\mu_{b,e} = \mu_{b,e}^b$ if and only if $\mu_{b,e}^b \geq \mu_{b,e}^e$, and $\mu_{b,e}^b \geq \mu_{b,e}^c$.

So, from (2.40),

$$(1-\beta) \geq \max\left\{\frac{\sqrt{N_e}}{\sqrt{N_b}}, \frac{N_e}{\sqrt{N_b}(\sqrt{N_b+N_e}-\sqrt{N_b})}\right\} \tag{2.41}$$

Note that $\sqrt{N_b}+\sqrt{N_e} \geq \sqrt{N_b+N_e}$. So the second upper bound in (2.41) is always larger or equal to the first one. Thus, we have

$$\mu_{b,e} = \mu^b_{b,e} \Leftrightarrow 0 \leq \beta \leq 1 - \frac{N_e}{\sqrt{N_b}(\sqrt{N_b+N_e}-\sqrt{N_b})}. \tag{2.42}$$

The two terms of the upper bound in (2.42) can be combined as

$$\frac{\sqrt{N_b}\sqrt{N_b+N_e}-N_b-N_e}{\sqrt{N_b}(\sqrt{N_b+N_e}-\sqrt{N_b})} = \frac{\sqrt{N_b+N_e}(\sqrt{N_e}-\sqrt{N_b+N_e})}{\sqrt{N_b}(\sqrt{N_b+N_e}-\sqrt{N_b})} < 0. \tag{2.43}$$

Hence, for all $N_b > 0$, the upper bound of β in (2.42) is always smaller than 0. Therefore, $\mu_{b,e} \neq \mu^b_{b,e}$ and $\mu^b_{b,e}$ cannot be the largest among the three $\mu^b_{b,e}$, $\mu^e_{b,e}$ and $\mu^c_{b,e}$. Based on the above analysis, scenario 1 can never happen in real cases.

- **Scenario 2** $\mu_{b,e} = \mu^e_{b,e}$: $\mu_{b,e} = \mu^e_{b,e}$ if and only if $\mu^e_{b,e} \geq \mu^b_{b,e}$ and $\mu^e_{b,e} \geq \mu^c_{b,e}$. Therefore, from (2.40),

$$(1-\beta) \leq \min\left\{\frac{\sqrt{N_e}}{\sqrt{N_b}}, \frac{\sqrt{N_e}(\sqrt{N_b+N_e}-\sqrt{N_e})}{N_b}\right\}. \tag{2.44}$$

Using the same analysis as in (2.42), the necessary and sufficient condition for scenario 2 is:

$$\mu_{b,e} = \mu^e_{b,e} \Leftrightarrow 1 - \frac{\sqrt{N_e}(\sqrt{N_b+N_e}-\sqrt{N_e})}{N_b} \leq \beta \leq 1. \tag{2.45}$$

- **Scenario 3** $\mu_{b,e} = \mu^c_{b,e}$: Since scenario 1 has been proven to not exist, $\mu_{b,e}$ must equal to one of $\mu^e_{b,e}$ and $\mu^c_{b,e}$. Therefore, the necessary and sufficient condition for Scenario 3 must be the compliment of the necessary and sufficient

condition for Scenario 2. Hence, :

$$\mu_{b,e} = \mu^c_{b,e} \Leftrightarrow 0 \leq \beta \leq 1 - \frac{\sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})}{N_b}. \tag{2.46}$$

From the above analysis on $P_d^{(i)}$, we can calculate the payoffs $\pi^{(i)}$ for all colluders for any given β. From the definition of the payoff function (2.4), colluders who receive fingerprinted copies of the same quality have the same payoff. We define $\pi_{b,e}$ as the payoff for colluders in $SC^{b,e}$, and π_b as the payoff for colluders in SC^b. Figure 2.10 illustrates π_b versus $\pi_{b,e}$, and the feasible set is shown by the solid line. The straight line segment corresponds to scenario 2, in which $\mu_{b,e} = \mu^e_{b,e} = \frac{\sqrt{N_e}}{K^{b,e}}$ and is independent of β. Therefore, $\pi^{b,e}$ keeps the same while π^b keeps decreasing with β increasing. Similarly, the curve segment in Figure 2.10 corresponds to scenario 3, in which $\mu_{b,e} = \mu^c_{b,e}$ and $\pi^{b,e}$ increases as β increases, while π^b decreases as β increases.

2. **Pareto Optimality**

After finding the feasible set, it is important to find the set of Pareto-Optimal points. A solution is Pareto-Optimal if and only if no player in the game can increase his/her payoff without decreasing others' [7]. In a bargaining situation, players would always like to settle at a pareto optimal outcome. This is because if the colluders select a point that is not Pareto-optimal, then there exists another solution where at least one player can have larger payoff without hurting the interest of other players. Therefore, the player who can have higher payoff without hurting others' has the incentive to push other players to deviate from the non-Pareto-optimal solution, and the other rational players will agree with him/her since their

interests are not influenced. Therefore, if possible, the colluders will always look for Pareto-optimal solutions to satisfy all the users in the colluder social network. Also, Pareto-optimal solutions are not unique in most cases. In this subsection, we investigate the Pareto-optimal points and analyzes the necessary and sufficient conditions for a point to be Pareto-optimal.

Note that from (2.40), colluders in SC^b can increase their payoff if and only if they select a smaller β. On the other hand, $\pi^{b,e}$ remains the same when scenario 3 happens. Therefore, we start our analysis of the Pareto-optimality by π^b.

- **Necessary Condition**: If a point is Pareto-Optimal, then decreasing μ_b and increasing the payoff of those colluders in SC^b must increase $\mu_{b,e}$ and decrease $\pi_{b,e}$. Note that from (2.40), μ_b is an increasing function of β. Thus, If a point is a Pareto-Optimal point, $\mu_{b,e}$ must be a decreasing function of β, which happens only when $\mu_{b,e} = \mu^c_{b,e}$. Consequently, if a point is Pareto-Optimal, β must satisfy (2.46), and (2.46) is the necessary condition of a Pareto Optimal point.

- **Sufficient Condition**: If $\mu_{b,e} = \mu^c_{b,e}$, then to increase the payoff of those colluders in $SC^{b,e}$, colluders must decrease $\mu_{b,e}$ by selecting a larger β. However, a larger β implies a larger μ_b, thus, it decreases the payoff of those colluders in SC^b. Consequently, those points that satisfy (2.40) are Pareto-Optimal points, and (2.40) is the sufficient condition of Pareto-Optimal.

To conclude, the collusion is Pareto-Optimal if and only if $\mu_{b,e} = \mu^c_{b,e}$ and (2.40) is satisfied, which is the curve segment in Figure 2.10.

3. **Absolute Fairness Solution** There are many ways for colluders to share the risk and the reward, depending on their definition of "fairness". Absolute fairness is widely adopted in the literature and most straight-forward, where all colluders have the same payoff. Based on the definition in (2.33) and the utility function in (2.4), the absolute fairness solution can be solve by

$$\frac{P_d^{b,e}(\beta)}{P_d^b(\beta)} = \frac{L^b + R}{L^{b,e} + R}, \tag{2.47}$$

where $P_d^{b,e}(\beta)$ and $P_d^b(\beta)$ are the sc^b and $sc^{b,e}$'s probability of being detected defined as in (2.40), and L^b and $L^{b,e}$ are the loss term claimed by the two players, respectively. According to the feasible set definition, $P_d^{b,e}(\beta)$ is a non-decreasing function of β, and $P_d^b(\beta)$ is a monotonely increasing function of β. Thus $P_d^{b,e}(\beta)/P_d^b(\beta)$ is a monotonely increasing function of β, and (2.47) can be easily solved by numerical method. then the absolute fairness solution exists, where $\beta' = 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$.

Optimal value of $L^{(i)}$: Suppose the absolute fairness solution exists, and sc^b wants to get more reward by falsely report the private information, the loss term L^b. Since π^b is a monotonely decreasing function of $P_d(\beta)$ and $P_d(\beta)$ is a monotonely decreasing function of $P_d^{b,e}(\beta)/P_d^b(\beta)$, π^b is a monotonely increasing function of $P_d^{b,e}(\beta)/P_d^b(\beta)$. Since $P_d^{b,e}(\beta)/P_d^b(\beta)$ satisfies (2.47) for the absolute fairness solution, it can easily be proven that if sc^b, instead of claiming the actual loss L^b, he/she cheats to claim a higher loss $L'^b > L^b$, the resulting absolute fairness solution will give a $P_d'^b < P_d^b$. Therefore, the bargained payoff π'^b by claiming higher loss is higher than the payoff π^b which is the absolute-fairness solution with honestly-reported loss L^b. Hence, sc^b can earn more payoff by cheating on his/her private

information. The same analysis can be applied to $sc^{b,e}$ and is not repeated here. To conclude, reporting higher loss will increase the user's payoff under absolute fairness condition. Thus any selfish and rational user is going to report the highest possible loss L_{max} to maximize his/her own interest. As a result, $L^b = L^{b,e} = L_{max}$, and based on (2.47), $P_d^{b,e} = P_d^b$ in absolute fairness solution.

4. **Max-Min Solution** In this example, the players' payoffs is affine to risk, hence the Max-Min solution can be rewritten as finding β_{maxmin} that

$$\beta_{maxmin} = arg \min_{\beta} \max \mu_b, \mu_{b,e}, \tag{2.48}$$

where μ_b and $\mu_{b,e}$ are defined in (2.40).

The Max-Min fairness solution with payoff function defined in (2.38) has the following property:

Properties: Max-Min solution always exists, and at least one of the Max-Min solution is Pareto-optimal. If the Max-Min solution is unique, then absolute fairness solution exists and the max-min solution is also the absolute fairness solution.

Proof: First prove the existence: since both π^b and $\pi^{b,e}$ are continuous functions of β, then $\min\{\pi^b, \pi^{b,e}\}$ is also a continuous function of β. Also $0 \leq \beta \leq 1$, therefore, the Max-Min solution always exists.

Suppose $\pi(\beta')$ is a Max-Min solution which is not Pareto-Optimal. Since $\pi^{b,e}$ remains the same in the feasible but not pareto-optimal set, and the largest π^b in the non-Pareto-optimal set is at the boundary to the Pareto-optimal set. Therefore, if $\pi(\beta') = \pi^{b,e}(\beta') <= \pi^b(\beta')$ is a Max-Min solution in the non-Pareto-optimal set,

the boundary $\beta'' = 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$ also gives a Max-Min solution because $\pi^b(\beta'') >= \pi^b(\beta') >= \pi^{b,e}(\beta') = \pi^{b,e}(\beta'')$. On the other hand, if $\pi^{b,e}(\beta') > \pi^b(\beta') = \pi(\beta')$ is a Max-Min solution in the non-Pareto-optimal set, then there exists a small positive number ε that $\pi^{b,e}(\beta' - \varepsilon) > \pi^b(\beta' - \varepsilon) = \pi^b(\beta') > \pi^{b,e}(\beta')$ in the non-Pareto-optimal set which contradict the assumption that $\pi(\beta')$ is the Max-Min solution.

If the Max-Min solution is unique, from the above proof, it can easily be shown that the solution must be Pareto-optimal. Since $\pi^b(\beta)$ is a monotonely decreasing function of β and $\pi^{b,e}(\beta)$ is a monotonely increasing function of β in the Pareto-optimal set, if $\pi_{maxmin}(\beta) = \pi^b(\beta) < \pi^{b,e}(\beta)$ is the unique Max-Min solution in the Pareto-optimal set, then there exists a small positive number ε that $\pi^{b,e}(\beta' - \varepsilon) > \pi^b(\beta' - \varepsilon) = \pi^b(\beta' - \varepsilon) > \pi^{b,e}(\beta')$ which contradicts the Max-Min assumption. Similarly, we can easily prove that $\pi_{maxmin}(\beta) = \pi^b(\beta) > \pi^{b,e}(\beta)$ also cannot be the unique Max-Min solution. As a result, the unique Max-Min solution must have the property that $\pi_{maxmin}(\beta) = \pi^b(\beta) = \pi^{b,e}(\beta)$ which is also the absolute fairness solution. □

Based on the above analysis, when the reward is evenly distributed among all colluders and Max-Min solution is unique, the Max-Min solution is similar to the absolute fairness solution with nice properties such as Pareto-optimal and existence. Solving Max-Min fairness is similar to solving absolute fairness, except the boundary points of the Pareto-optimal set have to be compared too.

Optimal value of $L^{(i)}$: If the Max-Min solution is unique, then it is the absolute

fairness solution by the above proof. Therefore, under such circumstance, reporting higher loss gives the player higher payoff and both players sc^b and $sc^{b,e}$ have the incentive to report the highest loss L_{max}.

If the Max-Min solution is not unique, based on the above analysis, some of the bargained solutions give $\pi^b(\beta) > \pi^{b,e}(\beta) = \pi^{b,e}_{max}$, where $\pi^{b,e}_{max}$ is the maximal payoff of $sc^{b,e}$. Hence the max-min solution gives maximal $\pi^{b,e}$ ($\beta = 1$). In such circumstance, the max-min solution already gives $sc^{b,e}$ the most advantage, as the result, $sc^{b,e}$ has no incentive to cheat on the loss term $L^{b,e}$ since he/she cannot earn more utility than the max-min solution.

On the other hand, if sc^b reports his/her lost to be $L^b + L'^b$, which makes $\pi'^b(\beta) = \pi^b(\beta) - L'^b < \pi^{b,e}(\beta) < \pi^{b,e}_{max}$ for some $0 \leq \beta < 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$. Thus by reporting higher loss term $L^b + L'^b$, sc^b can push the bargained max-min solution from the boundary of the Pareto-optimal set to the absolute solution inside the Pareto-optimal set. Apparently, from Figure 2.10, any point inside the Pareto-optimal set gives higher payoff for $sc^{b,e}$ than the boundary point of the Pareto-optimal set. Therefore, sc^b can gain higher payoff for by cheating on private information and sc^b has the incentive to report the highest loss $L^b = L_{max}$.

Base on the above analysis, sc^b always wants to report the highest loss, and sometimes $sc^{b,e}$ has the incentive to cheat (when the max-min solution is in the Pareto-optimal set) and sometimes does not. Since the loss L^b and $L^{b,e}$ are claimed before the bargaining process and $sc^{b,e}$ cannot predict whether the max-min solution will be Pareto-optimal before bargaining, the players should both claim L_{max} to ensure

the highest possible payoff.

5. **Max-Sum Solution** The Max-Sum solution can be formulated as minimizing

$$C_{sum} = P_d^b K^b (R + L^b) + P_d^{b,e} K^{b,e} (R + L^{b,e}). \tag{2.49}$$

As shown in the previous section, the max-sum solution is always Pareto-optimal. Therefore, the minimizer of the above function is either in the boundary or when the zero-deviation point. Taking the first derivative of the above function versus β, then

$$\frac{\partial C_{sum}}{\partial \beta} = \frac{\sigma_w}{\sqrt{2\pi}\sigma_n}[\ \frac{\sqrt{N_b}}{K^b} e^{-\frac{(h-\beta\sqrt{N_b}/K^b)^2}{\sigma_n^2}} K^b (R + L^b) - \frac{N_b}{K^{b,e}\sqrt{N_b+N_e}} e^{-\frac{(h-((1-\beta)N_b+N_e)/K^{b,e}\sqrt{N_b+N_e})^2}{\sigma_n^2}} K^{b,e}(R + L^{b,e})] \tag{2.50}$$

The max-sum solution can be solved numerically by the above equation.

Optimal value of $L^{(i)}$: Depending on the original Max-sum solution (both player report the loss honestly), the analysis of optimal value of $L^{(i)}$ can be divided into 3 cases: when $\beta = 1$, $\beta = 1 - \sqrt{N_e}(\sqrt{N_b+N_e} - \sqrt{N_e})/N_b$ or $\partial C_{sum}/\partial\beta$=0. From (2.50),

$$\begin{aligned} \frac{\partial^2 C_{sum}}{\partial \beta \partial L^b} &= \frac{(K^b + C)\sigma_w}{\sqrt{2\pi}\sigma_n} \frac{\sqrt{N_b}}{K^b} e^{-\frac{(h-\beta\sqrt{N_b}/K^b)^2}{\sigma_n^2}} K^b > 0, \text{ and} \\ \frac{\partial^2 C_{sum}}{\partial \beta \partial L^{b,e}} &= -\frac{(K^{b,e} + C)\sigma_w}{\sqrt{2\pi}\sigma_n} \frac{N_b}{K^{b,e}\sqrt{N_b+N_e}} e^{-\frac{(h-((1-\beta)N_b+N_e)/K^{b,e}\sqrt{N_b+N_e})^2}{\sigma_n^2}} K^{b,e} < 0 \text{ when } 0 \le \beta \le \end{aligned} \tag{2.51}$$

Hence, sc^b can push the max-sum solution to a smaller β (lower P_d^b thus higher payoff for sc^b) by reporting higher L^b. Similarly, $sc^{b,e}$ can also get higher payoff by reporting higher $L^{b,e}$. Hence both players have incentive to claim the highest loss L_{max}.

6. **Nash-Bargaining Solution**

Colluders may also select proportional fairness, where some colluders benefit more at a cost of higher risk. One popular solution is the Nash-Bargaining solution, which is based on the idea that players who can gain more will naturally ask for more in the bargain. The Nash-Bargaining solution is based on the definition of fairness that the additional payoff must be divided between the two players in a ratio equal to the rate at which this utility can be transferred.

The Nash-Bargaining solution is in the Pareto-Optimal set and, therefore, it always satisfies (2.45). Consequently, (2.37) becomes:

$$\begin{aligned}
&g(\beta) = A(\beta)^{B_{b,e}} B(\beta)^{B_b}, \text{ where} \\
&B(\beta) = (R+L^b)\left[Q\left(\frac{h-\frac{\sqrt{N_b}}{K^b}\sigma_w}{\sigma_n}\right) - Q\left(\frac{h-\frac{\beta\sqrt{N_b}}{K^b}\sigma_w}{\sigma_n}\right)\right], \\
&A(\beta) = (R+L^{b,e})\left[Q\left(\frac{h-\frac{\sqrt{N_b+N_e}}{K^{b,e}}\sigma_w}{\sigma_n}\right) - Q\left(\frac{h-\frac{(1-\beta)N_b+N_e}{K^b\sqrt{N_b+N_e}}\sigma_w}{\sigma_n}\right)\right]. \quad (2.52)
\end{aligned}$$

Note that Nash-Bargaining solution is always Pareto-Optimal and the set of β corresponding to the Pareto-Optimal points is closed. Thus, $g(\beta)$ is a concave function, and it is maximized when the gradient of $g(\beta)$ equals to zero or when β is on the boundary.

From (2.52), if $\partial g(\beta)/\partial\beta = 0$, then

$$\begin{aligned}
&\frac{N_b}{\sqrt{N_b+N_e}} B_{b,e} \frac{\partial A(\beta)}{\partial\beta} B(\beta) = \sqrt{N_b} B_b A(\beta) \frac{\partial B(\beta)}{\partial\beta}, \text{ where} \\
&\frac{\partial B(\beta)}{\partial\beta} = (R+L^b)\exp\left\{-\frac{(h-\frac{\beta\sqrt{N_b}}{K^b}\sigma_w)^2}{2\sigma_n^2}\right\}, \\
&\frac{\partial A(\beta)}{\partial\beta} = (R+L^{b,e})\exp\left\{-\frac{(h-\frac{(1-\beta)N_b+N_e}{K^{b,e}\sqrt{N_b+N_e}}\sigma_w)^2}{2\sigma_n^2}\right\}. \quad (2.53)
\end{aligned}$$

Note that both $B(\beta)$ and $\partial A(\beta)/\partial\beta$ are increasing functions of β, while $A(\beta)$ and $\partial B(\beta)/\partial\beta$ are decreasing functions of β. Thus, the solution of (2.53) is a monotonically decreasing function of $B_b/B_{b,e}$. It implies that the subgroup of colluders with a larger bargaining power benefits more than the others even the bargaining. Depending on the criteria of setting bargaining power in the Nash-bargaining problem, the bargaining power may change in different colluder social networks. One of the most common bargaining power is using the number of colluders, K^b and $K^{b,e}$.

Optimal value of $L^{(i)}$: Note that in (2.53), both sides of the equate have the common term $(R+L^b)(R+L^{b,e})$ and can be eliminated. Hence, the Nash-bargaining solution does not depend on the users' loss L^b and $L^{b,e}$. Therefore, the Nash-bargaining solution can be considered as cheat-proof, which is, the bargained solution remains the same even the players cheat on the private information.

To conclude, both players sc^b and $sc^{b,e}$ can gain higher reward by reporting higher loss if the fairness criteria is absolute fairness, Max-Min, or Max-Sum. And the Nash-bargaining solution is not influenced by the private information $L^{(i)}$ of each player. However, the loss is declared before the bargaining process, and at then the colluders do not know which solution the bargaining process will converge to. Therefore, both players sc^b and $sc^{b,e}$ have the incentive to report as higher loss as possible, resulting in $L^b = L^{b,e} = L_{max}$ being the same for all colluders in the utility function definition (2.4).

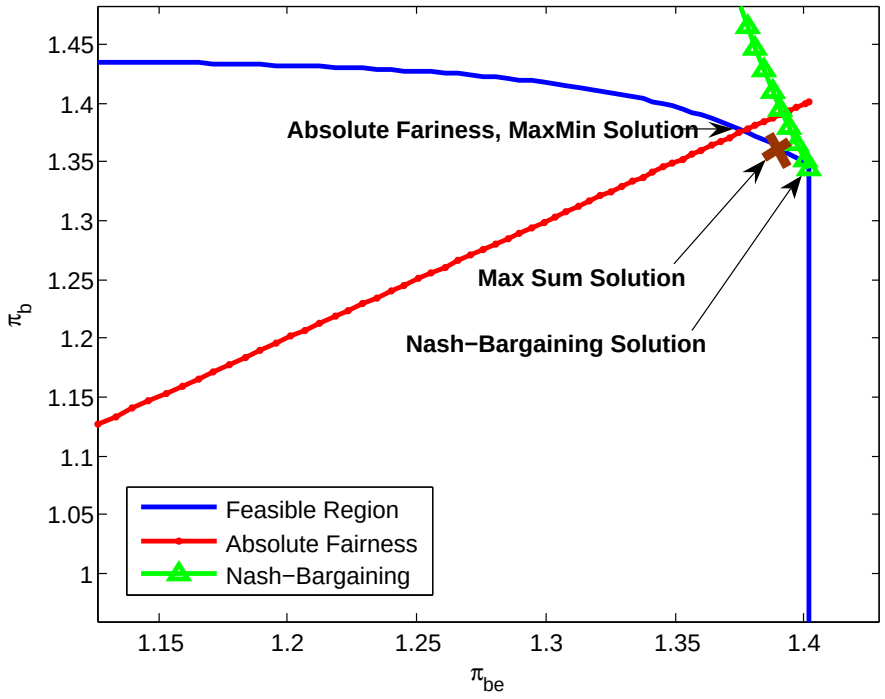


Figure 2.11: Feasible region and bargaining solutions with utility function as in (2.38),$P_{fa} = 10^{-3}$, $N_b = N_e = 50000$, $K^b = 100$, $K^{b,e} = 150$, and $|U^b| = |U^{b,e}| = 250$.

2.2.2.2 Case two: Reward is proportional to risk

In real-world social networks, reward is usually distributed unequally among the colluders. There are multiple reasons for the uneven reward distribution, for instance, each member has his/her own personal concern and position in the society. Therefore, some colluders might be more greedy and want to gain more reward in this collusion. Intuitively these colluders have to pay more cost (probability of being detected) to maintain fairness in the colluder social network. To address this issue, we also consider the more general utility function,

$$\pi^{(i)} = -P_d^{(i)} * L^{(i)} + \left(1 - P_d^{(i)}\right) \frac{F^c / F^{max}}{\left(K^b \left(F^b\right)^{0.1} P_d^b + K^{b,e} \left(F^{b,e}\right)^{0.1} P_d^{b,e}\right)/M} \left(F^{(i)}\right)^{0.1} P_d^{(i)} \quad (2.54)$$

to illustrate the feasible region and the bargaining solution when the colluders distribute reward proportional to each copy's quality and risk (probability of being detected).

In this case, the reward each colluder gets is linear to his/her probability of being detected. Also, colluders who subscribe to higher resolution copy also gain more reward. The analysis of the four bargaining solutions are similar as in Section 2.2.2.1 and not repeated here. Based on the same analysis, we can also conclude that both players have the incentive to report highest loss $L^b = L^{b,e} = L_{max}$ before collusion. Hence, we will show the bargaining solutions for this case by the simulations with both colluders claiming loss term L_{max}.

2.2.2.3 Simulation Setting and Results

In our simulations, we first generate independent vectors following Gaussian distribution $\mathcal{N}(0,1)$, and then apply Gram-Schmidt orthogonalization to generate orthogonal fingerprints. The lengths of the fingerprints embedded in the base layer and the enhancement layer are $N_b = N_e = 50000$, and both two layers contain 20 frames, respectively. The total number of users is 500, where $U^b = U^{b,e}$. The probability of accusing an innocent user, P_{fa}, is 10^{-3}. Among the $K = 250$ colluders, $K^b = 100$ of them receive the fingerprinted base layer only, and the other $K^{b,e} = 150$ of the colluders receive fingerprinted copies of high resolution.

Figure 2.11 shows the feasible region and the four bargaining solutions with utility function as in (2.38), and bargaining powers in (2.37) are $B_b = 2, B_{b,e} = 3$, which is proportional to K^b and $K^{b,e}$. Compared to the absolute fairness solution, the max-sum

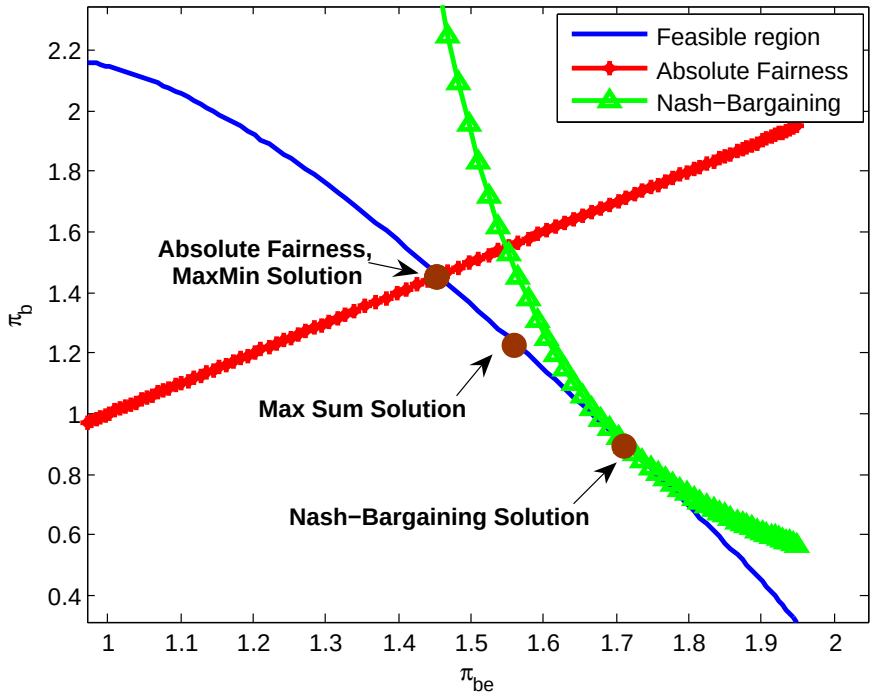


Figure 2.12: Feasible region and bargaining solutions with utility function as in (2.54),$P_{fa} = 10^{-3}$, $N_b = N_e = 50000$, $K^b = 100$, $K^{b,e} = 150$, and $|U^b| = |U^{b,e}| = 250$.

solution gives the group with more people more utility, which is $sc^{b,e}$ in this case. The Nash-Bargaining with bargain power $B_b = 2, B_{b,e} = 3$ even more favor $sc^{b,e}$ since now the number of colluders works as the exponential term rather than the linear term in the max-sum solution. The other reason for such phenomenon is, in this simulation setting, K^b is much smaller than $K^{b,e}$ ($2/3$ of $K^{b,e}$), therefore, according to the definition of Nash-bargaining solution in (2.37) the highest risk of SC^b, which can be considered as SC^b collude alone without $SC^{b,e}$ is much higher then that for $SC^{b,e}$. Therefore, the minimal payoff of SC^b, π^{b*}, is also smaller than the minimal payoff of $SC^{b,e}$, resulting in SC^b has more extra payoff for bargaining thus leads to better bargain position. Setting the bargain-

ing power to be the number of colluders who receive different quality copies matches the real-world scenario: the group of colluders with more users act together and should have more bargain power.

Figure 2.12 shows the feasible region and the four bargaining solutions with utility function defined in (2.54). First, the whole feasible set is Pareto-optimal since π^b is a monotonely decreasing function of $\pi^{b,e}$ as shown in the figure. There is no non-Pareto-optimal feasible points as the straight line segment in Figure 2.11. The reason for such result is that, although for all $\beta > 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$, $P_d^{b,e}$ is the same, but P_d^b keeps reducing as β increases. Hence, for all $u^{(i)}$ who receives higher resolution copies, the denominator of the second term in the utility function (2.54) keeps increasing as β increases while the numerator is the same. As a result, unlike case 1 in which $\pi^{b,e}(\beta)$ is a constant for all $\beta > 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$, $\pi^{b,e}(\beta)$ is a decreasing function of β when $\beta > 1 - \sqrt{N_e}(\sqrt{N_b + N_e} - \sqrt{N_e})/N_b$ in case 2 thus all the points in the feasible set are also Pareto-optimal.

The four bargaining solution in Figure 2.12 shows the same trend as in Figure 2.11: the max-min solution is the same as the absolute fairness solution, the max-sum solution favors $sc^{b,e}$ better that, and the Nash-bargaining solution with $B_b = 2, B_{b,e} = 3$ gives $sc^{b,e}$ maximal utility. The same trend of the four bargaining solutions in these two cases shows our methodology can fit to different collusion problems once the utility function is defined since our analysis in on the bargaining level and the trend of the bargaining solutions are independent of utility function definitions. Nevertheless, the "absolute fairness solution" under proportional reward distribution also has proportional fairness characteristics.

Furthermore, comparing the feasible region in Figure 2.11 and Figure 2.12, it is

clear that both the maximum utilities that sc^b and $sc^{b,e}$ can achieve are much higher if reward is distributed proportionally (π^b_{max} = 1.441 and $\pi^{b,e}_{max}$ = 1.403 in Figure 2.11 while π^b_{max} = 2.182 and $\pi^{b,e}_{max}$ = 1.947 in Figure 2.12). These maximal utilities happen for extreme β value when it approaches to 1 or $1-\sqrt{N_e}(\sqrt{N_b+N_e}-\sqrt{N_e})/N_b$, under which one of P^b_d or $P^{b,e}_d$ is much higher than the other, and one of sc^b or $sc^{b,e}$ earn most of the reward resulting in high payoff.

2.2.3 Time-sensitive Bargaining Model

In the previous subsections, we have discussed several fairness criteria. The two players sc^b and $sc^{b,e}$ can keep offering each other until one of the fair solution is achieved. In such model, we did not take the time-sensitiveness of the value of multimedia signals into account. In this section, we further extend our behavior modelling of the colluder social networks to a time-sensitive bargaining model, provide the optimal bargaining offer of both players in the game, and reach the equilibrium.

2.2.3.1 Bargaining Model and Payoff Functions

The reward of redistributing the colluded multimedia signal depends on not only the colluded copy's quality, but also on the time that the copy being released. The market value of colluded copy with lower quality decreases faster than higher-resolution copy. For instance, when the movie is still in theater, people might want to watch the low-resolution colluded copy to catch the trend. But if the movie is off theater and its DVD has been released, people might still want to purchase the high-resolution pirated copy since the

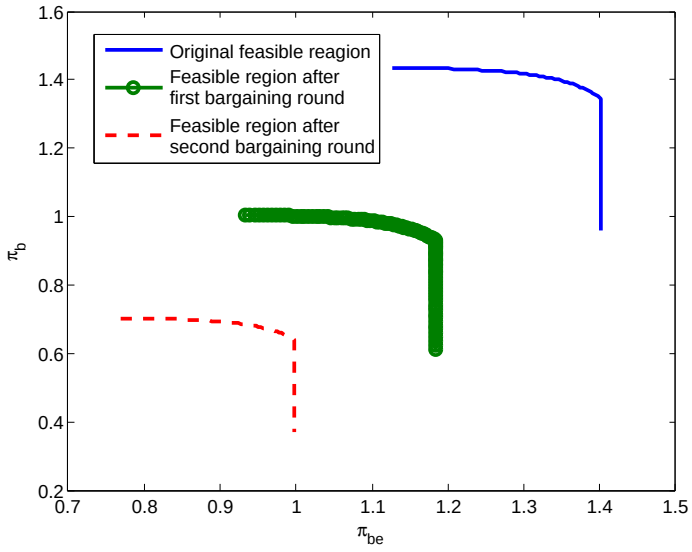


Figure 2.13: Feasible region for bargaining after the first two rounds

cost would be lower than DVD, but the incentive of paying for low-resolution pirated version is very little, since the DVD is easily accessible and not very costy. Also, if the colluded copy is the only pirated copy in the market, all the market value will go for it and not shared with other copies. Therefore, the colluders are competing not only with the movie industry but also the other colluders over the speed of generating the pirated copy. An illustration of the time-sensitiveness of the colluders' reward is shown in Figure 2.13. The blue solid curve is the feasible region that the colluders can bargain with before the bargaining process, the green circled curve and the red dashed curve are the feasible region after the first and second bargaining rounds, respectively. As in Figure 2.13, the colluders have to finish their bargaining process as soon as possible, to avoid the utility loss.

Under such circumstance, both groups of colluders, sc^b and $sc^{b,e}$ would want to reach agreement as soon as possible. We model the process of reaching agreement among

colluders using a time-sensitive bargaining model:

- In the first bargaining stage, $sc^{b,e}$ offers the collusion parameter β_1 that uniquely maps to the utility pair $(\pi_1^b, \pi_1^{b,e})$ on the Pareto-optimal set, in which both sc^b and $sc^{b,e}$ can not increase their payoff without decreasing the other's. The example of Pareto-optimal set is illustrated in Fig 2.10.

- Upon receiving the offer, sc^b has the choice to accept this offer and gets the payoff π_1^b, or reject and offer back β_2, which corresponds to payoff pair $(\pi_2^b, \pi_2^{b,e})$ and continues to the second stage.

- If sc^b decided to offer back, $sc^{b,e}$ again has the choice to accept the offer $(\pi_2^b, \pi_2^{b,e})$ or offer back. The bargaining process would continue until both groups of colluders agree on one offer.

Here we adopt the exponentially decay model for the market value of the colluded copy [7]. The reward that player i get in the next round of bargaining will be decayed by a constant $\delta_{(i)}$. If the two players sc^b and $sc^{b,e}$ reach agreement at the k^{th} bargaining stage, their reward would be decreased to δ_b^{k-1} and $\delta_{b,e}^{k-1}$ times, which means:

$$\pi_k^{(i)} = -P_d^{(i)}(\beta_k) * L + \left(1 - P_d^{(i)}\right) \delta_{(i)}^{k-1} R^{(i)}, \tag{2.55}$$

where $(i) \in \{(b), (b,e)\}$ and $R^{(i)}$ as defined in (2.5). $0 < \delta_b < 1$ and $0 < \delta_{b,e} < 1$ are the reward-decay constant of sc^b and $sc^{b,e}$, respectively. The market value of the high-resolution copy is more resistant to time than low resolution copies. For instance, after the DVD of the movie available at the rental stores, the low-resolution copies almost have no value in the market, but high-resolution copies still conserve parts of the value as long

as their prices are lower than the rental fee. Therefore, a reasonable constraint of the decaying factors is $\delta_{b,e} \geq \delta_b$.

In this model, $sc^{b,e}$ makes offer first since colluders with higher resolution copies take advantage during bargaining. This advantage comes from that even $sc^{b,e}$ cannot reach agreement with sc^b, they can still release their high-resolution colluded copy with high market value, but on the other hand, sc^b themselves can only generate low-resolution colluded copy. Hence $sc^{b,e}$ has more bargain power over sc^b, and should make the offer first.

The *equilibrium* in this time-sensitive bargaining game is the "offer pairs" that both players will agree immediately upon offered. From the offerer side of view, he/she wants to make the offer attractive enough that the other player will agree on the offer and not offer back to reserve the full value of the colluded multimedia signal. On the other hand, the offerer also does not want to make an offer that benefit the other player too much an hurt his/her own interest. Therefore, the equilibrium pair $((\pi_k^b, \pi_k^{b,e}), (\pi_{k+}^b 1, \pi_{k+1}^{b,e})$ that the colluders would reach agreement at the k^{th} bargaining stage has the following property: suppose $SC^{b,e}$ makes an offer $(\pi_k^b, \pi_k^{b,e})$ at the k^{th} stage, then π_k^b should be large enough that sc^b will accept it, and no larger. On the other hand, sc^b should accept π_k^b if it is not smaller than the discounted payoff $-P_d^{(i)}(\beta_{k+1}) * L + \delta_{(i)}^k R^{(i)}$ that sc^b would receive if $SC^{b,e}$ accept their counter offer. Therefore,

$$-P_d^b(\beta_k) * L + \left(1 - P_d^{(i)}\right) \delta_b^{k-1} R^b = -P_d^b(\beta_{k+1}) * L + \left(1 - P_d^{(i)}\right) \delta_b^k R^b, \tag{2.56}$$

and a similar consideration (for the dual game) for $SC^{b,e}$ gives

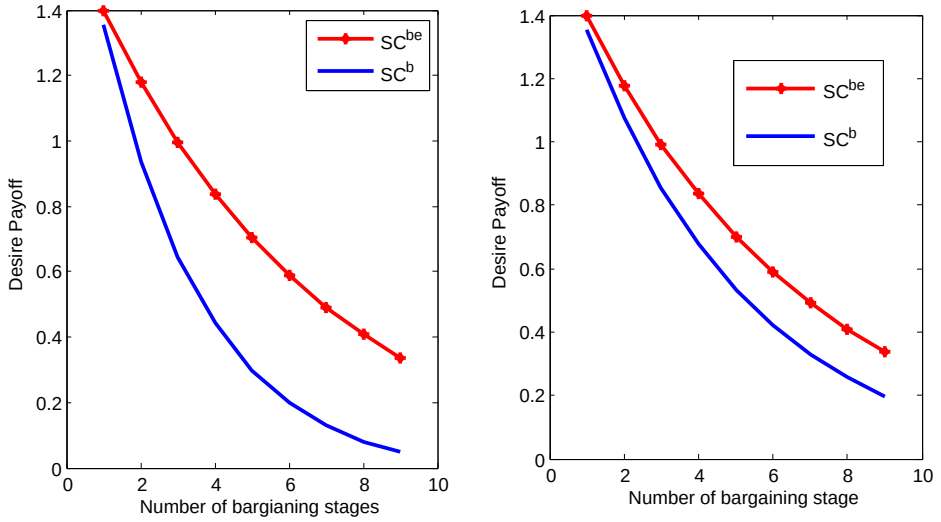
$$-P_d^{b,e}(\beta_{k+1}) * L + \left(1 - P_d^{(i)}\right) \delta_{b,e}^k R^{b,e} = -P_d^{b,e}(\beta_k) * L + \left(1 - P_d^{(i)}\right) \delta_{b,e}^{k-1} R^{b,e}. \tag{2.57}$$

We assume the worst-case scenario for the fingerprint detector that the colluders have perfect information about the detector's detection strategy. This is the widely adopted concept in the collusion analysis toward the best protection of multimedia. Thus $P_d^{b,e}(\beta)$ and $P_d^b(\beta)$ are known to the colluders. So we have two linear-independent equations with two unknowns and the time-sensitive bargaining equilibrium can only be found numerically since $P_d^{b,e}(\beta)$ and $P_d^b(\beta)$ involve the Gaussian tail function.

2.2.3.2 Case Study and Simulation Results

In this section, we take the second utility functions as in Section 2.2.2 as examples to illustrate the time-sensitive bargaining in the colluder social network. Since in real-world social networks, reward is usually distributed unequally because every member has different personal concern and position in the society, thus we consider the general utility function as in (2.54) to illustrate the time-sensitiveness when the colluders distribute reward proportional to each copy's quality and the user's risk (probability of being detected). We apply our analysis to the real video data and verify our results.

In our simulations, we test over the first 40 frames of "carphone", and use $F_b = \{1,3,\cdots,39\}$ and $F_e = \{2,4,\cdots,40\}$ as an example of the temporal scalability. The lengths of the fingerprints embedded in the base layer and enhancement layer are $N_b = 85938$ and $N_e = 85670$ respectively. We assume that there are a total of $M = 500$ users and $|\mathbf{U}^b| = |\mathbf{U}^{b,e1}| = 250$. We first generate independent vectors following Gaussian distribution $\mathcal{N}(0,1/9)$, and then apply Gram-Schmidt orthogonalization to generate orthogonal fingerprints for different users.



(a) $\delta_b = 0.7$ and $\delta_{b,e} = 0.85$ (b) $\delta_b = 0.8$ and $\delta_{b,e} = 0.85$

Figure 2.14: Utilities of SC_b and $SC^{b,e}$ versus number of bargaining rounds.$P_{fa} = 10^{-3}$, $N_b = N_e = 50000$, $K^b = 100$, $K^{b,e} = 150$, and $|U^b| = |U^{b,e}| = 250$ with different discount factors.

During collusion, the colluders apply the intra-group collusion followed by the inter-group collusion, and follow the above analysis when choosing the collusion parameters. In our simulations, we adjust the power of the additive noise such that $||\mathbf{n}_j||^2 = ||JND_j\mathbf{W}_j^{(i)}||^2$ for every frame j in the video sequence. The probability of accusing an innocent user, P_{fa}, is 10^{-3}. Among the $K = 250$ colluders, $K^b = 100$ of them receive the fingerprinted base layer only, and the other $K^{b,e} = 150$ of the colluders receive fingerprinted copies of high resolution.

Figure 2.14 shows the bargaining equilibrium versus the number of stages that the colluders need to make agreement with utility function as in (2.54), and different discount factors: Figure 2.14(a) uses $\delta_b = 0.7, \delta_{b,e} = 0.85$, and Figure 2.14 (b) is the result

of $\delta_b = 0.7, \delta_{b,e} = 0.85$. The feasible region and the Pareto-optimal set with the same utility function for the first stage of the game is shown in Figure 2.13. It is clear from Figure 2.14 that both colluders have incentive to finish the bargaining process as soon as possible under both settings of discount constants, especially for SC^b, who's utility decays faster than $SC^{b,e}$. Therefore, at the very first bargaining stage, the first-mover will offer based on the equilibrium by solving (2.56) and (2.57). Thus SC^b would let $SC^{b,e}$ to take the advantage of offering first. It is clear by comparing Figure 2.14(a) and (b) that higher discount factor results in higher payoff. The discount factors $\delta_b, \delta_{b,e}$ can also be considered as the power of bargaining for SC^b and $SC^{b,e}$. For instance, if the two groups of colluders cannot make agreement and they decide to collude within groups and generate two colluded copy with different qualities. Apparently SC^b would get much less reward than $SC^{b,e}$ since their colluded copy has lower quality. Thus SC^b has much more intention to cooperate with $SC^{b,e}$, and this intention leads to less bargaining power.

2.3 Equilibriums of the Detector-Colluder Game

In the previous sections, we have discussed how the colluders bargain with each other and what are the fair types of collusion that can satisfy all colluders and lead to a successful collusion. A successful collusion must not only be fair to all colluders but also maximize all colluders' utility under the fairness constraint. On the other hand, the fingerprint detector also has to adjust its strategy according to the collusion type to achieve highest probability of detection. Therefore, there exists complex dynamics among the colluders and the fingerprint detector and they altogether also form a social network, called the mul-

timedia fingerprint social network [17]. Hence the bargaining solutions that the colluders being willing to follow have to be the best response to the detector's optimal strategy and form equilibriums between the fingerprint detector and the colluders.

Hence, in this section, we will prove that the bargaining solutions we discussed in this section are also the equilibria strategies for the colluders in the detector-colluder game thus are the best move for the colluders under different fairness constraints.

2.3.1 Stackelberg Game Model of dynamics between colluders and fingerprint detector

To capture users' behavior in strategic situations, in which an individual's success in making choices depends on the choices of others, Game Theory [7], [8] is a useful tool to model the complex dynamics among multimedia social network members. Therefore, to analyze the optimal strategies of both fingerprint detector and the colluders under the fairness constraints, we formulate the interaction between the two groups of the multimedia fingerprinting social network users as a game with two players: the colluders acting as one single player and the fingerprint detector as the other [17].

Game between colluders and fingerprint detector

- **Players**: There are two players: colluders who make the move first as the leader, followed by the follower, who is the fingerprint detector that applies detection after receiving a suspicious copy.

- **Payoff Function**: In this game, what colluders gain is the lost of the detectors, thus the two group of users, colluders and the fingerprint detector in the fingerprinting

social network have totally conflict objectives. Therefore, the sum of the utilities of all colluder equals to the utility of the digital right enforcer with negative sign. Based on the utility of each individual colluder during bargaining as in (2.4) and the assumption that all the colluders, the payoff functions of the colluders and the fingerprint detector can be defined as

$$\pi_C = R_{sum} - \ P_d^b K^b (L_{max} + R^b) - P_d^{b,e} K^{b,e} (L_{max} + R^{b,e})$$
$$\text{and} \qquad \pi_D = -\pi_C, \tag{2.58}$$

where R_{sum} is the total reward of redistributing the colluded copy, and π_C and π_D are the utility functions for colluders and the fingerprint detector, respectively.

Based on the utility function definition as in (), all colluders has the same goal of minimizing his/her risk of being detected $P^{(i)}$ under fairness constraint. From the detector's point of view, whatever the colluders' gain is the lost of the digital right enforcer, so we can define the detector's payoff as $\pi_D = -\pi_C$. Therefore, to maximize his/her own payoff, the fingerprint detector also have the incentive to maximize the probability of catching colluders in both group, P_d^b and $P_d^{b,e}$.

- **Colluders' Strategies**: The colluders' strategies are the set of all possible collusion parameter β that achieves fairness for each colluder leads to one strategy for the colluders in the colluder-detector game. Therefore, the colluders have uncountably infinite number of strategies.

- **Detector's Strategies**: Since the fingerprinting is Gaussian and orthogonal, the best detector is the correlation detector. Upon receiving the suspicious copy, the

correlation-based fingerprint detector can decide which part of the suspicious he/she is going to use for detection. Note that for users in SC^b, since their copies only contain the base layer, the detector only has one choice, which is utilizing the whole base layer for identification.

Hence, as discussed in Chapter 2, the detector's strategies includes the collective detector, single-layer detector, and the self-probing detector. The collective detector uses the whole sequence to identify $SC^{b,e}$; the single-layer detector uses either base layer or enhancement layer to identify $SC^{b,e}$; the self-probing detector, as introduced in Section 2.1.1, probe the side information (the mean of the detection statistics) first, and then choose to use collective detector or the single-layer detector for detection.

In this game, there are multiple detection statistics that the fingerprint detector can use to identify colluders. However, by the analysis and simulation results shown in our previous work [17], the self-probing detector can always achieves better or equal performance as all other detectors (collective detector and single-layer detector). Thus to maximize his/her payoff, the fingerprint detector always probes side information about collusion and selects the detection statistics that has the largest chance of successfully capturing colluders.

From the angle of game theoretical analysis, probing side-information is equivalent to observing the colluders' action. The near-optimal performance of the self-probing detector implies the detector (follower in this game) can observe the colluders' action completely. Furthermore, to provide the best protection of multimedia content, here we

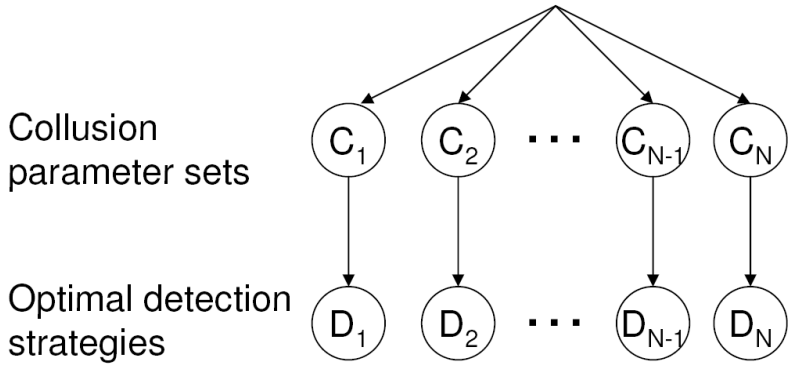


Figure 2.15: Game tree illustration of the colluder-detector dynamics. $C_1, C_2, \cdots, C_N$ are the N possible sets of collusion parameters that achieve bargaining solutions under various fairness constraints when the fingerprint detector uses the optimal detection statistics to identify colluders; while D_1, D_2, $\cdots$, D_N are the corresponding optimal fingerprint detection strategies.

assume the worst case scenario that the colluders know exactly the detector's strategy, which means the colluders (leader) know that the detector observes their action. Hence, colluders as the leader have perfect knowledge of the detection strategies that the fingerprint detector will use, because the detector has no incentive to deviate from the self-probing detector. Therefore, the detector have no means of committing to a follower action that deviates from the self-probing detector, which is the best response, and the colluders know this. Therefore, the colluder-detector game is a Stackelberg game [8] with perfect information.

2.3.2 Equilibrium Analysis

With the self-probing fingerprint detection process in Section 2.1.1, for each type of collusion, the fingerprint detector can always choose the detection statistics that gives the best probability of detection performance for $SC^{b,e}$. Such phenomenon can be illustrated as the game tree shown in Figure 3.5. In this game, assuming that there are N possible collusion strategies under the fairness constraint (can be either absolute fairness, max-min fairness, max-sum fairness, Nash-Bargaining, or the time-sensitive bargaining), the colluders first choose the fair collusion strategy based on Section 2.2 and Section 2.2.3, and then the fingerprint detector selects the optimal detection statistics.

Since the follower (detector) can observe the leader's (colluders') strategy, the equilibrium of the game model can be solved by backward induction. By backward induction, since both the colluders and the fingerprint detector know that the optimal detection statistics will be used to identify colluders, once attackers determine the collusion strategy, their payoff is fixed and the colluders can accurately estimate their payoff. The colluders consider what the best response of the detector is, i.e. how the detector will respond once he/she observes the leader's strategy. The colluders then pick a strategy that maximizes its payoff, anticipating the predicted response of the detector. The detector actually observes this by using the self-probing detector and in equilibrium picks the expected quantity as a response.

Hence, the equilibrium of the detector-colluder game is as follows: during collusion, colluders should always consider the self-probing detector as the detector's strategy, and find the bargaining solutions under the fairness constraint. And on the other hand, the

detector always use the self-probing detector. Since the bargaining solutions discussed in Section 2.2 and Section 2.2.3 are based on the self-probing detector, they are the equilibria strategies of the colluders.

2.4 Chapter Summary

This chapter studies user dynamics in multimedia social networks and provides a case study of cooperation analysis for multiuser collusion in multimedia fingerprinting. We build a game-theoretic framework to analyze the complex dynamics among colluders and model the dynamics among colluders as a bargaining process, where colluders negotiate with each other to achieve fairness of collusion. In this chapter, we analyze the necessary conditions for attackers to cooperate with each other, examine the impact of the selection of fellow attackers on an attacker's payoff, and investigate the fair collusion strategies that the colluders will all agree with.

We first model the fairness dynamics among colluders as a non-cooperative game, where each colluder aims to maximize his/her own utility through bargaining to achieve fair agreement. We propose a general model of utility functions which allows uneven reward-distribution. We then consider a scenario where all attackers receive fingerprinted copies of the same resolution and the colluded copy is a simple average of all copies with equal weights. In such a scenario, we first investigate K_0, the smallest number of colluders that gives attackers a non-negative payoff. Attackers collude with each other if and only if the total number of colluders is larger than or equal to K_0. We then show that colluding with more attackers does not always increase a colluder's payoff, and analyze

the optimum number of colluders (K_{max}) that maximizes a colluder's utility.

We further extend scenario to that attackers receive fingerprinted copies of different resolutions. Our analysis shows that in this scenario, colluding with more attackers does not always increase an attacker's payoff and attackers may not always want to cooperate with each other. They collude with each other if and only if cooperation helps increase all attackers' utilities. We first investigate the necessary conditions for colluders to cooperate with each other. We analyze $\mathbb{K}_p$, the set including all pairs of (K^b, K^{be}) where it is possible for all colluders to benefit from cooperation, and explore all possible collusion strategies that increase every attacker's utility for a given $(K^b, K^{be}) \in \mathbb{K}_p$. We then examine how the number of colluders in each subgroup, (K^b, K^{be}), affects colluders' utilities.

In the last part, we analyze human behavior by four bargaining criteria: absolute fairness, max-min, max-sum, and Nash-Bargaining solution. Then we extend our model to address the special time-sensitive property of multimedia contents analyze the colluders' behavior by modelling collusion as a time-sensitive bargaining process and find the equilibrium of the bargaining game. Our analysis shows that in the colluder social network, the colluders will make agreement at the first bargaining stage and reach equilibrium; and if the market value of the colluded copy is not time-sensitive, colluders choose different points in the feasible set, depending on the colluders' definition of "fairness" and their agreement on how to distribute the risk and the profit among themselves. Furthermore, we also prove that all the bargaining solutions that satisfy the fairness criteria are also the equilibrium in the colluder-detector game. Such result shows the bargaining solutions are the best strategies for the colluders under the fairness criteria with the corresponding optimal correlation-based detector that all colluders would satisfy and not

deviate from. Therefore, the possible types of collusion can be reduced to the set of these bargaining solutions. This chapter provides a methodology that can fit human behavior analysis in different social networks.

Chapter 3

Equilibriums of Multimedia Fingerprinting Social Networks with Side Information

Multimedia fingerprinting is an emerging forensic tool to protect multimedia from illegal alteration and unauthorized redistribution. It uses traditional data-hiding techniques [38] to embed a unique label, known as "fingerprint", into each distributed copy to track the usage of multimedia data. Multiuser collusion is a powerful attack against multimedia fingerprinting, where a group of attackers collectively and effectively mount attacks to remove traces of the identifying fingerprints [39]. To offer consistent and reliable traitor tracing, multimedia fingerprinting should resist such multiuser collusion as well as attacks by a single adversary [40].

In multimedia fingerprinting, colluders and the fingerprint detector form a multimedia social network: colluders who apply multiuser collusion attempt to remove the identifying fingerprints in their copies, and the digital rights enforcer detects the embedded fingerprints in the suspicious copy to capture colluders. There are many collusion strategies that the colluders can use to remove the identifying fingerprints. Also, the de-

tector can apply different detection strategies to identify the colluders. Most prior work focuses on the modelling and analysis of collusion and design multimedia fingerprints that can resist collusion attacks [10–12, 14, 41–45]. However, the complicated dynamics between the colluders and the fingerprint detector has not been studied and the investigation of possible side information that can help detection is also lacked.

In this chapter, we investigate two important issues in multimedia fingerprinting social networks. First, we study the impact of the dynamics between the two group of users (colluders and the fingerprint detector) in the social network when *side information* is available. If some information of collusion attacks can be made available during the colluder identification process, intuitively, utilizing such information can help improve the traitor tracing performance. We define this information about collusion that can improve detection probability as side information. Second, we model the user dynamics using a game-theoretic framework and find the optimal strategies for all users.

The rest of the chapter is organized as follows. Section 3.1 introduces the multimedia fingerprinting system. In Section 3.2, we investigate how the fingerprint detector probes and utilizes side information about collusion to improve the collusion resistance. In section 3.3, we analyze the equilibrium of the colluder-detector game, study the colluders' strategies to minimize their risk under the fairness constraint, and finds the solution to the min-max formulation of the colluder-detector dynamics. Section 3.4 shows the simulation results, and conclusions are drawn in Section 3.5.

3.1 Multimedia Fingerprinting System

In this section, we will briefly introduce the structure and users involved in a multimedia fingerprinting social network.

3.1.1 Temporally Scalable Video Coding Systems

As multimedia networking develops, scalability in multimedia coding becomes increasingly important for rich media access from anywhere by anyone [46]. Scalable video coding encodes multimedia into several bit streams (or layers) of different priorities; the base layer contains the most important information and must be received by all users, while the enhancement layers refine the resolution of the receiver's reconstructed copy and have lower priorities. Such an encoding structure provides flexible solutions for multimedia transmission and offers adaptivity to heterogeneous networks, varying channel conditions and diverse computing capability at the receiving terminals.

Without loss of generality, we use temporally scalable video coding as an example which provides multiple versions of the same video with different frame rates. Following the same model in [15], we consider a temporally scalable video coding system with three-layer scalability, and we use frame skipping and frame copying to implement temporal decimation and interpolation, respectively. In such a video coding system, different frames in the video sequence are encoded in different layers. Define F_b, F_{e1} and F_{e2} as the sets containing indices of the frames that are encoded in the base layer, enhancement layer 1 and enhancement layer 2 respectively. $F^{(i)}$ includes the indices of the frames in the copy that user $\mathbf{u}^{(i)}$ receives. $\mathbf{U}^b = \{i : F^{(i)} = F_b\}$ is the subgroup of users who re-

ceive the base layer only, $\mathbf{U}^{b,e1} = \{i : F^{(i)} = F_b \cup F_{e1}\}$ contains all users who subscribe to the medium-resolution version with the base layer and the enhancement layer 1, and $\mathbf{U}^{all} = \{i : F^{(i)} = F_b \cup F_{e1} \cup F_{e2}\}$ contains the indices of the users who receive all three layers.

3.1.2 Multimedia Fingerprinting System and Collusion Attacks

3.1.2.1 Fingerprint Embedding

Proven to be robust against many single-copy attacks and common signal processing, spread spectrum embedding is a popular data hiding technique to embed fingerprints into the host multimedia signals [39, 47]. For the j^{th} frame in the video sequence represented by a vector $\mathbf{S}_j$, and for each user $\mathbf{u}^{(i)}$ who subscribes to frame j, the content owner generates a unique fingerprint $\mathbf{W}_j^{(i)}$ of the same length as $\mathbf{S}_j$. The fingerprinted frame j that is distributed to $\mathbf{u}^{(i)}$ is $X_j^{(i)}(k) = S_j(k) + JND_j(k) \cdot W_j^{(i)}(k)$, where $X_j^{(i)}(k)$, $S_j(k)$ and $W_j^{(i)}(k)$ are the kth components of the fingerprinted frame $\mathbf{X}_j^{(i)}$, the host signal $\mathbf{S}_j$ and the fingerprint vector $\mathbf{W}_j^{(i)}$, respectively. JND_j is used to control the energy and achieve the imperceptibility of the embedded fingerprints [47].

We consider orthogonal fingerprint modulation [10] in this chapter. We first generate independent vectors following Gaussian distribution $\mathcal{N}(0, \sigma_W^2)$, and then apply Gram-Schmidt orthogonalization to produce fingerprints that are strictly orthogonal to each other with equal energies.

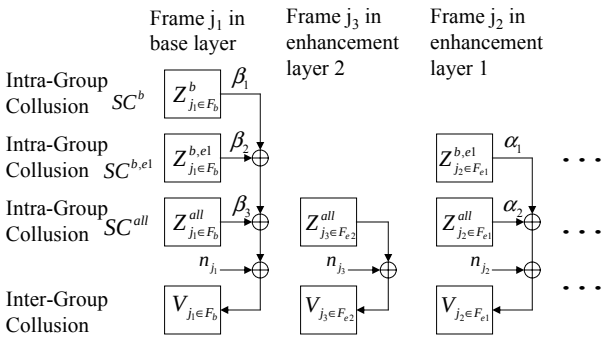


Figure 3.1: Two-stage collusion for scalable-encoded multimedia content

3.1.2.2 Collusion Attacks

During multiuser collusion, attackers collectively mount attacks to effectively remove traces of the embedded fingerprints. Since no one is willing to take a higher risk than the others, an important issue during collusion is to distribute the risk evenly among colluders and achieve fairness of the attack. As studied in [43], given the same amount of noise, for Gaussian fingerprint, the nonlinear attack can be modelled as averaging attack, which is a special case of the optimal attack when all colluders receive the same quality copy. The work in [15] studied how to ensure that all attackers have the same probability of being captured when they receive fingerprinted copies of different quality due to network and device heterogeneity.

Let SC^b be the set with the indices of the colluders who receive the fingerprinted base layer only; $SC^{b,e1}$ contains the indices of all colluders who subscribe to the medium resolution copy; and SC^{all} contains the indices of the colluders who receive all three layers. $K^b = |SC^b|$, $K^{b,e1} = |SC^{b,e1}|$ and $K^{all} = |SC^{all}|$ are the number of colluders in SC^b, $SC^{b,e1}$ and SC^{all}, respectively. $K = K^b + K^{b,e1} + K^{all}$ is the total number of colluders.

Following the two-stage collusion model in [15], colluders first apply intra-group

collusion as shown in Figure 3.1.2.2. For each frame $j \in F_b$ in the base layer, colluders in SC^b generate $\mathbf{Z}_j^b = \sum_{k \in SC^b} \mathbf{X}_j^{(k)} / K^b$; for each frame $j \in F_b \cup F_{e1}$ that they receive, colluders in $SC^{b,e1}$ calculate $\mathbf{Z}_j^{b,e1} = \sum_{k \in SC^{b,e1}} \mathbf{X}_j^{(k)} / K^{b,e1}$; and for every frame $j = F_b \cup F_{e1} \cup F_{e2}$ in the video sequence, the colluders in SC^{all} generate $\mathbf{Z}_j^{all} = \sum_{k \in SC^{all}} \mathbf{X}_j^{(k)} / K^{all}$. Then, colluders combine these three copies, $\{\mathbf{Z}_j^b\}_{j \in F_b}$, $\{\mathbf{Z}_j^{b,e1}\}_{j \in F_b \cup F_{e1}}$, and $\{\mathbf{Z}_j^{all}\}_{j \in F_b \cup F_{e1} \cup F_{e2}}$, and apply inter-group collusion. For each frame $j \in F_b$ in the base layer, the colluded frame is

$$\mathbf{V}_j = \beta_1 \mathbf{Z}_j^b + \beta_2 \mathbf{Z}_j^{b,e1} + (1 - \beta_1 - \beta_2) \mathbf{Z}_j^{all} + \mathbf{n}_j \tag{3.1}$$

where $0 \leq \beta_1, \beta_2, 1 - \beta_1 - \beta_2 \leq 1$. For each frame $j_2 \in F_{e1}$ in the enhancement layer 1, colluders calculate

$$\mathbf{V}_{j_2} = \alpha_1 \mathbf{Z}_{j_2}^{b,e1} + (1 - \alpha_1) \mathbf{Z}_{j_2}^{all} + \mathbf{n}_{j_2} \tag{3.2}$$

where $0 \leq \alpha_1 \leq 1$. For each frame $j_3 \in F_{e2}$ in the enhancement layer 2, the colluded frame j is

$$\mathbf{V}_{j_3} = \mathbf{Z}_{j_3}^{all} + \mathbf{n}_{j_3}. \tag{3.3}$$

$\mathbf{n}_j$ is additive noise to further hinder detection.

During collusion, the colluders seek the collusion parameters α_1, β_1 and β_2 to minimize their risk under the constraint that all colluders have the same probability of being detected. From the above collusion model, the collusion parameters α_j and β_l directly reflect the collusion strategy. And the side information we will discuss in the following sections is the information hidden in the colluded copy that can give detector better estimation of the collusion, and lead to a better detection performance. If the detector is correlation-based, then the mean value of the detection statistics can be used as side

information, which we will show in Section 3.2.

3.1.2.3 Fingerprint Detection and Colluder Identification

We consider a non-blind detection scenario where the host signal is first removed from the test copy before colluder identification. The detector then extracts the fingerprint $\mathbf{Y}_j$ from the j^{th} frame $\mathbf{V}_j$ in the colluded copy. Then, he/she calculates the similarity between the extracted fingerprint $\mathbf{Y}$ and each of the original fingerprints $\{\mathbf{W}^{(i)}\}$, compares with a pre-determined threshold h, and outputs the estimated identities of the colluders $\widehat{SC}$.

To analyze the performance of multimedia fingerprints, we adopt the commonly used criteria in the literature [10]. In order to measure the performance of the fingerprint system under various conditions, such as top-secret scenario in which the fingerprint detector aim to catch as many colluders as possible and the popular commercial scenario in which the non of the innocent user should be falsely accused. Let $P_d^{(i)}$ is the probability of user i being accused as a colluder, we use the following measurements:

- P_d: the probability of capturing at least one colluder. The motivating application of P_d is to provide digital evidence in the court of law. From the analysis in [10], P_d can be formulated as $1-\prod_{i\in SC}(1-P_d^{(i)})$, where SC is the set of the colluders.

- P_{fp}: the probability of accusing at least one innocent user. P_{fp} serves as the probability of false alarm in high-security system. It reflects the confidence of the detector about the accused users–the lower the P_{fp} is, the higher the detection confidence. P_{fp} can be formulated as $1-\prod_{i\notin SC}(1-P_d^{(i)})$.

- $E[F_d]$: the expected fraction of colluders that are successfully captured. When the

digital rights enforcer's concern is to catch as many colluders as possible, $E[F_d]$ is a suitable performance criteria. Mathematically, $E[F_d] = \sum_{i \in SC} P_d^{(i)}/K$, where K is the number of colluders.

- $E[F_{fp}]$: the expected fraction of innocent users that are falsely accused. $E[F_{fp}]$ and $E[F_d]$ are used to show the balance between capturing colluders and placing innocents under suspicion, where $E[F_{fp}] = \sum_{i \notin SC} P_d^{(i)}/(M-K)$. Here, M is the total number of users.

3.2 Analysis of Detector's Strategies with Side Information

This section analyzes how side information about collusion can help improve the collusion resistance and influence the detector's action. We study how to probe side information about collusion from the colluded copy. Consider the scenario where the colluded copy contains all three layers and has the highest quality, and the analysis for other scenarios, such if the colluders only have two layers of the video, is similar. Without loss of generality, we use users in $\mathbf{U}^{all}$ as an example to demonstrate the detection process and analyze the performance. For users in $\mathbf{U}^{b,e1}$ and $\mathbf{U}^{b}$, the colluder identification process and the performance analysis are similar.

3.2.1 Different Fingerprint Detection Strategies

As we discussed in Section 3.1.2.3, when detecting fingerprints, there are many different ways to measure the similarity between the extracted fingerprint $\mathbf{Y}$ and the originally embedded one $\mathbf{W}^{(i)}$.

3.2.1.1 A Collective Fingerprint Detector

The work in [15] considered a simple fingerprint detector that uses fingerprints extracted from all layers collectively to identify colluders. For each user $\mathbf{u}^{(i)}$, the detector first calculates $\breve{F}^{(i)} = F^{(i)} \cap F^c$, where $F^{(i)}$ contains the indices of the frames received by $\mathbf{u}^{(i)}$ and F^c contains the indices of the frames in the colluded copy. Then, the detector calculates

$$TN_c^{(i)} = \left(\sum_{j \in \breve{F}^{(i)}} \langle \mathbf{Y}_j, \mathbf{W}_j^{(i)} \rangle \right) / \sqrt{\sum_{j \in \breve{F}^{(i)}} ||\mathbf{W}_j^{(i)}||^2}, \tag{3.4}$$

where $||\mathbf{W}_j^{(i)}||$ is the Euclidean norm of $\mathbf{W}_j^{(i)}$. Given a pre-determined threshold h, $\widehat{SC}_c = \{i : TN_c^{(i)} > h\}$.

Assume that the colluders choose the parameters $\{\alpha_k, \beta_l\}$ in the same way as in [15]. Without loss of generality, we consider the scenario where the colluders generate a colluded copy of the highest resolution and $F^c = F_b \cup F_{e1} \cup F_{e2}$ [48]. With orthogonal fingerprint modulation as in Section 3.1.2.1, under the assumption that the detection noises are i.i.d. and follow Gaussian distribution $\mathcal{N}(0, \sigma_n^2)$, the detection statistics $\{TN_c^{(i)}\}$ in

(3.4) are independent Gaussian with marginal distribution

$$TN_c^{(i)} \sim \begin{cases} \mathcal{N}\left(\mu_c^{(i)}, \sigma_n^2\right), & \text{if } i \in SC, \\ \mathcal{N}\left(0, \sigma_n^2\right), & \text{if } i \notin SC, \end{cases}$$
$$\text{where} \quad \mu_c^{(i)} = \frac{(1-\beta_1-\beta_2)N_b + (1-\alpha_1)N_{e1} + N_{e2}}{K^{all}\sqrt{N_b + N_{e1} + N_{e2}}}\sigma_W^2.$$

N_b, N_{e1} and N_{e2} are the lengths of the fingerprints embedded in the base layer, enhancement layer 1 and enhancement layer 2, respectively. For a given user $\mathbf{u}^{(i)}$, define $P_s^{(i)}$ as the probability of successfully capturing him/her if he/she is guilty, and $P_{fa}^{(i)}$ is the probability of falsely accusing him/her if he/she is innocent. With the detector in (3.4), we have

$$\begin{aligned} P_s^{(i)} &= Q\left(\frac{h-\mu_c^{(i)}}{\sigma_n}\right) \text{ if } \quad i \in SC, \\ \text{and} \quad P_{fa}^{(i)} &= Q\left(\frac{h}{\sigma_n}\right) \quad \text{if} \quad i \notin SC, \end{aligned} \tag{3.5}$$

where $Q(\cdot)$ is the Gaussian tail function. Therefore, we have

$$\begin{aligned} P_d &= P\left[\max_{i\in SC} TN_c^{(i)} > h\right] = 1 - \prod_{i\in SC} P_s^{(i)} = 1 - \left[1 - Q\left(\frac{h-\mu_c^{(i)}}{\sigma_n}\right)\right]^K, \\ P_{fp} &= P\left[\max_{i\notin SC} TN_c^{(i)} > h\right] = 1 - \prod_{i\notin SC} (1 - P_{fa}^{(i)}) = 1 - \left[1 - Q\left(\frac{h}{\sigma_n}\right)\right]^{M-K}, \\ E[F_d] &= \sum_{i\in SC} P[TN_c^{(i)} > h]/K = \sum_{i\in SC} P_s^{(i)}/K = Q\left(\frac{h-\mu_c^{(i)}}{\sigma_n}\right), \\ \text{and} \quad E[F_{fp}] &= \sum_{i\notin SC} P[TN_c^{(i)} > h]/(M-K) = \sum_{i\notin SC} P_{fa}^{(i)}/(M-K) = Q\left(\frac{h}{\sigma_n}\right). \end{aligned} \tag{3.6}$$

Assuming that the fingerprint detector will always use (3.5) and fingerprints extracted from all layers collectively to determine if $\mathbf{u}^{(i)}$ participates in collusion, the work in [15] studied how the colluders should select the parameters α_1, β_1 and β_2 such that

$\{P_s^{(i)}\}$ are the same for all colluders $i \in SC$ and will be compared with the results in Section 3.3.

3.2.1.2 Fingerprint Detection at Each Individual Layer

Given $\mathbf{Y}_{e2}$, $\mathbf{Y}_{e1}$ and $\mathbf{Y}_b$ which are the fingerprints extracted from the enhancement layer 2, enhancement layer 1 and the base layer, respectively, in addition to the collective detector (3.4) in Section 3.2.1.1, the digital rights enforcer can also examine $\mathbf{Y}_{e2}$, $\mathbf{Y}_{e1}$ and $\mathbf{Y}_b$ independently and use the detection results at each individual layer to estimate the colluders' identities. Therefore, in addition to the collective detector, the digital rights enforcer can also use *detectors at base layer, enhancement layer1, and enhancement layer 2*. To demonstrate this colluder identification process and analyze its performance, we use users in $\mathbf{U}^{all}$ who receive all three layers as an example. The analysis for users in $\mathbf{U}^{b,e1}$ and $\mathbf{U}^{b}$ is similar and thus omitted.

Let F_t be the set of indices of the frames in layer t in which $t = b, e1, e2$ represents base layer, enhancement layer 1, and enhancement layer 2, respectively. For user $\mathbf{u}^i \in \mathbf{U}^{all}$ who receive all three layers from the content owner, given $\{\mathbf{Y}_j\}_{j\in F_t}$, the fingerprints from layer t of the colluded copy, the detector at layer t calculates the detection statistics

$$TN_t^{(i)} = \left(\sum_{j\in F_t} \langle \mathbf{Y}_j, \mathbf{W}_j^{(i)} \rangle \right) / \sqrt{\sum_{j\in F_t} ||\mathbf{W}_j^{(i)}||^2} \tag{3.7}$$

to measure the similarity between the extracted fingerprint and the originally embedded fingerprint. The detector at layer t accused $u^{(i)}$ as a colluder if $TN_t^{(i)} > h$, and sets $i \in \widehat{SC}$, which is the suspicious-colluder set. Here, h here is a predetermined threshold.

The analysis of the detection statistics $TN_t^{(i)}$ in (3.8) is similar to that of $TN^{(i)}$ in

(3.4). If the detection noises are i.i.d. and follow Gaussian distribution $\mathcal{N}(0,\sigma_n^2)$, for user $\mathbf{u}^{(i)} \in \mathbf{U}^{all}$, $TN_t^{(i)}$ are independent Gaussian with marginal distribution

$$TN_t^{(i)} \sim \begin{cases} \mathcal{N}(\mu_t^{(i)},\sigma_n^2) & \text{if} \quad i \in SC, \\ \mathcal{N}(0,\sigma_n^2) & \text{if} \quad i \notin SC, \end{cases}$$

$$\text{where} \qquad \mu_b^{(i)} = (1-\beta_1-\beta_2)\frac{\sqrt{N_b}}{K^{all}}\sigma_W,$$

$$\mu_{e1}^{(i)} = (1-\alpha_1)\frac{\sqrt{N_{e1}}}{K^{all}}\sigma_W, \quad \text{and } \mu_{e2}^{(i)} = \frac{\sqrt{N_{e2}}}{K^{all}}\sigma_W. \tag{3.8}$$

Therefore, for user $\mathbf{u}^{(i)} \in \mathbf{U}^{all}$, the probability of successfully capturing him/her if he/she is guilty is

$$P_s^{(i)} = Q\left(\frac{h-\mu_t^{(i)}}{\sigma_n}\right), \tag{3.9}$$

and the probability of falsely accusing him/her if he/she is innocent is

$$P_{fa}^{(i)} = Q\left(\frac{h}{\sigma_n}\right). \tag{3.10}$$

The analysis of P_d, P_{fp}, $E[F_d]$ and $E[f_{fp}]$ is the same as that in Section 3.2.1.1 and not repeated. It is clear from (3.9) and (3.8) that the higher the $\mu_t^{(i)}$ is, the better the traitor-tracing performance.

3.2.2 Performance Comparison

This section compares the performance of the four detection statistics (3.4) and (3.8) when identifying colluders in SC^{all}. From the above analysis, for a given h and a fixed P_{fp}, comparing P_d of different detection statistics is equivalent to comparing their means.

For a colluder $i \in SC^{all}$, Figure 3.2 shows an example of the means of the detection statistics in (3.4) and (3.8). In Figure 3.2, we first generate independent vectors follow-

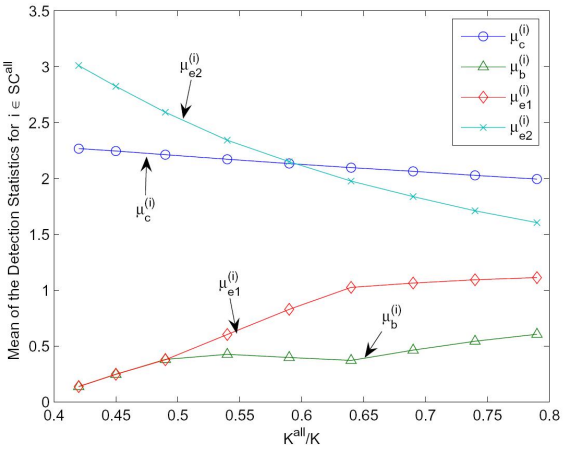


Figure 3.2: Comparison of μ_c in (3.5), $\mu_{e2}^{(i)}$, $\mu_{e1}^{(i)}$, and $\mu_b^{(i)}$ in (3.9) for $i \in SC^{all}$. $(N_b, N_{e1}, N_{e2}) = (50000, 50000, 100000)$. $K = 250$ and $K^b = 50$. Each point on the X axis corresponds to a unique triplet (K^b, K^{e1}, K^{e2}). $F^c = F_b \cup F_{e1} \cup F_{e2}$.

ing Gaussian distribution $\mathcal{N}(0,1)$, and then apply Gram-Schmidt orthogonalization to generate orthogonal fingerprints for different users. The lengths of the fingerprints embedded in the base layer, enhancement layer 1 and enhancement layer 2 are $N_b = 50000$, $N_{e1} = 50000$ and $N_{e2} = 100000$, respectively. In Figure 3.2, we fix the total number of colluders $K = 250$, and $K^b = 50$ of them receive the fingerprinted base layer only. Each point on the X axis corresponds to a unique triplet (K^b, K^{e1}, K^{e2}). The colluders follow the work in [15] to select the collusion parameters and generate a colluded copy with all three layers under the fairness constraints.

From Figure 3.2, $TN_c^{(i)}$ in (3.4) has the best performance when more than 60% of the colluders receive a high-quality copy with all three layers. This is because in this scenario, $\mathbf{u}^{(i)}$'s fingerprints are spread all over the entire colluded copy $\mathbf{V}$, and $\mathbf{W}^{(i)}$'s energy is evenly distributed in the three layers of $\mathbf{V}$. Therefore, from detection theory [49], fin-

gerprints extracted from all layers should be used during detection to improve the performance. When $K^{all}/K < 0.6$, due to the selection of the collusion parameters, a significant portion of $\mathbf{W}^{(i)}$'s energy is in the enhancement layer 2, while the other two layers of the colluded copy contain little information of $\mathbf{u}^{(i)}$'s identity. Thus, in this scenario, $TN_{e2}^{(i)}$ in (3.8) gives the best detection performance. Also, since larger K^{all} introduces smaller fingerprint energy in enhancement layer 2 for SC^{all}, and the total number of colluders remains the same, thus smaller K^b and $K^{b,e1}$ result in higher fingerprint energy for SC^b and SCb,e in base layer and enhancement layer 1. Therefore, α_1, β_1, β_2 must be lower to ensure equal probability of being detected for every user. Hence μ_{e1} and μ_b for SC^{all} may increase as K^{all} increases.

3.2.3 Colluder Identification with Side Information

For the four detection statistics in Section 3.2.1, their traitor tracing capability is determined by their *statistical means*. The larger the statistical mean is, the better the performance. Note that from the above analysis, the collusion parameters ($\{\alpha_j\}$ and $\{\beta_l\}$ in the two-stage collusion model) determine the means of the detection statistics. Thus, if *side information about the statistical means* of different detection statistics (or equivalently, the collusion parameters) is available to the fingerprint detector, he/she should select the detection statistics that has the largest statistical mean to improve the traitor-tracing capability.

During the fingerprint detection and colluder identification process, the fingerprint detector should first examine the colluded copy and probe such side information, then

select the best detection statistics and identify colluders. As an example, to identify colluders who receive all three layers, the key steps in probing the means of the detection statistics and selecting the optimum detection statistics are as follows:

- For every user $\mathbf{u}^{(i)}$ in $\mathbf{U}^{all}$, the detector first calculates $TN_c^{(i)}$, $TN_{e2}^{(i)}$, $TN_{e1}^{(i)}$ and $TN_b^{(i)}$ as in Section 3.2.1, and obtains

$$\begin{aligned} \widehat{SC}_c^{all} &= \{i : TN_c^{(i)} > h_t\}, \quad \widehat{SC}_{e2}^{all} = \{i : TN_{e2}^{(i)} > h_t\}, \\ \widehat{SC}_{e1}^{all} &= \{i : TN_{e1}^{(i)} > h_t\}, \quad \text{and} \\ \widehat{SC}_b^{all} &= \{i : TN_b^{(i)} > h_t\} \end{aligned} \tag{3.11}$$

  for a given h_t.

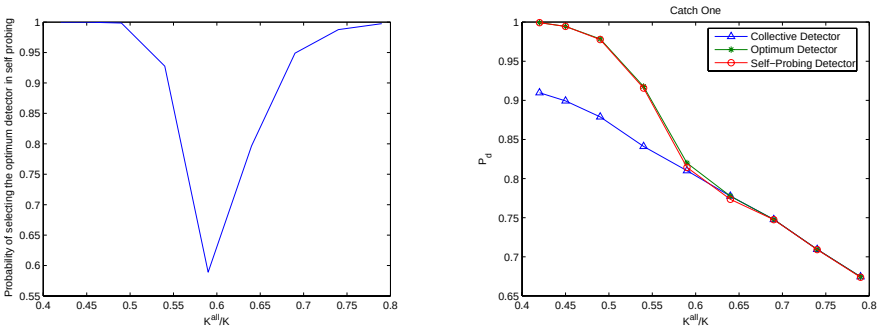
- The detector combines the above four sets of estimated colluders in $\mathbf{U}^{all}$ and lets $\widehat{SC}^{all} = \widehat{SC}_c^{all} \cup \widehat{SC}_{e2}^{all} \cup \widehat{SC}_{e1}^{all} \cup \widehat{SC}_b^{all}$.

- Given $\widehat{SC}^{all}$, the detector estimates the means of the four detection statistics in Section 3.2.1

$$\begin{aligned} \hat{\mu}_c &= \sum_{k \in \widehat{SC}^{all}} \frac{TN_c^{(k)}}{|\widehat{SC}^{all}|}, \quad \hat{\mu}_{e2} = \sum_{k \in \widehat{SC}^{all}} \frac{TN_{e2}^{(k)}}{|\widehat{SC}^{all}|}, \\ \hat{\mu}_{e1} &= \sum_{k \in \widehat{SC}^{all}} \frac{TN_{e1}^{(k)}}{|\widehat{SC}^{all}|}, \quad \hat{\mu}_b = \sum_{k \in \widehat{SC}^{all}} \frac{TN_b^{(k)}}{|\widehat{SC}^{all}|}. \end{aligned} \tag{3.12}$$

- The detector compares $\hat{\mu}_c$, $\hat{\mu}_{e2}$, $\hat{\mu}_{e1}$ and $\hat{\mu}_b$ and selects the detection statistics with the largest estimated mean. For example, the collective detector in (3.4) is chosen if $\hat{\mu}_c$ has the largest value.

When identifying colluders in $SC^{b,e1}$, the side information probing process is similar and



(a) (b)

Figure 3.3: Performance of the self-probing fingerprint detector for the example in Figure 3.2. (a) Probability of selecting the optimum detection statistics when identifying colluders in $\mathbf{U}^{all}$. (b) P_d of the collective detector, the optimum detector with perfect knowledge of the detection statistics' means, and the self-probing detector that probes the side information itself. h_t is chosen to let $P_{fa}^{(i)} = 10^{-2}$ for an innocent user $i \notin SC$. $P_{fp} = 10^{-3}$. The result is based on 10000 simulation runs.

not repeated. Then, the fingerprint detector follows Section 3.2.1 and estimates the identities of the colluders.

3.2.4 Performance Analysis and Simulation Results

In our simulations, we simulate three different fingerprint detectors: the simple collective detector in (3.4); the optimum detector with perfect knowledge of the statistical means of the four detection statistics; and the self-probing detector, which first uses the algorithm in Section 3.2.3 to select the best detection statistics and then follows Section 3.2.1 to identify colluders.

The simulation setup is the same as that in Figure 3.2. We choose the parameters

based on the analysis in [15], which shows the total number of 250 colluders in a 750-user system is large enough to effectively reduce the fingerprint energy and reduce the probability of each colluder to be accused to around 10%, in which the fingerprint system can barely provide protection. Hence under such tough scenario, we would test whether the proposed self-probing detector can provide better collusion resistance.

There are a total of $K = 250$ colluders, and $K^b = 50$ of them receive the fingerprinted base layer only. Each point on the X axis in Figure 3.3 corresponds to a unique triplet $(K^b, K^{b,e1}, K^{all})$. The colluders select $\{\alpha_k, \beta_l\}$ in the same way as in [15] and generate a colluded copy with all three layers. For each frame j in the colluded copy, we adjust the power of the additive noise such that $||\mathbf{n}_j||^2 = ||\mathbf{W}_j^{(i)}||^2$. Other values give the same trend.

Figure 3.3 (a) plots the probability that the proposed probing algorithm in Section 3.2.3 selects the optimum detection statistics when identifying colluders in $\mathbf{U}^{all}$. In the example in Figure 3.2, we only choose between $TN^{(i)}$ and $TN_{e2}^{(i)}$ since $TN_{e1}^{(i)}$ and $TN_b^{(i)}$ never outperform the other two. From Figure 3.3 (a), the proposed probing algorithm selects the optimum detection statistics with probability 0.6 when $K^{all}/K \approx 0.6$; while in other scenarios, the detector always picks the best detection statistics. Note that from Figure 3.2, when $K^{all}/K \approx 0.6$, μ_c and $\mu_{e2}^{(i)}$ have similar values and, therefore, $TN^{(i)}$ and $TN_{e2}^{(i)}$ have approximately the same performance. Consequently, in this scenario, choosing the sub-optimum detection statistics does not significantly deteriorate the detection performance. When μ_c and $\mu_{e2}^{(i)}$ differ significantly from each other, the self-probing detector always chooses the optimal detection statistics when identifying colluders in $\mathbf{U}^{all}$.

To evaluate the traitor-tracing performance of the proposed colluder identification algorithm with side information, we consider the catch one scenario, where the fingerprint

detector aims to capture at least one colluder without falsely accusing any innocents. In this scenario, the criteria used to measure the performance is P_d and P_{fp}. The analysis for other scenarios using other performance criteria is similar and gives the same trend. For a fixed $P_{fp} = 10^{-3}$, Figure 3.3 (b) shows P_d of the three detectors. From Figure 3.3 (b), utilizing side information about the means of different detection statistics can help the fingerprint detector significantly improve its performance, especially when K^{all}/K is small and the colluders' fingerprints are not evenly distributed in the three layers of the colluded copy. Furthermore, from Figure 3.3 (b), when the difference between μ_c and $\mu_{e2}^{(i)}$ is large, the side information probing algorithm in Section 3.2.3 helps the detector choose the best detection statistics and achieve the optimal performance. When μ_c and $\mu_{e2}^{(i)}$ are approximately the same, the performance of the self-probing fingerprint detector is almost the same as that of the optimal detector with perfect knowledge of the means of the detection statistics, and the difference between these two is no larger than 0.005 and can be ignored.

3.2.5 Impact of Side Information on Fairness of Multi-user Collusion

Without probing side information, the detector will always use all the frames collectively to identify the colluders, hoping that more frames will give him/her more information about colluders' identities. On the other side, colluders adjust the collusion parameters $\{\alpha_j\}$ and $\{\beta_l\}$ to seek the collective fairness. Under such circumstances, the colluders and the fingerprint detector reaches the *collective fairness equilibrium*. However, side information about collusion not only improves the fingerprint detector's performance, it

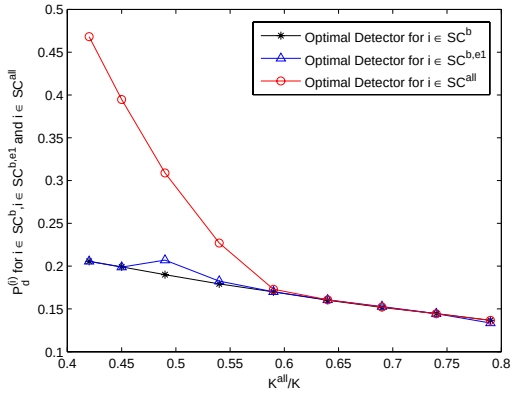


Figure 3.4: Each colluder's probability of being detected ($P_s^{(i)}$) with the self-probing fingerprint detector. The simulation setup is the same as that in Figure 3.3, and colluders follow [15] when selecting the collusion parameters $\{\alpha_k\}$ and $\{\beta_l\}$. The threshold h is selected to satisfy $P_{fp} = 10^{-3}$. The results are based on 10000 simulation runs.

also affects each colluder's probability of being detected and influences how they collude [50]. Thus, side information breaks the collective fairness equilibrium between the colluders and the fingerprint detector, and both sides need to search for a new equilibrium.

To demonstrate how side information breaks the collective fairness equilibrium, Figure 3.4 shows each colluder's probability of being detected with the self-probing fingerprint detector. The simulation setup is the same as that in Figure 3.3. In Figure 3.4, colluders follow [15] to select the collusion parameters $\{\alpha_j\}$ and $\{\beta_l\}$ during the two-stage collusion, and we adjust the power of the additive noise such that $||\mathbf{n}_j||^2 = ||\mathbf{W}_j^{(i)}||^2$ for each frame in the video sequence. From Figure 3.4, when $K^{all}/K < 0.6$, those colluders who receive all three layers have a much larger probability of being detected than the others. In this example, during collusion, attackers only consider the collective detector in (3.5), and they select the parameters $\{\alpha_j\}$ and $\{\beta_l\}$ such that $\{TN_c^{(i)}\}$ in (3.5)

have the same statistical mean for all attackers. However, during the colluder identification process, the fingerprint detector considers all possible detection strategies in Section 3.2.1, probes side information about detection statistics, and uses the one that gives the best collusion resistance. Therefore, with the self-probing fingerprint detector in Section 3.2.3, colluders have to find a new set of collusion parameters to ensure the equal risk of all attackers.

3.3 Equilibrium of the Colluder-Detector Game With Side Information

In this section, we will model the behavior dynamics with side information between the two group of users in the multimedia fingerprinting social network as a two-person two-stage game. We formulate the equilibrium of this colluder-detector game as a min-max problem and find the optimal strategy of all users in the social network.

3.3.1 Game-Theoretical Modelling of Colluder-Detector Dynamics

In the multimedia fingerprint social network, different members have different goals and utilities: the colluders mount attacks to generate the colluded copy for redistribution, and the forensic detector try to identify the colluders from the redistributed colluded copy. The colluders gain rewards by redistributing the colluded content and they take the risk to be caught by the digital rights enforcer. In this game, the colluders' gain is the detector's loss, thus the two group of members in the fingerprinting social network have totally

conflicting objectives.

Stackelberg Game Model

To capture users' behavior in strategic situations, in which an individual's success in making choices depends on the choices of others, game theory [7], [8] is a useful tool to model the complex dynamics among multimedia social network members. Hence, to analyze the optimal strategies of both fingerprint detector and the colluders, we formulate the interaction between the two groups of social network members as a game with two players: the colluders acting as one single player and the fingerprint detector as the other.

- **Players**: There are two players: colluders who make decision first as the *leader*, followed by the fingerprint detector who apply detection as a *follower*.

- **Payoff Function Definition**: To analyze the dynamic between colluders and the forensic detector, we assume all the colluders have the same objectives and agree to share the same risk and reward. Therefore, during the fair collusion, every colluder has the same goal of minimizing his/her risk of being detected $P_s^{(i)}$ under the constraint that $\{P_s^{(i)}\}$ are the same for all colluders. Thus, a natural definition of colluder i's payoff function is $\pi^C = 1 - P_s^{(i)}$, the probability that each colluder successfully removes traces of his/her fingerprint during collusion. From the detector's point of view, the colluders' gain is the loss of the digital rights enforcer, so we can define the detector's payoff as $\pi^D = -\pi^C$.

- **Colluders' Strategies**: Each set of the collusion parameters $\{\alpha_1, \beta_1, \beta_2\}$ that achieves equal probability of detection for each colluder leads to one strategy for the colluders in the colluder-detector game.

- **Detector's Strategies**: As discussed in Section 3.1.2.3, the detector's strategies include the collective detector, single-layer detector, and the self-probing detector. We assume the detector can probe the side information (the mean of the detection statistics) when he/she chooses the strategy.

In this game, there are multiple detection statistics that the fingerprint detector can use to identify colluders. However, by the analysis and simulation results shown in Section 3.2.3, the self-probing detector can always achieve better or equal performance as all other detectors (collective detector and single-layer detector). Thus, to maximize his/her payoff, the fingerprint detector always probes side information about collusion and selects the detection statistics that has the largest chance of successfully capturing colluders. From the angle of game theoretical analysis, probing side-information is equivalent to observing the colluders' action. This scenario implies that the detector (follower in this game) can observe the colluders' action, and the colluders (leader) know that the detector observes their action. Hence, colluders as the leader have perfect knowledge of the detection strategies that the fingerprint detector will use, because the detector has no incentive to deviate from the self-probing detector. Therefore, the detector has no means of committing to a follower action that deviates from the self-probing detector which is the best response, and the colluders know this. Therefore, the colluder-detector game is a Stackelberg game [8] with perfect information.

Equilibrium Analysis As shown in Figure 3.4, with side information available to the fingerprint detector, the selected collusion parameters in [15] cannot guarantee the fairness of collusion. Therefore, the colluders need to find new sets of collusion

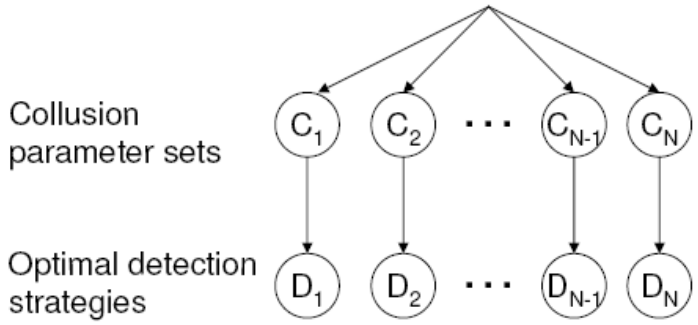


Figure 3.5: Game tree illustration of the colluder-detector dynamics. C_1, C_2, $\cdots$, C_N are the N possible sets of collusion parameters that achieve absolute fairness when the fingerprint detector uses the optimal detection statistics to identify colluders; while D_1, D_2, $\cdots$, D_N are the corresponding optimal fingerprint detection strategies. For the example of $(K^b, K^{b,e1}, K^{all}) = (50, 25, 175)$ in Section 3.3.5, N=3,C_1 set of parameters satisfies (3.45), C_2set of parameters satisfies (3.46), and C_3 set of parameters satisfies (3.47). In D_1, the fingerprint detector uses $TN_b^{(i)}$ for $i \in \mathbf{U}^{b,e1}$ and $TN_c^{(j)}$ for $j \in \mathbf{U}^{all}$. In D_2, the fingerprint detector uses $TN_{e1}^{(i)}$ for $i \in \mathbf{U}^{b,e1}$ and $TN_c^{(j)}$ for $j \in \mathbf{U}^{all}$. In D_3, the fingerprint detector uses $TN_c^{(i)}$ for $i \in \mathbf{U}^{b,e1}$ and $TN_c^{(j)}$ for $j \in \mathbf{U}^{all}$.
parameters to achieve fairness.

With the proposed self-probing fingerprint detection process in Section 3.2.3, for every type of collusion, the fingerprint detector will always choose the detection statistics that gives the best traitor-tracing performance which can be illustrated as the game tree shown in Figure 3.5. In this game, assuming that there are N possible collusion strategies under the fairness constraint, the colluders first choose the collusion strategy, and then the fingerprint detector selects the optimal detection statistics.

Since the follower (detector) can observe the leader's (colluders') strategy, the game

model can be solved by backward induction. The backward induction starts from the last stage of the game, which is the detector's strategy. As shown in Section 3.2.3, the self-probing detector is the optimal strategy for all the fair collusion. Hence we can move forward to the previous stage in the game, which is the colludes' strategy. Since both the colluders and the fingerprint detector know that the optimal detection statistics will be used to identify colluders, once attackers determine the collusion strategy, their payoff is fixed and the colluders can accurately estimate their payoff. The colluders consider what the best response of the detector is, i.e., how the detector will respond once he/she observes the leader's strategy. The colluders then pick a strategy that maximizes its payoff, anticipating the predicted response of the detector. Hence, during collusion, colluders should consider the worst case scenario where the fingerprint detector always makes the right decision when selecting which detection statistics to use. They select the collusion parameters to minimize their risk under the constraint that all colluders have the same probability of being detected. Thus, the equilibrium of this game can be modelled as a *min-max problem*.

As we discussed in Section 3.2.5, without side information, the colluders and the detector achieve the collective fairness equilibrium: the fingerprint detector uses the collective detection statistics in (3.4), and the colluders select the collusion parameter as in [15] to ensure the same risk under the collective detector. Probing and utilizing side information moves the equilibrium of the colluder-detector game from the collective one to the min-max solution as discussed in Section 3.3.3.

3.3.2 Min-Max Problem Formulation of the Equilibrium

For each user $\mathbf{u}^{(i)}$, define $\mathfrak{D}^{(i)}$ as the set including all possible detection statistics that can be used to measure the similarity between the extracted fingerprint $\mathbf{Y}$ and $\mathbf{u}^{(i)}$'s fingerprint $\mathbf{W}^{(i)}$. For example, $\mathfrak{D}^{(i)} = \{TN_c^{(i)},\ TN_{e2}^{(i)},\ TN_{e1}^{(i)},\ TN_b^{(i)}\}$ for a colluder $i \in SC^{all}$ who receives all three layers, while $\mathfrak{D}^{(i)} = \{TN_c^{(i)},\ TN_{e1}^{(i)},\ TN_b^{(i)}\}$ for user $i \in SC^{b,e1}$ who receives a medium resolution copy. Define $P_s^{(i)}\left(\mathfrak{D}^{(i)}, \{\alpha_k, \beta_l\}\right)$ as the probability that colluder $\mathbf{u}^{(i \in SC)}$ is captured by the digital rights enforcer.

Consequently, we can model the problem as a min-max problem:

$$\begin{aligned}
&\min_{\{\alpha_k,\beta_l\}} \max_{\mathfrak{D}^{(i)}} P_s^{(i)}\left(\mathfrak{D}^{(i)}, \{\alpha_k, \beta_l\}\right) \\
&\text{s.t.} \max_{\mathfrak{D}^{(i_1)}} P_s^{(i_1)}\left(\mathfrak{D}^{(i_1)}, \{\alpha_k, \beta_l\}\right) = \max_{\mathfrak{D}^{(i_2)}} P_s^{(i_2)}\left(\mathfrak{D}^{(i_2)}, \{\alpha_k, \beta_l\}\right), \quad \forall i_1, i_2 \in SC
\end{aligned} \tag{3.13}$$

From the analysis in the previous section, for a given threshold h and fixed σ_n^2, $P_s^{(i)}$ is determined by the mean of the detection statistics that are used. Therefore, for colluder $i_1 \in SC^b$, $i_2 \in SC^{b,e1}$ and $i_3 \in SC^{all}$, (3.13) can be simplified to

$$\begin{aligned}
&\min_{\{\alpha_k,\beta_l\}} \mu = \mu_{max}^{(i_1)} = \mu_{max}^{(i_2)} = \mu_{max}^{(i_3)}, \\
&\text{s.t.} \quad 0 \le \alpha_k \le 1,\ 0 \le \beta_l \le 1, \\
&\text{where} \quad \mu_{max}^{(i_1)} = \mu_c^{(i_1)}, \\
&\qquad\qquad \mu_{max}^{(i_2)} = \max\{\mu_b^{(i_2)}, \mu_{e1}^{(i_2)}, \mu_c^{(i_2)}\}, \\
&\text{and } \mu_{max}^{(i_3)} = \max\{\mu_b^{(i_3)}, \mu_{e1}^{(i_3)}, \mu_{e2}^{(i_3)}, \mu_c^{(i_3)}\}.
\end{aligned} \tag{3.14}$$

In (3.14),

$$\begin{aligned}
\mu_c^{(i_1)} &= \frac{\beta_1\sqrt{N_b}}{K^b}\sigma_W, \\
\mu_b^{(i_2)} &= \frac{\beta_2\sqrt{N_b}}{K^{b,e1}}\sigma_W, \quad \mu_{e1}^{(i_2)} = \frac{\alpha_1\sqrt{N_{e1}}}{K^{b,e1}}\sigma_W, \quad \mu_c^{(i_2)} = \frac{\beta_2 N_b + \alpha_1 N_{e1}}{K^{b,e1}\sqrt{N_b + N_{e1}}}\sigma_W,
\end{aligned}$$

$$\mu_b^{(i_3)} = \frac{(1-\beta_1-\beta_2)\sqrt{N_b}}{K^{all}}\sigma_W, \quad \mu_{e1}^{(i_3)} = \frac{(1-\alpha_1)\sqrt{N_{e1}}}{K^{all}}\sigma_W, \quad \mu_{e2}^{(i_3)} = \frac{\sqrt{N_{e2}}}{K^{all}}\sigma_W,$$
$$\text{and} \quad \mu_c^{(i_3)} = \frac{(1-\beta_1-\beta_2)N_b + (1-\alpha_1)N_{e1} + N_{e2}}{K^{all}\sqrt{N_b+N_{e1}+N_{e2}}}\sigma_W \tag{3.15}$$

from the analysis in Section 3.2.1.

Given $(K^b, K^{b,e1}, K^{all})$ and (N_b, N_{e1}, N_{e2}), for colluder $i_1 \in SC^b$, $i_2 \in SC^{b,e1}$ and $i_3 \in SC^{all}$ who receive fingerprinted copies of different resolutions, they first find all possible sets of collusion parameters $\{\alpha_k, \beta_l\}$ that satisfy $\mu_{max}^{(i_1)} = \mu_{max}^{(i_2)} = \mu_{max}^{(i_3)}$. Then, they select the one that gives them the minimum risk of being detected.

3.3.3 Analysis of $\mu_{max}^{(i)}$

To solve the problem of (3.14), we first need to analyze $\mu_{max}^{(i)}$ for each colluder $\mathbf{u}^{(i)}$ and study which detection statistics have the maximum mean under which condition.

For Colluder $i \in SC^{b,e1}$

For colluder $i \in SC^{b,e1}$ who receives a medium resolution copy, there are three possibilities: $\mu_{max}^{(i)} = \mu_b^{(i)}$, $\mu_{max}^{(i)} = \mu_{e1}^{(i)}$ and $\mu_{max}^{(i)} = \mu_c^{(i)}$.

$\mu_{max}^{(i)} = \mu_b^{(i)}$**:** If $\mu_{max}^{(i)} = \mu_b^{(i)}$, then $\mu_b^{(i)} \geq \mu_{e1}^{(i)}$ and $\mu_b^{(i)} \geq \mu_c^{(i)}$. Thus, from (3.15),

$$\mu_b^{(i)} \geq \mu_{e1}^{(i)} \Leftrightarrow \frac{\beta_2\sqrt{N_b}}{K^{b,e1}}\sigma_W \geq \frac{\alpha_1\sqrt{N_{e1}}}{K^{b,e1}}\sigma_W \Leftrightarrow \beta_2 \geq \frac{\alpha_1\sqrt{N_{e1}}}{\sqrt{N_b}}. \tag{3.16}$$

Similarly, we have

$$\mu_b^{(i)} \geq \mu_c^{(i)} \Leftrightarrow \frac{\beta_2\sqrt{N_b}}{K^{b,e1}}\sigma_W \geq \frac{\beta_2 N_b + \alpha_1 N_{e1}}{K^{b,e1}\sqrt{N_b+N_{e1}}}\sigma_W \Leftrightarrow \beta_2 \geq \frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})} \tag{3.17}$$

Note that $\sqrt{N_b}+\sqrt{N_{e1}} \geq \sqrt{N_b+N_{e1}}$. Thus $\sqrt{N_{e1}} \geq \sqrt{N_b+N_{e1}}-\sqrt{N_b}$ and $\frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})} \geq \frac{\alpha_1\sqrt{N_{e1}}}{\sqrt{N_b}}$. Therefore, combining (3.16) and (3.17), for colluder $i \in SC^{b,e1}$,

$$\mu_{max}^{(i)} = \mu_b^{(i)} \quad \text{if and only if} \quad \beta_2 \geq \frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}. \tag{3.18}$$

$\mu_{max}^{(i)} = \mu_{e1}^{(i)}$**:** In this case, $\mu_{e1}^{(i)} \geq \mu_b^{(i)}$ and $\mu_{e1}^{(i)} \geq \mu_c^{(i)}$. Thus,

$$\begin{aligned} & \mu_{e1}^{(i)} \geq \mu_b^{(i)} \quad \Leftrightarrow \quad \frac{\alpha_1\sqrt{N_{e1}}}{K^{b,e1}}\sigma_W \geq \frac{\beta_2\sqrt{N_b}}{K^{b,e1}}\sigma_W \Leftrightarrow \beta_2 \leq \frac{\alpha_1\sqrt{N_{e1}}}{\sqrt{N_b}}, \\ \text{and} \quad & \mu_{e1}^{(i)} \geq \mu_c^{(i)} \quad \Leftrightarrow \quad \frac{\alpha_1\sqrt{N_{e1}}}{K^{b,e1}}\sigma_W \geq \frac{\beta_2 N_b + \alpha_1 N_{e1}}{K^{b,e1}\sqrt{N_b+N_{e1}}}\sigma_W \\ & \qquad\qquad \Leftrightarrow \quad \beta_2 \leq \frac{\alpha_1\sqrt{N_{e1}}(\sqrt{N_b+N_{e1}}-\sqrt{N_{e1}})}{N_b}. \end{aligned} \tag{3.19}$$

It is straightforward to show that $\sqrt{N_b+N_{e1}} - \sqrt{N_{e1}} \leq \sqrt{N_b}$ and $\frac{\alpha_1\sqrt{N_{e1}}(\sqrt{N_b+N_{e1}}-\sqrt{N_{e1}})}{N_b} \geq \frac{\alpha_1\sqrt{N_{e1}}}{\sqrt{N_b}}$. Thus, combining the results in (3.19), we have

$$\mu_{max}^{(i)} = \mu_{e1}^{(i)} \quad \text{if and only if} \quad \beta_2 \leq \frac{\alpha_1\sqrt{N_{e1}}(\sqrt{(N_b+N_{e1})}-\sqrt{N_{e1}})}{N_b}. \tag{3.20}$$

$\mu_{max}^{(i)} = \mu_c^{(i)}$**:** This scenario happens if $\mu_c^{(i)} \geq \mu_b^{(i)}$ and $\mu_c^{(i)} \geq \mu_{e1}^{i)}$. Following the same analysis as in the previous two scenarios,

$$\begin{aligned} & \mu_c^{(i)} \geq \mu_b^{(i)} \quad \Leftrightarrow \quad \beta_2 \leq \frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}, \\ \text{and} \quad & \mu_c^{(i)} \geq \mu_{e1}^{(i)} \quad \Leftrightarrow \quad \beta_2 \geq \frac{\alpha_1\sqrt{N_{e1}}(\sqrt{N_b+N_{e1}}-\sqrt{N_{e1}})}{N_b}. \end{aligned} \tag{3.21}$$

Note that $\sqrt{N_b+N_{e1}} \leq \sqrt{N_b} + \sqrt{N_{e1}}$. Therefore, we have

$$\begin{aligned} & \sqrt{N_b+N_{e1}} - \sqrt{N_b} \leq \sqrt{N_{e1}} \quad \text{and} \quad \sqrt{N_b+N_{e1}} - \sqrt{N_{e1}} \leq \sqrt{N_b} \\ \Leftrightarrow \quad & \frac{\sqrt{N_b+N_{e1}}-\sqrt{N_{e1}}}{\sqrt{N_b}} \leq \frac{\sqrt{N_{e1}}}{\sqrt{N_b+N_{e1}}-\sqrt{N_b}} \\ \Leftrightarrow \quad & \frac{\alpha_1\sqrt{N_{e1}}(\sqrt{N_b+N_{e1}}-\sqrt{N_{e1}})}{N_b} \leq \frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}. \end{aligned} \tag{3.22}$$

Consequently, $\mu_{max}^{(i)} = \mu_c^{(i)}$ if and only if

$$\frac{\alpha_1\sqrt{N_{e1}}(\sqrt{(N_b+N_{e1})}-\sqrt{N_{e1}})}{N_b} \leq \beta_2 \leq \frac{\alpha_1 N_{e1}}{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}. \tag{3.23}$$

For Colluder $i \in SC^{all}$

For Colluder $i \in SC^{all}$, if the colluded copy includes all three layer, there are four possibilities for $\mu_{max}^{(i)}$: $\mathbf{u}_{max}^{(i)} = \mathbf{u}_b^{(i)}$, $\mathbf{u}_{max}^{(i)} = \mathbf{u}_{e1}^{(i)}$, $\mathbf{u}_{max}^{(i)} = \mathbf{u}_{e2}^{(i)}$, and $\mathbf{u}_{max}^{(i)} = \mathbf{u}_c^{(i)}$.

$\mathbf{u}_{max}^{(i)} = \mathbf{u}_b^{(i)}$**:** Following the same analysis as the previous section,

$$\begin{aligned}
\mu_{max}^{(i)} = \mu_b^{(i)} \quad &\Leftrightarrow \quad \mu_b^{(i)} \geq \mu_{e1}^{(i)}, \quad \mu_b^{(i)} \geq \mu_{e2}^{(i)}, \quad \text{and} \quad \mu_b^{(i)} \geq \mu_c^{(i)}, \\
\text{where} \quad \mu_b^{(i)} \geq \mu_{e1}^{(i)} \quad &\Leftrightarrow \quad \beta_1 + \beta_2 \leq 1 - (1-\alpha_1)\frac{\sqrt{N_{e1}}}{\sqrt{N_b}}, \\
\mu_b^{(i)} \geq \mu_{e2}^{(i)} \quad &\Leftrightarrow \quad \beta_1 + \beta_2 \leq 1 - \frac{\sqrt{N_{e2}}}{\sqrt{N_b}}, \\
\text{and} \quad \mu_b^{(i)} \geq \mu_c^{(i)} \quad &\Leftrightarrow \quad \beta_1 + \beta_2 \leq 1 - \frac{(1-\alpha_1)N_{e1} + N_{e2}}{\sqrt{N_b}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_b})}.
\end{aligned} \tag{3.24}$$

Note that we have the constraint $0 \leq \beta_1, \beta_2 \leq \beta_1 + \beta_2 \leq 1$ in (3.14) when selecting the collusion parameters. Therefore, from (3.24), in order to satisfy $\mu_b^{(i)} \geq \mu_{e2}^{(i)}$ and let $\mu_b^{(i)} = \max\{\mu_b^{(i)}, \mu_{e1}^{(i)}, \mu_{e2}^{(i)}, \mu_c^{(i)}\}$, $N_{e2} \leq N_b$ must be true. This explains why that in the example shown in Figure 3.2 where $N_{e2} = 2N_b$, among the four detection statistics, $TN_b^{(i)}$ never achieves the best performance.

$\mathbf{u}_{max}^{(i)} = \mathbf{u}_{e1}^{(i)}$**:** In this scenario,

$$\begin{aligned}
\mu_{max}^{(i)} = \mu_{e1}^{(i)} \quad &\Leftrightarrow \quad \mu_{e1}^{(i)} > \mu_b^{(i)}, \quad \mu_{e1}^{(i)} \geq \mu_{e2}^{(i)}, \quad \text{and} \quad \mu_{e1}^{(i)} \geq \mu_c^{(i)}, \\
\text{where} \quad \mu_{e1}^{(i)} \geq \mu_b^{(i)} \quad &\Leftrightarrow \quad \alpha_1 \geq 1 - (1-\beta_1-\beta_2)\frac{\sqrt{N_{e2}}}{\sqrt{N_{e1}}}, \\
\mu_{e1}^{(i)} \geq \mu_{e2}^{(i)} \quad &\Leftrightarrow \quad \alpha_1 \leq 1 - \frac{\sqrt{N_{e2}}}{\sqrt{N_{e1}}}, \\
\text{and} \quad \mu_{e1}^{(i)} \geq \mu_c^{(i)} \quad &\Leftrightarrow \quad \alpha_1 \leq 1 - \frac{(1-\beta_1-\beta_2)N_b + N_{e2}}{\sqrt{N_{e1}}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_{e1}})}.
\end{aligned} \tag{3.25}$$

From (3.25), $N_{e2} \leq N_{e1}$ must hold in order to satisfy $\mu_{e1}^{(i)} \geq \mu_{e2}^{(i)}$ and let $\mu_{e1}^{(i)} = \max\{\mu_b^{(i)}, \mu_{e1}^{(i)}, \mu_{e2}^{(i)}, \mu_c^{(i)}\}$. This is the reason that in Figure 3.2 with $N_{e2} = 2N_{e1}$, $TN_{e2}^{(i)}$ never gives the best traitor-tracing performance.

$\mathbf{u}_{max}^{(i)} = \mathbf{u}_{e2}^{(i)}$**:** Here,

$$\begin{aligned}
\mu_{max}^{(i)} = \mu_{e2}^{(i)} \quad &\Leftrightarrow \quad \mu_{e2}^{(i)} \geq \mu_b^{(i)}, \quad \mu_{e2}^{(i)} \geq \mu_{e1}^{(i)}, \quad \text{and} \quad \mu_{e2}^{(i)} \geq \mu_c^{(i)}, \\
\text{where} \quad \mu_{e2}^{(i)} \geq \mu_b^{(i)} \quad &\Leftrightarrow \quad \beta_1 + \beta_2 \geq 1 - \frac{\sqrt{N_{e2}}}{\sqrt{N_b}},
\end{aligned}$$

$$\begin{aligned}
&\mu_{e2}^{(i)} \geq \mu_{e1}^{(i)} \quad \Leftrightarrow \quad \alpha_1 \geq 1 - \frac{\sqrt{N_{e2}}}{\sqrt{N_{e1}}}, \\
\text{and} \quad &\mu_{e2}^{(i)} \geq \mu_{c}^{(i)} \quad \Leftrightarrow \quad \beta_1 + \beta_2 \geq 1 + \frac{(1-\alpha_1)N_{e1} - \sqrt{N_{e2}}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_{e2}})}{N_b}
\end{aligned} \tag{3.26}$$

Note that if $N_{e2} \geq N_{e1}$ and $N_{e2} \geq N_b$, $\mu_{e2}^{(i)} \geq \mu_{b}^{(i)}$ and $\mu_{e2}^{(i)} \geq \mu_{e1}^{(i)}$ will always hold.

$\mathbf{u}_{max}^{(i)} = \mathbf{u}_{c}^{(i)}$: Following the same analysis as in the previous section,

$$\begin{aligned}
&\mu_{max}^{(i)} = \mu_{c}^{(i)} \quad \Leftrightarrow \quad \mu_{c}^{(i)} \geq \mu_{b}^{(i)}, \quad \mu_{c}^{(i)} \geq \mu_{e1}^{(i)}, \quad \text{and} \quad \mu_{c}^{(i)} \geq \mu_{e2}^{(i)}, \\
\text{where} \quad &\mu_{c}^{(i)} \geq \mu_{b}^{(i)} \quad \Leftrightarrow \quad \beta_1 + \beta_2 \geq 1 - \frac{(1-\alpha_1)N_{e1} + N_{e2}}{\sqrt{N_b}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_b})}, \\
&\mu_{c}^{(i)} \geq \mu_{e1}^{(i)} \quad \Leftrightarrow \quad \beta_1 + \beta_2 \leq 1 - \frac{(1-\alpha_1)\sqrt{N_{e1}}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_{e1}}) - N_{e2}}{N_b}, \\
\text{and} \quad &\mu_{c}^{(i)} \geq \mu_{e2}^{(i)} \quad \Leftrightarrow \quad \beta_1 + \beta_2 \leq 1 + \frac{(1-\alpha_1)N_{e1} - \sqrt{N_{e2}}(\sqrt{N_b + N_{e1} + N_{e2}} - \sqrt{N_{e2}})}{N_b}
\end{aligned} \tag{3.27}$$

3.3.4 Analysis of the Feasible Set

Given the above analysis on $\mu_{max}^{(i)}$, for each given (N_b, N_{e1}, N_{e2}) and $(K^b, K^{b,e1}, K^{all})$, the next step is to study how attackers achieve fairness of collusion and let $\mu_{max}^{(i)}$ be the same for all colluders. This section investigates the constraints on collusion to ensure the fair play of the attack.

Without loss of generality, in this section, we use $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$ as an example to illustrate how colluders achieve fairness of the attack and analyze the constraints on collusion. We assume that colluders generate a high-resolution colluded copy including all three layers. In this scenario, from the analysis in the above section, for a colluder $i_2 \in SC^{b,e1}$ who receives a medium resolution copy, $\mu_{max}^{(i_2)}$ has three possible values: $\mu_{max}^{(i_2)} = \mu_{b}^{(i_2)}$, $\mu_{max}^{(i_2)} = \mu_{e1}^{(i_2)}$ and $\mu_{max}^{(i_2)} = \mu_{c}^{(i_2)}$. Furthermore, for a colluder $i_3 \in SC^{all}$ who receives all three layers, $\mu_{max}^{(i_3)}$ equals to either $\mu_{e1}^{(i_3)}$ or $\mu_{c}^{(i_3)}$, and $\mu_{max}^{(i_3)} \neq \mu_{b}^{(i_3)}$ and $\mu_{max}^{(i_3)} \neq \mu_{e1}^{(i_3)}$. Thus, there are a total of 6 possible scenarios.

Scenario 1 $\mu_{max}^{(i_2)} = \mu_b^{(i_2)}$ **for** $i_2 \in SC^{b,e1}$ **and** $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$ **for** $i_3 \in SC^{all}$

In this scenario, for three colluders $i_1 \in SC^b$, $i_2 \in SC^{b,e1}$ and $i_3 \in SC^{all}$, from (3.15),

$$\mu^{(i_1)} = \frac{\beta_1\sqrt{N_b}}{K^b}\sigma_W, \quad \mu_{max}^{(i_2)} = \frac{\beta_2\sqrt{N_b}}{K^{b,e1}}\sigma_W, \quad \text{and} \quad \mu_{max}^{(i_3)} = \frac{\sqrt{N_{e2}}}{K^{all}}\sigma_W. \tag{3.28}$$

To achieve fairness of the attack, colluders select the collusion parameters $\{\alpha_k, \beta_l\}$ such that $\mu^{(i_1)} = \mu_{max}^{(i_2)} = \mu_{max}^{(i_3)}$. Therefore, we have

$$\beta_1 = \frac{\sqrt{N_{e2}}}{\sqrt{N_b}}\frac{K^b}{K^{all}} = \sqrt{2}\frac{K^b}{K^{all}}, \quad \text{and} \quad \beta_2 = \frac{K^{b,e1}}{K^b}\beta_1 = \sqrt{2}\frac{K^{b,e1}}{K^{all}}. \tag{3.29}$$

In this scenario, since $\mu_b^{(i_2)}$ is the largest among $\{\mu_b^{(i_2)}, \mu_{e1}^{(i_2)}, \mu_c^{(i_2)}\}$, from (3.18), the selected collusion parameters must satisfy

$$\alpha_1 \leq \beta_2\frac{\sqrt{N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}{N_{e1}} = \frac{\sqrt{2N_b}(\sqrt{N_b+N_{e1}}-\sqrt{N_b})}{N_{e1}}\frac{K^{b,e1}}{K^{all}} \triangleq A. \tag{3.30}$$

$A = (2-\sqrt{2})K^{b,e1}/K^{all}$ in our example of $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$. Similarly, to let $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$, from (3.26), α_1, β_1 and β_2 must satisfy

$$\begin{aligned}\alpha_1 &\geq 1 + \frac{(1-\beta_1-\beta_2)N_b - \sqrt{N_{e2}}(\sqrt{N_b+N_{e1}+N_{e2}}-\sqrt{N_{e2}})}{N_{e1}} \\ &= 1 + \frac{N_b}{N_{e1}} - \frac{\sqrt{N_{e2}}(\sqrt{N_b+N_{e1}+N_{e2}}-\sqrt{N_{e2}})}{N_{e1}} - \frac{\sqrt{2}N_b}{N_{e1}}\cdot\frac{K^b+K^{b,e1}}{K^{all}} \triangleq B\end{aligned} \tag{3.31}$$

$B = 4-2\sqrt{2}-\sqrt{2}(K^b+K^{b,e1})/K^{all}$ if $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$.

Define $R^b = K^b/K$, $R^{b,e1} = K^{b,e1}/K$ and $R^{all} = K^{all}/K$ as the percentages of colluders who are in SC^b, $SC^{b,e1}$ and SC^{all}, respectively. $0 \leq R^b, R^{b,e1}, R^{all} \leq R^b + R^{b,e1} + R^{all} = 1$ and $R^{all} \leq 1 - R^b$. Note that during the two-stage collusion in Section 3.1.2.2, $0 \leq \beta_1, \beta_2 \leq \beta_1 + \beta_2 \leq 1$ and $0 \leq \alpha_1 \leq 1$. Therefore, from (3.29), the triplet $(R^b, R^{b,e1}, R^{all})$ must satisfy

$$\sqrt{2}\frac{R^b + R^{b,e1}}{R^{all}} \leq 1 \quad \Leftrightarrow \quad R^{all} \geq \frac{\sqrt{2}}{1+\sqrt{2}}. \tag{3.32}$$

Furthermore, in order to be able to select a α_1 that satisfies both $B \le \alpha_1 \le A$ and $0 \le \alpha_1 \le 1$, it is required that $A \ge 0$ (which is always true for all $K^{b,e1} \ge 0$ and $K^{all} \ge 0$) , $B \le 1$ and $B \le A$. Consequently, $(R^b, R^{b,e1}, R^{all})$ must satisfy

$$\begin{aligned} B \le 1 \quad &\Leftrightarrow \quad R^{all} \le \frac{\sqrt{2}}{3-\sqrt{2}}, \\ \text{and} \quad B \le A \quad &\Leftrightarrow \quad R^{all} \le \frac{2-(2-\sqrt{2})R^b}{6-2\sqrt{2}}. \end{aligned} \tag{3.33}$$

Since $\frac{\sqrt{2}}{3-\sqrt{2}} > \frac{2}{6-2\sqrt{2}} \ge \frac{2-(2-\sqrt{2})R^b}{6-2\sqrt{2}}$ for all $R^b \ge 0$, combining (3.32) and (3.33), we have

$$\frac{\sqrt{2}}{1+\sqrt{2}} \le R^{all} \le \min\left\{ \frac{2-(2-\sqrt{2})R^b}{6-2\sqrt{2}}, 1-R^b \right\}. \tag{3.34}$$

To summarize, if $(R^b, R^{b,e1}, R^{all})$ satisfies (3.34), colluders can achieve fairness of the attack by following, and the resulting feasible set is the black area in Figure 3.6(a). (3.29)-(3.31) when selecting the collusion parameters $\{\alpha_k, \beta_l\}$. In this scenario, $\mu_{max}^{(i_2)} = \mu_b^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$ for $i \in SC^{all}$. Figure 3.6 (a) plots all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.34).

Scenario 2 $\mu_{max}^{(i_2)} = \mu_{e1}^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$ for $i_3 \in SC^{all}$

Following the same analysis as in Section 3.3.4, for the example of $N_b : N_{e1} : N_{e2} = 1:1:2$, if $(R^b, R^{b,e1}, R^{all})$ satisfied

$$\max\left\{ \sqrt{2}R^b, (2-\sqrt{2})(1-R^b) \right\} \le R^{all} \le \min\left\{ \frac{2-(2-\sqrt{2})R^b}{6-2\sqrt{2}}, \frac{\sqrt{2}-\sqrt{2}R^b}{3-\sqrt{2}}, 1-R^b \right\}, \tag{3.35}$$

colluders can guarantee the equal risk of all attackers by selecting

$$\begin{aligned} &\alpha_1 = \sqrt{2}\frac{K^{b,e1}}{K^{all}}, \quad \beta_1 = \sqrt{2}\frac{K^b}{K^{all}}, \\ \text{and} \quad &4-\sqrt{2}-\sqrt{2}\frac{K}{K^{all}} \le \beta_2 \le \min\left\{ 1-\sqrt{2}\frac{K^b}{K^{all}}, (2-\sqrt{2})\frac{K^{b,e1}}{K^{all}} \right\}. \end{aligned} \tag{3.36}$$

Figure 3.6 (b) shows all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.35).

Scenario 3 $\mu_{max}^{(i_2)} = \mu_c^{(i_2)}$ **for** $i_2 \in SC^{b,e1}$ **and** $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$ **for** $i_3 \in SC^{all}$

Given $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$, if $(R^b, R^{b,e1}, R^{all})$ satisfies

$$\max\left\{\frac{2-(2-\sqrt{2})R^b}{4}, \frac{(2-\sqrt{2})+(2\sqrt{2}-2)R^b}{3-\sqrt{2}}\right\} \le R^{all} \le \left\{\frac{2-(2-\sqrt{2})R^b}{6-2\sqrt{2}}, 1-R^b\right\}, \tag{3.37}$$

and colluders select

$$\begin{aligned}
&\beta_1 = \sqrt{2}\frac{K^b}{K^{all}},\\
&\max\left\{2\frac{K^{b,e1}}{K^{all}} - 1, (2-\sqrt{2})\frac{K^{b,e1}}{K^{all}}\right\} \le \beta_2 \le \min\left\{\sqrt{2}\frac{K^{b,e1}}{K^{all}}, 1-\sqrt{2}\frac{K^b}{K^{all}}\right\},\\
&\text{and} \quad \alpha_1 = 2\frac{K^{b,e1}}{K^{all}} - \beta_2,
\end{aligned} \tag{3.38}$$

then $\mu_{max}^{(i_2)} = \mu_c^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_{e2}^{(i_3)}$ for $i_3 \in SC^{all}$, and all colluders have the same probability of being detected by the fingerprint detector. Figure 3.6 (c) plots all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.37).

Scenario 4 $\mu_{max}^{(i_2)} = \mu_b^{(i_2)}$ **for** $i_2 \in SC^{b,e1}$ **and** $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ **for** $i_3 \in SC^{all}$

Given $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$, if

$$\max\left\{2-\sqrt{2}, \frac{4-\sqrt{2}+(\sqrt{2}-1)R^b}{6-\sqrt{2}}\right\} \le R^{all} \le 1-R^b \tag{3.39}$$

holds, by choosing the collusion parameters as

$$\begin{aligned}
&\beta_1 \ge \max\left\{\frac{4K^b}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}}, \frac{3K^b}{K+K^{all}}, \frac{\sqrt{2}K^b}{K^{all}}\right\},\\
&\beta_1 \le \min\left\{\frac{K^b}{K-K^{all}}, \frac{4K^b}{K+K^{all}}\right\},\\
&\beta_2 = \frac{K^{b,e1}}{K^b}\beta_1, \quad \text{and} \quad \alpha_1 = 4 - \frac{K+K^{all}}{K^b}\beta_1,
\end{aligned} \tag{3.40}$$

colluders achieve fairness of collusion and $\mu_{max}^{(i_2)} = \mu_b^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for $i_3 \in SC^{all}$ in this scenario. Figure 3.6 (d) plots all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.39).

Scenario 5 $\mu_{max}^{(i_2)} = \mu_{e1}^{(i_2)}$ **for** $i_2 \in SC^{b,e1}$ **and** $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ **for** $i_3 \in SC^{all}$

Here, under the constraint that $(R^b, R^{b,e1}, R^{all})$ satisfies

$$\max\left\{4R^b - 1, \sqrt{2}R^b, \frac{4-\sqrt{2}-(5-\sqrt{2})R^b}{6-\sqrt{2}}, \frac{\sqrt{2}}{4-\sqrt{2}}\right\} \leq R^{all} \leq 1 - R^b, \tag{3.41}$$

all colluders have the same probability of being detected if they select

$$\begin{aligned}
&\beta_1 \geq \max\left\{\frac{4K^b}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}, \frac{3K^b}{K + K^{all} - K^b}, \frac{\sqrt{2}K^b}{K^{all}}\right\}, \\
&\beta_1 \leq \min\left\{\frac{K^b}{K^{b,e1}}, \frac{4K^b}{K + K^{all}}\right\}, \\
&\beta_2 = 4 - \frac{K + K^{all}}{K^b}\beta_1, \quad \text{and} \quad \alpha_1 = \frac{K^{b,e1}}{K^b}\beta_1
\end{aligned} \tag{3.42}$$

during collusion. In this scenario, $\mu_{max}^{(i_2)} = \mu_{e1}^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for $i_3 \in SC^{all}$. Figure 3.6 (e) shows all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.41).

Scenario 6 $\mu_{max}^{(i_2)} = \mu_c^{(i_2)}$ **for** $i_2 \in SC^{b,e1}$ **and** $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ **for** $i_3 \in SC^{all}$

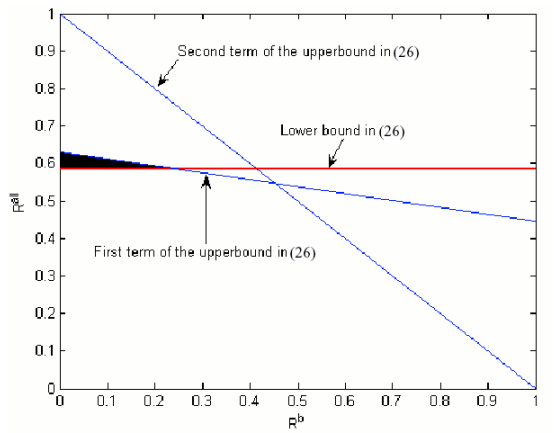
If $(R^b, R^{b,e1}, R^{all})$ satisfies the constraint

$$\max\left\{\frac{3\sqrt{2}-4-(3\sqrt{2}-7)R^b}{3\sqrt{2}-2}, \frac{\sqrt{2}-(\sqrt{2}-1)R^b}{3\sqrt{2}-2}\right\} \leq R^{all} \leq 1 - R^b \tag{3.43}$$

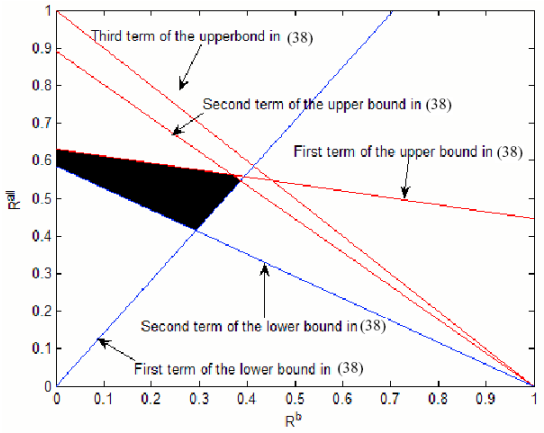
and if the selected parameters are

$$\begin{aligned}
&\beta_1 = \frac{4K^b}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}, \\
&\alpha_1 \geq \max\left\{\frac{3\sqrt{2}K^{b,e1} + 3K^b - 2K^{all}}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}, \frac{4(\sqrt{2}-1)K^{b,e1}}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}\right\}, \\
&\alpha_1 \leq \frac{4K^{b,e1}}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}, \\
\text{and} \quad &\beta_2 + \alpha_1 = \frac{4\sqrt{2}K^{b,e1}}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}},
\end{aligned} \tag{3.44}$$

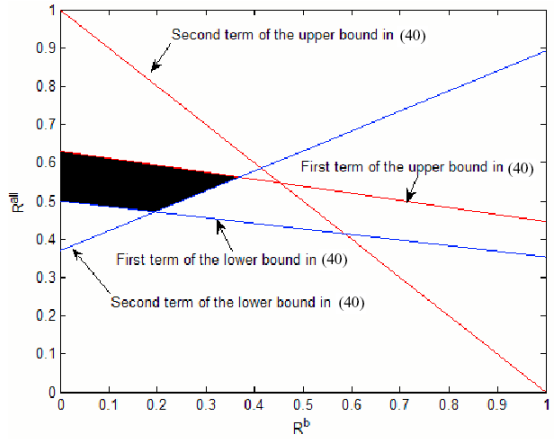
then all colluders have the same risk and $\mu_{max}^{(i_2)} = \mu_c^{(i_2)}$ for $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for $i_3 \in SC^{all}$. Figure 3.6 (f) plots all the $(R^b, R^{b,e1}, R^{all})$ that satisfy (3.43).



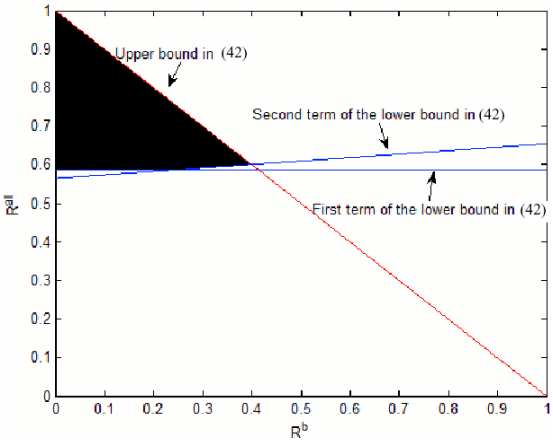
(a)



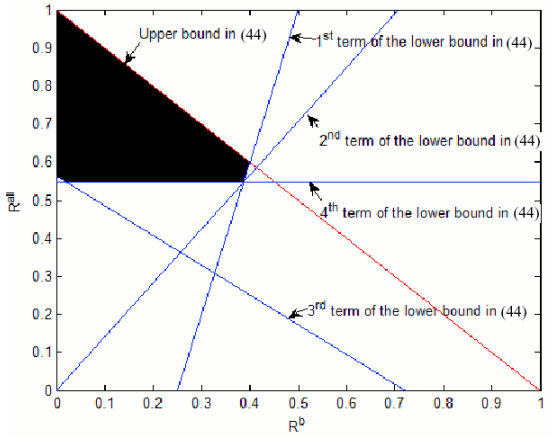
(b)



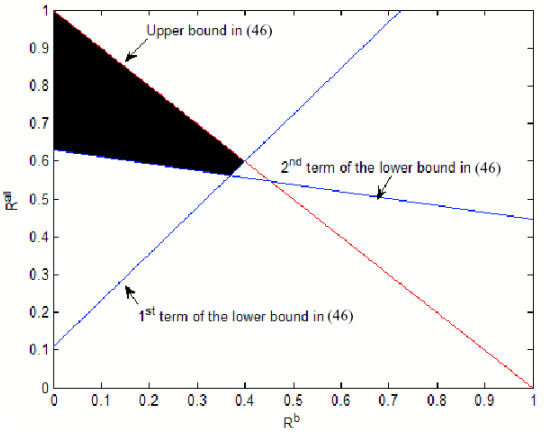
(c)



(d)



(e)



(f)

Figure 3.6: $(R^b, R^{b,e1}, R^{all})$ that satisfy (a): (3.34) in Scenario 1, (b): (3.35) in Scenario 2, (c): (3.37) in Scenario 3, (d): (3.39) in Scenario 4, (e): (3.41) in Scenario 5, and (f): (3.43) in Scenario 6. Here, $N_b : N_{e1} : N_{e2} = 1 : 1 : 2$.

3.3.5 Min-Max Solution

Given the analysis in Section 3.3.4, for three colluders $i_1 \in SC^b$, $i_2 \in SC^{b,e1}$ and $i_3 \in SC^{all}$, the colluders first identify all the possible collusion parameters $\{\alpha_l, \beta_k\}$ that satisfy $\mu_{max}^{(i_1)} = \mu_{max}^{b,e1} = \mu_{max}^{all}$ under the constraints, and then select the one that gives them the minimum risk of being detected.

To demonstrate this process, we use the system setup in Figure 3.3 as an example, where the lengths of the fingerprints embedded in the base layer, the enhancement layer 1 and the enhancement layer 2 are $N_b = 5000$, $N_{e1} = 5000$ and $N_{e2} = 10000$, respectively. When generating fingerprints, we first generate independent Gaussian vectors following distribution $\mathcal{N}(0,1)$ and then apply Gram-Schmidt orthogonalization to produce fingerprints that have equal energies and are strictly orthogonal to each other.

Assume that there are a total of $K = 250$ colluders. Among the 250 colluders, if $K^b = 50$, $K^{b,e1} = 25$, and $K^{all} = 175$, i.e., $(R^b, R^{b,e1}, R^{all}) = (0.2, 0.1, 0.7)$, then from Section 3.3.4, $(R^b, R^{b,e1}, R^{all})$ satisfies the constraint (3.39) in Scenario 4 as in Appendix, the constraint (3.41) in Scenario 5, and the one (3.43) in Scenario 6.

- Since $(R^b, R^{b,e1}, R^{all})$ satisfies the constraint (3.39) in Scenario 4, colluders can guarantee the equal risk of all colluders if they choose

$$\begin{aligned}
&\beta_1 \geq \max\left\{\frac{4K^b}{\sqrt{2}K - (\sqrt{2}-1)K^b + (2-\sqrt{2})K^{all}}, \frac{3K^b}{K+K^{all}}, \frac{\sqrt{2}K^b}{K^{all}}\right\} = 0.4594,\\
&\beta_1 \leq \min\left\{\frac{K^b}{K-K^{all}}, \frac{4K^b}{K+K^{all}}\right\} = 0.4706,\\
&\beta_2 = \frac{K^{b,e1}}{K^b}\beta_1, \quad \text{and} \quad \alpha_1 = 4 - \frac{K+K^{all}}{K^b}\beta_1.
\end{aligned} \tag{3.45}$$

Here, $\mu_{max}^{(i_2)} = \mu_b^{(i_2)}$ for colluder $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for colluder $i_3 \in SC^{all}$.

For any colluder $i \in SC$, $\mu_{max}^{(i)}$ has the smallest possible value of 2.0545 when $\beta_1 = 0.4594$, $\beta_2 = 0.2297$, and $\alpha_1 = 0.0951$.

- Following (3.42), when colluders select parameters

$$\begin{aligned}
&\beta_1 \geq \max\left\{\frac{4K^b}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}}, \frac{3K^b}{K+K^{all}-K^b}, \frac{\sqrt{2}K^b}{K^{all}}\right\} = 0.4594,\\
&\beta_1 \leq \min\left\{\frac{K^b}{K^{b,e1}}, \frac{4K^b}{K+K^{all}}\right\} = 0.4706,\\
&\beta_2 = 4 - \frac{K+K^{all}}{K^b}\beta_1, \quad \text{and} \quad \alpha_1 = \frac{K^{b,e1}}{K^b}\beta_1,
\end{aligned} \tag{3.46}$$

they have the same probability of being detected. Here, $\mu_{max}^{(i_2)} = \mu_{e1}^{(i_2)}$ for colluder $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for colluder $i_3 \in SC^{all}$. For any colluder $i \in SC$, $\mu_{max}^{(i)}$ reaches its minimum value of 2.0545 when $\beta_1 = 0.4594$, $\beta_2 = 0.0951$, and $\alpha_1 = 0.2297$.

- Following (3.44), colluders can also achieve fairness of collusion by selecting

$$\begin{aligned}
\beta_1 &= \frac{4K^b}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}} = 0.4594,\\
\alpha_1 &\geq \max\left\{\frac{3\sqrt{2}K^{b,e1}+3K^b-2K^{all}}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}}, \frac{4(\sqrt{2}-1)K^{b,e1}}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}}\right\} = 0.2297,\\
\alpha_1 &\leq \frac{4K^{b,e1}}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}} = 0.0951, \text{ and}\\
\beta_2 &= \frac{4\sqrt{2}K^{b,e1}}{\sqrt{2}K-(\sqrt{2}-1)K^b+(2-\sqrt{2})K^{all}} - \alpha_1 = 0.3248 - \alpha_1
\end{aligned} \tag{3.47}$$

during collusion. In this scenario, $\mu_{max}^{(i_2)} = \mu_c^{(i_2)}$ for colluder $i_2 \in SC^{b,e1}$ and $\mu_{max}^{(i_3)} = \mu_c^{(i_3)}$ for colluder $i_3 \in SC^{all}$, and $\mu_{max}^{(i)} = 2.0545$ for all colluders.

The means of the detection statistics in these three scenarios are the same; therefore, colluders can choose either (3.45), (3.46) or (3.47) during collusion. (In fact, (3.45) and (3.46) are the two boundaries of (3.47).)

In the example of $(K^b, K^{b,e1}, K^{all}) = (50, 75, 125)$, the constraints (3.35) in Scenario 2 and (3.37) in Scenario 3 are satisfied, and the minimum value of $\mu_{max}^{(i)}$ equals to 2.5298, when colluders select $(\beta_1 = 0.5657,\ \beta_2 = 0.0544,\ \alpha_1 = 0.4485)$ or use $(\beta_1 = 0.5657,\ \beta_2 = 0.3929,\ \alpha_1 = 0.4071)$ during collusion.

If $(K^b, K^{b,e1}, K^{all}) = (50, 125, 75)$, none of the six constraints in Section 3.3.4 are satisfied, and colluders cannot generate a high-resolution colluded copy while still achieving fairness of the attack. They have to lower the resolution of the attacked copy to medium to guarantee the equal risk of all colluders.

3.4 Simulation Results

In our simulations, we test over the first 40 frames of "carphone", and use $F_b = \{1, 5, \cdots, 37\}$, $F_{e1} = \{3, 7, \cdots, 39\}$ and $F_{e2} = \{2, 4, \cdots, 40\}$ as an example of the temporal scalability. The lengths of the fingerprints embedded in the base layer, enhancement layer 1 and enhancement layer 2 are $N_b = 42987,\ N_{e1} = 42951$ and $N_{e2} = 85670$, respectively. We assume that there are a total of $M = 750$ users and $|\mathbf{U}^b| = |\mathbf{U}^{b,e1}| = |\mathbf{U}^{all}| = 250$. We first generate independent vectors following Gaussian distribution $\mathcal{N}(0, 1/9)$, and then apply Gram-Schmidt orthogonalization to generate orthogonal fingerprints for different users.

We assume that $0 \leq K^b, K^{b,e1}, K^{all} \leq 250$ are the number of colluders in subgroups SC^b, $SC^{b,e1}$ and SC^{all}, respectively, and the total number of colluders is fixed to 250. During collusion, the colluders apply the intra-group collusion followed by the inter-group collusion, and follow Section 3.3 when choosing the collusion parameters. In our simulations, we adjust the power of the additive noise such that $||\mathbf{n}_j||^2 = ||JND_j \mathbf{W}_j^{(i)}||^2$

for every frame j in the video sequence.

The fingerprint detector follows Section 3.2.3 when identifying selfish colluders. The detector first estimates the means of different detection statistics, selects the detection statistics with the largest estimated mean, and then identifies the colluders.

In Figure 3.7, we compare the performance of three detectors: the simple collective detector in (3.4), the optimum detector which always selects the detection statistics with the largest mean, and the self-probing detector in Section 3.2.3. Similar to Figure 3.3, when the means of different detection statistics differ significantly from each other, the proposed self-probing detector in Section 3.2.3 always selects the optimum detection statistics with the largest mean. When the difference between different means is small, the optimum and the suboptimum detection statistics have approximately the same performance. Thus, even though the proposed method may make errors when deciding which detection statistics give the best performance, selecting the suboptimum detection strategy does not significantly deteriorate the detection performance when compared with the optimum detection statistics. In Figure 3.7, the performance gap is smaller than 2×10^{-3} and can be ignored. Exploring side information about collusion can significantly help improve the detection performance, and the proposed self-probing detector has approximately the same performance as the optimum detector with perfect knowledge of the detection statistics' means.

Figure 3.8 plots each colluder's probability of being detected when they follow Section 3.3 to select the collusion parameters. It is obvious that in this example, all colluders have the same probability of being detected and this multi-user collusion achieves fairness of the attack.

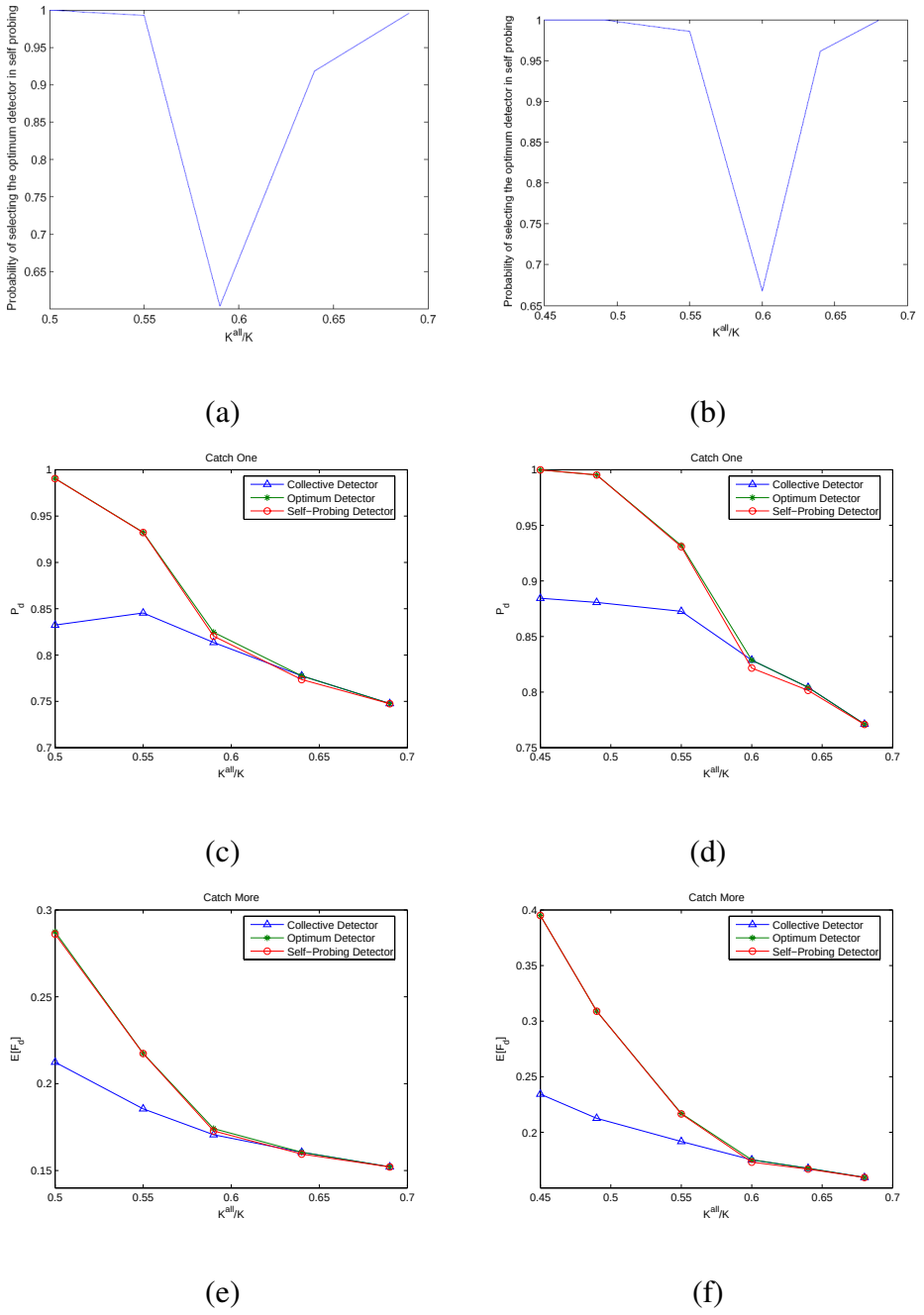
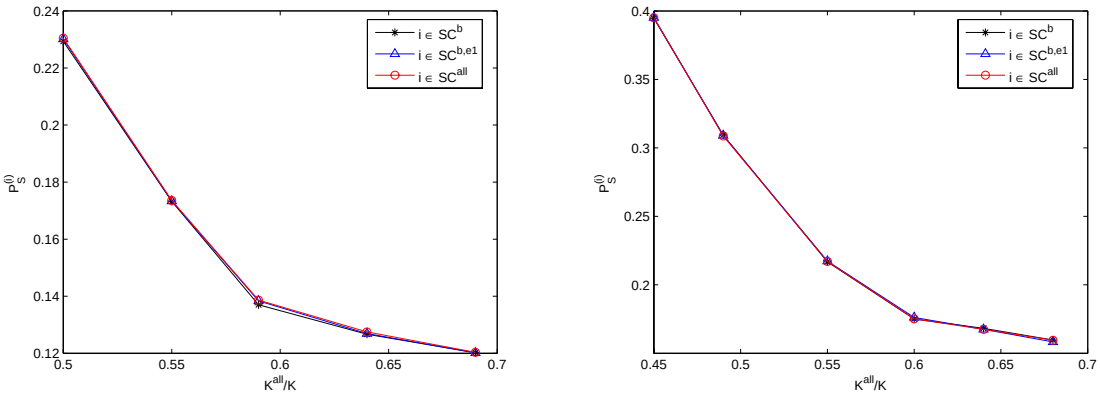


Figure 3.7: Simulation results on the first 40 frames of sequence "carphone" from 10000 simulation runs. (a) and (b): Probability that the self-probing detector selects the optimum detection statistics with the largest mean. (c) and (d): P_d when $P_{fp} = 10^{-3}$. (e) and (f): $E[F_d]$ with $E[F_{fp}]$ fixed as 10^{-3}. In (a), (c), and (e), $R^b = 0.2$ and each point on the x axis corresponds to a unique triplet $(K^b, K^{b,e1}, K^{all})$ where $K^b = 50$ and $K^{b,e1} = K - K^b - K^{all}$. In (b), (d), and (f), $R^b = 0.25$, and each point corresponds to a unique triplet $(K^b, K^{b,e1}, K^{all})$ where $K^b = 73$, and $K^{b,e1} = K - K^b - K^{all}$. Results are based 10000 simulation runs.



(a) $R^b = 0.20$ (b) $R^b = 0.25$

Figure 3.8: Each colluder's probability of being detected ($P_s^{(i)}$) when they follow Section 3.3 to select the collusion parameters. The simulation setup is similar to that in Figure 3.7. There are a total of $K = 250$ colluders. In (a), $K^b = 50$ of them receive the fingerprinted based layer only, and each point on the x axis corresponds to a unique triplet $(K^b, K^{b,e1}, K^{all})$ where $K^b = 50$ and $K^{b,e1} = K - K^b - K^{all}$. In (b), $K^b = 75$. Results are based 10000 simulation runs.

3.5 Chapter Summary

This chapter studies user behavior in the multimedia fingerprint social networks. We model the complex dynamics of the users in the social network using game theory and find the optimal strategies of both players in the game. We study how side information about collusion can help the fingerprint detector increase the traitor-tracing capability, and influence the strategies of the colluders and the forensic detector.

We first investigated multimedia forensics with side information. Our analysis and simulation results show that the side information about the means of the detection statistics can help the detector significantly improve the collusion resistance. We then propose a method for the detector himself/herself to probe such side information from the colluded copy. Our simulation results demonstrate that the proposed self-probing detector has approximately the same performance as the optimal fingerprint detector, and the difference between these two can be ignored.

Side information not only improves the fingerprint detector's collusion resistance, but it also affects each colluder's probability of being detected and makes some colluders take a larger risk than others. Thus, it breaks the collective fairness equilibrium between the colluders and the fingerprint detector, and they have to choose different strategies. We model the colluder-detector dynamics with side information as a zero-sum game. We show that under the assumption that colluders demand absolute fairness of the attack, the min-max solution achieves the equilibrium which is the optimal strategy of all users in the multimedia fingerprint social network. Neither the colluders nor the fingerprint detector can further increase their payoff and, therefore, they have no incentive to move away from

this equilibrium.

Chapter 4

Incentive Cooperation Strategies for Peer-to-Peer Live Multimedia Streaming Social Networks

Peer-to-Peer (P2P) live streaming is one of the biggest multimedia social networks on the Internet in which users collaborate with each other to watch live broadcasting TV programs over networks simultaneously. Due to the fully distributed nature, centralized architecture is not an option to enforce and regulate users cooperation. Therefore, it is critical to analyze the users' behavior to fully understand how would the users behave to maximize their own utilities, thus provide incentives and develop optimal strategies for cooperation. In addition, some users in P2P live streaming systems are strategic and rational, in that they are likely to manipulate any incentive system (for example, by cheating) to maximize their payoff. And some *malicious* are attackers and aim to exhaust others' resources and attack the system. As such, in large-scale social networks, users influence each other's decisions and performance, and there exist complicated dynamics among users. It is of ample importance to investigate user behavior and analyze the impact of human factors on multimedia social networks.

In the literature, there have been a lot of work on providing incentives for cooperation in P2P file sharing [51, 51–53]. However, providing incentives in P2P live streaming is much more challenging than file sharing, and only a few work has addressed this problem [24, 26, 27]

The above prior work on incentive mechanisms for P2P live streaming either relied on trusted central billing services to implement micro-payment, or they assumed that all users are rational and honest. In real-world social networks, there are always users with different objectives, for example, rational users and attackers, and everyone wants to maximize his or her own payoff as in Figure 4.1. Hence, in this chapter, we will focus on designing distributed, cheat-proof and attack-resistant cooperation stimulation strategies for P2P live streaming social networks under a game theoretic framework. We first consider a simple scenario with non-scalable video coding and study a game with only two players. We investigate the Nash equilibriums of the game and derive cheat-proof stimulation strategies. This analysis aims to stimulate each pair of users in P2P live streaming to cooperate with each other and achieve better performance. Then, we address the issue of cooperation stimulation among multiple users with non-scalable video coding, and investigate cheat-proof and attack-resistant incentive mechanisms. Finally, we design a chunk-request algorithm to maximize users' video quality when the layered video coding is used, which is the unique issue in P2P live streaming. We combine the algorithm together with our proposed cheat-proof and attack-resistant strategies to provide incentives for cooperation. Our proposed cheat-proof and attack-resistant mechanism rewards users who contribute more with more video chunks and thus better quality. It includes a request-answering algorithm for the data supplier to upload more to the peers

Figure 4.1: User dynamics in real world social networks

from which it downloads more, and a chunk-request algorithm for the requestor to address the tradeoffs among different quality measure and to optimize the reconstructed video quality.

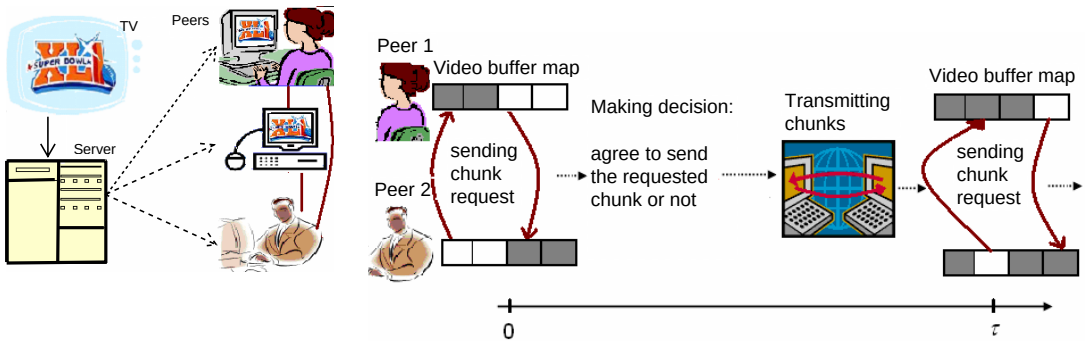
The rest of this chapter is organized as follows. Section 4.1 introduces the mesh-pull P2P live streaming system model, studies the two-player game model, and analyzes the Nash equilibriums. In section 4.2, we propose a cheat-proof and attack-resistant strategy to stimulate user cooperation among all peers in P2P live streaming, and prove that it achieves Nash equilibrium, Pareto optimality, and subgame perfectness. Section 4.3 proposes a cheat-proof and attack-resistant incentive mechanism with layered video coding. Section 4.4 shows simulation results to evaluate the performance of the proposed strategies. Finally, Section 4.5 concludes this chapter.

4.1 Optimal Strategies in a Two-Player P2P Live Streaming Game

In this section, we first describe how two users in a P2P live streaming social network cooperate with each other. We then define the payoff function and introduce the game-theoretic formulation of user dynamics.

4.1.1 Mesh-pull P2P Live Streaming

We first introduce the basic protocol and streaming mechanisms of mesh-pull P2P live streaming system as in Fig. 4.2(a). In a mesh-pull delivery architecture for live video streaming [25], a compressed video bit stream of bit rate B bps is divided into media chunks of M bits per chunk, and all the chunks are available at the original streaming server. When a peer wants to view the video, he/she obtains a list of peers that are currently watching the video, and establishes partnership with several peers. At any instance, a peer buffers up to a few minutes worth of chunk within a sliding window. Each user keeps a "buffer map", indicating the chunks that he/she has currently buffered and can share with others, and they exchange their buffer maps with each other frequently. For example, in Figure 4.2(b), peer 1 has first 2 chunks, while peer 2 has last 2 chunks, indicated by grey blocks in their video buffer maps. After peer 1 receives peer 2's buffer map, peer 1 can request one or more chunks that peer 2 has advertised in his/her buffer map. Time is divided into rounds of τ seconds. Figure 4.2 (b) shows how the peers cooperate with each other: at the beginning of each round, every user sends a chunk request either



(a) Overall system (b) Peer cooperation

Figure 4.2: Mesh-pull P2P Live Streaming Model

to one of his peers or to the original streaming server. Then, the supplier either replies with the requested chunk or rejects the request.

4.1.2 Two-Player Game Model

We assume that there are totally N users in the live streaming social network and every user buffers L chunks. The video stream is originally stored in the streaming server whose upload bandwidth can only afford transmitting K' chunks in one round (τ seconds) with $K' << N$. The server has no information of users' network topology, and the peer-to-peer system is information-pull, which means that the server only sends chunks that are requested by some users, and it replies the chunk requests in a round robin fashion. Due to the playback time lags among peers [25], different users request different chunks, and the server cannot answer all users' requests. In such a scenario, peers have to help each other to receive more chunks and thus better-quality videos.

This section investigates the incentive mechanisms for peer cooperation in live streaming. We start with a simple scenario with two cooperating users and nonscalable video coding structure. To simplify the analysis, in this chapter, we consider a simple

scenario where in each round, every peer can only request one chunk from the other peer and also uploads at most one chunk to the other.

We first define the utility (payoff) function of the two-player game. In each round, if player i answers the other player k's request and sends the requested data chunk to k, we define i's cost c_i as the percentage of his/her upload bandwidth used to transmit the requested chunk. That is, $c_i = M/(W_i\tau)$, where W_i is player i's total available upload bandwidth, M is the size of the chunk and τ is the time duration of the round. If player k forwards the data that i requested and player i receives the chunk correctly, then i receives a gain of g_i, which is a user-defined value between 0 and 1. Every user in the p2p live streaming social network defines his/her own value of g_i depending on how much he/she wants to watch the video. For instance, if all the user does is watching the live streaming and not distracted by other activities, g_i can be chosen as 1, which also implies that the user is willing to cooperate with others to get the better-quality video. On the other hand, if the user is watching several videos, browsing the Internet, or downloading files simultaneously, he/she will not value the live streaming much thus set lower value g_i. Intuitively, if the user does not care about the video quality, g_i would be set to 0 and the user will download the video directly from the server and not join the live streaming social network, since by joining the live streaming social network, some of his/her upload bandwidths would be occupied. We assume that c_i is upper bounded by c_{max}, which is the same as if there exists a minimum upload bandwidth W_{min} for all users such that $W_i \geq W_{min}$. The minimum upload bandwidth constraint is necessary since if the user can not even completely upload a chunk in one round period, other users have no incentive to cooperate with him/her. Here, W_i and g_i are player i's private information, and it is not

known to the other player unless player i reports them.

Let the action of player i takes *at each round* be a_i. In each round, player i can choose its action a_i from $0,1$, where $a_i = 0$ means in this round, player i chooses not to respond to the other player's request, while $a_i = 1$ indicates that player i is willing to cooperate at this round. Let P_{12} denote the probability that the chunk is successfully transmitted from user 1 to user 2, and P_{21} is defined as the probability that user 2 successfully transmits the requested chunk to user 1. Then, for each round, provided that the action profile (a_1, a_2) being taken, player 1 and 2's payoffs are calculated as follows:

$$\begin{aligned}\pi_1(a_1,a_2) &= (a_2P_{21})g_1 - a_1c_1 = (a_2P_{21})g_1 - a_1\frac{M}{W_1\tau}\\ \pi_2(a_1,a_2) &= (a_1P_{12})g_2 - a_2c_2 = (a_1P_{12})g_2 - a_2\frac{M}{W_2\tau}.\end{aligned} \tag{4.1}$$

The above payoff function consists of two terms: the first term in π_i denotes the gain of user i with respect to the other's action, and the second term denotes his/her cost with respect to his/her own action. From (4.1), it is reasonable to assume that $P_{21}g_1 \geq c_1$ and $P_{12}g_2 \geq c_2$, since users will only cooperate with each other if cooperation can benefit both users and give them positive payoffs. Let $\pi(a_1,a_2) = (\pi_1(a_1,a_2), \pi_2(a_1,a_2))$ be the payoff profile.

It is easy to check that, if this game will only be played for onetime, the only Nash equilibrium (NE) is $(0,0)$, which means no one will answer the other's request. According to the backward induction principle [55], this is also true when the repeated game will be played for finite times with game termination time known to both players. Therefore, in such scenarios, for each player, its only optimal strategy is to always play noncooperatively. However, in live streaming systems, these two players will interact

many rounds and no one can know exactly when will the other user quit the game. Next, we show that cooperative strategies can be obtained under a more realistic setting. Let $\mathbf{s}_i$= $(a_i^{(1)}, a_i^{(2)}, ...)$ denote player i's behavior strategy in the infinitely repeated game, where $a_i^{(j)}$ be the action that player i takes at the j^{th} round, and $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2)$ is the strategy profile. When the game is played more than one time, sum of payoff in every time should be considered as each players utility. However, in infinite time game model, sum of payoff usually goes to infinity, therefore, averaged payoff is considered instead. Which means, we consider the following utility function of the infinitely repeated game:

$$U_i(\mathbf{s}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \pi_i(\mathbf{s}). \tag{4.2}$$

Let us analyze the NEs for the infinitely repeated game with utility function U_i defined as above. According to the Folk theorem [55], there exists at least one NE to achieve every feasible and enforceable payoff profile. A feasible payoff profile is the payoff that can be achieved; an enforceable payoff profile is the payoff that can be enforced by any mechanism to be achieved, which is, a feasible payoff profile that every players payoff is larger than or equal to zero. The set of feasible payoff profiles for the above game is:

$$\begin{aligned} V_0 = \text{convex hull}\{(v_1, v_2) | \exists\ (a_1, a_2) \text{ with } (\pi_1(a_1, a_2), \pi_2(a_1, a_2)) = (v_1, v_2)\}, \\ \text{where } a_1, a_2 \in \{0, 1\}. \end{aligned} \tag{4.3}$$

and the set of enforceable payoff, denoted by V_1,

$$V_1 = \{(v_1, v_2) | (v_1, v_2) \in V_0 \quad \text{and} \quad v_1, v_2 \geq 0\}. \tag{4.4}$$

Figure 4.3 illustrate the feasible and the enforceable regions of the above infinitely repeated game. The feasible region is inside the convex hull of $\Big\{(0,0), (P_{21}g_1, -\frac{M}{W_2\tau}), (P_{21}g_1 - \frac{M}{W_1\tau}, P_{12}g_2 -$

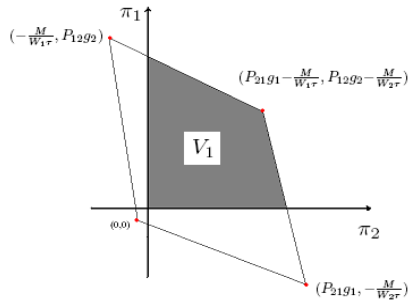


Figure 4.3: Feasible and Enforceable payoff profiles

V_1 is the gray region shown in Figure 4.3, which is the intersection of the feasible region and the first quadrant. It is clear that there exists an infinite number of Nash Equilibriums. To simplify our equations, in this chapter, we use $\mathbf{x} = (x_1, x_2)$ to denote the set of NE strategies corresponding to the enforceable payoff profile $(\pi_1(\mathbf{x}), \pi_2(\mathbf{x})) = \left(x_2 P_{21} g_1 - x_1 \frac{M}{W_1 \tau}, x_1 P_{12} g_2 - x_2 \frac{M}{W_2 \tau}\right)$. Intuitively, the NE strategy x_i can be viewed as the averaged action that player i takes over all rounds in the infinite game. Thus $x_i = \lim_{T\to\infty} \sum_{j=0}^{T} a_i^{(j)}/T$, and $0 \leq x_i \leq 1$.

4.1.3 Nash Equilibrium Refinement

From the above analysis, one can see that the infinitely repeated game has infinite number of Nash Equilibriums, and apparently, not all of them are simultaneously acceptable. For example, the payoff profile $(0,0)$ is not acceptable from both players' point of view. Therefore, in this section, we will discuss how to refine the equilibriums based on new optimality criteria to eliminate those less rational Nash Equilibriums and find out which equilibrium is cheat-proof. In this section, we consider the most widely used optimality criteria in the literature [7], [55]: Pareto optimality, proportional fairness, and absolute fairness.

4.1.3.1 Pareto Optimality

A payoff profile $v \in V_0$ is Pareto Optimal if and only if there is no $v' \in V_0$ such that $v'_i \geq v_i$ for all $i \in N$ [7]. Pareto optimality means no one can increase his/her payoff without degrade other's, which the rational players will always go to. It's clear from Figure 4.3 that the line segment between $(P_{21}g_1, -\frac{M}{W_2\tau})$ and $(P_{21}g_1 - \frac{M}{W_1\tau}, P_{12}g_2 - \frac{M}{W_2\tau})$ in the first quadrant and the line segment between $(-\frac{M}{W_1\tau}, P_{12}g_2)$ and $(P_{21}g_1 - \frac{M}{W_1\tau}, P_{12}g_2 - \frac{M}{W_2\tau})$ in the first quadrant is the Pareto-Optimal set.

4.1.3.2 Proportional Fairness

Next, we will further refine the solution set based on the criterion of proportional fairness. Here, a payoff profile is proportionally fair if the product $U_1(s)U_2(s)$ can be maximized, which can be achieved by maximizing the product $\pi_1(x)\pi_2(x)$ in every round. It has been shown that the proportional fairness solution is always Pareto Optimal. The proportional fairness point $x^* = (x_1^*, x_2^*)$ can be derived by solving :

$$\begin{aligned} \max_{x_1,x_2} f(x_1,x_2) &= x_1x_2(P_{12}P_{21}g_1g_2 + c_1c_2) - x_1^2c_1P_{12}g_2 - x_2^2c_2P_{21}g_1 \\ s.t. &\quad 0 \leq x_1, x_2 \leq 1. \end{aligned} \tag{4.5}$$

In (4.5), same as in (4.1), $c_i = M/(W_i\tau)$ for $i = 1,2$. It can be easily shown that the objective function $f(x_1,x_2)$ and the constraint functions are continuously differentiable at any feasible points, satisfying the Karush-Kuhn-Tucker conditions [56]. Thus the maximizer (x_1^*, x_2^*) either satisfies $\nabla f(x_1^*, x_2^*) = 0$ or is on the boundary of the feasible region. If

$\nabla f(x_1^*, x_2^*) = 0$, then (x_1^*, x_2^*) satisfies

$$\begin{aligned}
\left.\frac{\partial \pi_1(x)\pi_2(x)}{\partial x_1}\right|_{(x_1^*, x_2^*)} &= x_2^*(P_{12}P_{21}g_1g_2 + c_1c_2) - 2x_1^*P_{12}g_2c_1 = 0 \\
\left.\frac{\partial \pi_1(x)\pi_2(x)}{\partial x_2}\right|_{(x_1^*, x_2^*)} &= x_1^*(P_{12}P_{21}g_1g_2 + c_1c_2) - 2x_2^*P_{21}g_1c_2 = 0,
\end{aligned} \tag{4.6}$$

which has only one solution $(x_1^* = 0, x_2^* = 0)$ with $f(0,0) = 0$. Apparently, it is not a desired solution. If (x_1^*, x_2^*) is on the boundary of the feasible region, then it satisfies

$$\begin{aligned}
x_1^* = 1,\ x_2^* &= \min\left\{1, \arg\max_{x_2} f(1, x_2)\right\} = \min\left\{1, \frac{P_{12}P_{21}g_1g_2 + c_1c_2}{2c_2P_{21}g_1}\right\}, \\
\text{or} \quad x_2^* = 1,\ x_1^* &= \min\left\{1, \arg\max_{x_1} f(x_1, 1)\right\} = \min\left\{1, \frac{P_{12}P_{21}g_1g_2 + c_1c_2}{2c_1P_{12}g_2}\right\}.
\end{aligned} \tag{4.7}$$

Combining (4.6) and (4.7), we can obtain the unique proportional fairness point:

$$x^* = \begin{cases}
\left(\frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{12}g_2M/(W_1\tau)}, 1\right) & \text{if} \quad \frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{12}g_2M/(W_1\tau)} \le 1, \\
(1,1) & \text{if} \quad \frac{2}{P_{21}g_1W_1\tau/M + M/(P_{12}g_2W_2\tau)} \le 1 \le \frac{P_{12}g_2}{2M/(W_2\tau)} + \frac{M/(W_1\tau)}{2P_{21}g_1}, \\
\left(1, \frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{21}g_1M/(W_2\tau)}\right) & \text{if} \quad \frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{21}g_1M/(W_2\tau)} \le 1.
\end{cases} \tag{4.8}$$

4.1.3.3 Absolute Fairness

Although absolute fairness solution is not always Pareto-Optimal, it is also an important criteria in many situations. Here we consider the absolute fairness in payoff, which refer to intuitively the most direct fairness criteria that the payoff of every player in the game is the same. By solving $\pi_1(x^*) = \pi_2(x^*)$, we can get the unique absolute fairness solution as follows:

$$x^* = \begin{cases}
\left(\frac{P_{21}g_1 + M/(W_2\tau)}{P_{12}g_2 + M/(W_1\tau)}, 1\right) & \text{if} \quad P_{12}g_2 + \frac{M}{W_1\tau} \ge P_{21}g_1 + \frac{M}{W_2\tau} \\
\left(1, \frac{P_{12}g_2 + M/(W_1\tau)}{P_{21}g_1 + M/(W_2\tau)}\right) & \text{if} \quad P_{21}g_1 + \frac{M}{W_2\tau} \ge P_{12}g_2 + \frac{M}{W_1\tau}.
\end{cases} \tag{4.9}$$

4.1.4 Optimal and cheat-proof Strategies

In Section 5.2.2,we obtained several unique equilibriums with different optimality criteria. However, as in (4.8) and (4.9), all these solutions involve some private information (g_i, W_i, P_{ji}) reported by each player. Due to players' greediness, honestly reporting private information cannot be taken for granted and players may tend to cheat whenever they believe cheating can increase their payoffs.

4.1.4.1 Cheat on Private Information (g_i, W_i, P_{ji})

One way of cheating is to cheat on the private information (g_i, W_i, P_{ji}). First, let us examine whether the proportional fairness solution in (4.8) is cheat-proof with respect to (g_i, W_i, P_{ij}).

From (4.8), when

$$\frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{21}g_1M/(W_2\tau)} = \frac{P_{12}g_2}{2M/(W_2\tau)} + \frac{M/(W_1\tau)}{2P_{21}g_1} \leq 1, \tag{4.10}$$

$x_1^* = 1$ is fixed and

$$x_2^* = \frac{P_{12}g_2}{2M/(W_2\tau)} + \frac{M/(W_1\tau)}{2P_{21}g_1}. \tag{4.11}$$

From (4.11), if user 2 reports false and lower values of the product $P_{12}g_2W_2$, he/she can lower x_2^* and, therefore, further increase his/her own payoff $\pi_2(1, x_2^*) = P_{12}g_2 - x_2^*\frac{M}{W_2\tau}$. Similarly, when

$$\frac{P_{12}P_{21}g_1g_2 + M^2/(W_1W_2\tau^2)}{2P_{12}g_2M/(W_1\tau)} = \frac{P_{21}g_1}{2M/(W_1\tau)} + \frac{M/(W_2\tau)}{2P_{12}g_2} \leq 1, \tag{4.12}$$

$x_2^* = 1$ is fixed and

$$x_1^* = \frac{P_{21}g_1}{2M/(W_1\tau)} + \frac{M/(W_2\tau)}{2P_{12}g_2}. \tag{4.13}$$

By falsely reporting lower values of the product $P_{21}g_1W_1$, user 1 can lower x_1^* and thus further increases his/her own payoff $\pi_1(x_1^*,1) = P_{21}g_1 - x_1^*\frac{M}{W_1\tau}$. Therefore, the proportional fairness solution in (4.8) is not cheat-proof. Applying similar analysis on the absolute fairness solution in (4.9), we can also prove that the absolute fairness solution is also not cheat-proof with respect to private information. Therefore, players have no incentives to honestly report their private information. On the contrary, they will cheat whenever cheating can increase their payoff.

From the above analysis, to maximize their own payoffs, both players will report the minimum value of the product $P_{ji}g_iW_i$. Since we have assume that $P_{ji}g_i \geq c_i = M/(W_i\tau)$ and $W_i \geq W_{min}$, both players will claim $P_{ji}g_iW_i = M/\tau$, and the solution (4.8) and (4.9) become

$$\mathbf{x}^* = (1,1), \tag{4.14}$$

and the corresponding payoff profile is:

$$\mathbf{v}^* = (P_{21}g_1 - \frac{M}{W_1\tau}, P_{12}g_2 - \frac{M}{W_2\tau}). \tag{4.15}$$

It implies that both players should always cooperate with each other. It is clear that the solution in (5.12) forms an Nash Equilibrium, is Pareto-Optimal, and is cheat-proof with respect to private information g_i, W_i and P_{ji}

4.1.4.2 Cheat on Buffer Map Information

Here we assume every user has a buffer with fixed length L, which means the buffer stores L future chunks. At the beginning of each round, each player has to exchange his/her own *buffer information* with the other player, that is, telling the other player which

chunks he/she has in the buffer. The other way of cheating is to cheat on the buffer map information, that is, although player i has the kth chunk, LC_k, in the buffer, he/she tell the other player, player j, that he/she does not have LC_k. By reporting this wrong buffer map information, the cheating user i can reduce the number of requests from user j since user j will never ask for the cheated chunk LC_k. As a result, increasing the cheating player's own payoff by lower s_i.

The only circumstance that cheating on buffer information is effective is that, when the cheated chunk LC_k is the only chunk that the honest player needs, and the honest user has other chunks that the cheater needs. Which means, the cheater can ask the honest user for help, but the honest user can not ask the cheater for help because there is no chunk in the cheater's buffer that the honest user need. Under this circumstance, the cheater get the reward, but the honest user gets nothing. To prevent this kind of cheating, each player should not send chunks more than the other one sent.

To summarize, our *two-player cheat-proof P2P live streaming cooperation strategy* is as follows: in the two-player P2P live streaming game, in order to maximize each user's own payoff and be resistant to possible cheating behavior, a player should not send more chunks than its opponent does for it. Specifically, for each player in each round, it should always agree to send the requested chunk unless its opponent refused it in the previous round or there's no useful chunk in the opponent's buffer.

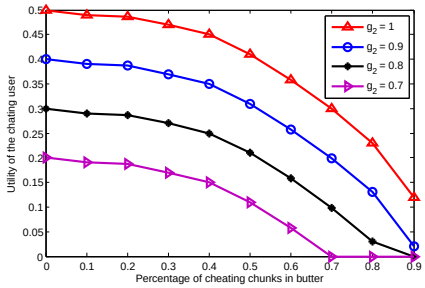


Figure 4.4: Simulation results on 2-person cheat-proof P2P live streaming cooperation strategy.

4.1.5 Performance of 2-Person cheat-proof Cooperation Strategy

Here we study the performance of the two-player cheat-proof P2P live streaming cooperation strategy proposed above. In our simulation setting, there are totally 500 users in the network, and everyone is downloading chunks directly from the server. Each peer is either a DSL peer with 768 kbps uplink bandwidth, or a cable peer with 300 kbps uplink bandwidth. We fix the ratio between DSL peers and cable peers as 4:6. The video is initially stored at an original server with upload bandwidth of 3 Mbps. The request round is 1 second and each peer has a buffer that can store 30 seconds' video. We choose the "Foreman" video sequence with 352x288 spatial resolution at frame rate 30 frames per second and padding the video by another to two-hour long. A MPEG-4 video codec [57] is used to encode the video sequence into a non-scalable bit stream with bit rate 150 kbps. We divide the video into 1-second chunks, thus each chunk has $M = 150$K bits. Among those peers, we randomly choose two who cooperate with each other using the proposed two-player cheat-proof P2P live streaming cooperation strategy. We set $g_1 = 1$, $g_2 = 1, 0.9, 0.8, 0.7$, and every peer claims the lowest bandwidth $W_{min} = 300$kbps.

In our simulations, user 1 always reports accurate private information to user 2,

and user 2 cheats on his/her buffer map information. Among all the chunks that user 2 received, he/she randomly selects p_c percent of them, manipulates his/her buffer map, and tells user 1 that he/she does not have those selected chunks in the buffer. Figure 4.4 shows the utility of user 2 with different gain g_2, where the x axis is p_c and the y axis is the utility v_i. From Figure Figure 4.4, for a given g_2, a higher value of P_c gives the cheating user a lower payoff. In addition, when g2 is small (for example, when g_2=0.7), if the cheating user selects a larger p_c, then he/she receives a zero payoff. That is, cheating cannot help a user increase his/her payoff, but rather reduces his/her utilities. It clearly demonstrates the cheat-proof property of our proposed strategy in in Section 4.1.4. In addition, from our simulations, by cooperating with each other, both peers double the number of chunks that they receive, which is 278 without cooperation and 542 after cooperation. Therefore, users can reconstruct a better-quality video.

4.2 P2P Live Streaming Game

4.2.1 Multi-user Game Model

Next, we will investigate how to stimulate cooperation for all members in peer-to-peer live streaming over heterogeneous and error-prone networks, and analyze users' behavior dynamics. We focus on the scenario that video streaming will keep alive for a relatively long time, and there exist a finite number of users, for example, people watch live Super Bowl over the Internet. Each user will stay in the social network for a reasonably long time, for example, from the beginning to the end of the game. They are allowed to leave

and reconnect to the network when necessary.For each user, uploading chunks to other users will incur some cost, and successfully receiving chunks can improve the quality of his/her video and thus brings some gain. To simplify the analysis, in this section, we assume the video stream is encoded using non-scalable video codec. Therefore, for each user i, each received chunk gives the same gain g_i, whose value is specified by the user individually and independently. As discussed in Section 4.1.2, g_i, the gain of receiving a chunk for the live video, is evaluated by user i by how much he/she wants to watch the video. For instance, g_i should be set to 1 if at this moment, all user i wants to do is watch the live streaming. The more activities user i is doing simultaneously using the network bandwidth, the lower the g_i is. If user i is utilizing lots of his/her upload bandwidth and does not care about the quality of the live video stream, g_i should be set to 0, and user i will not join the P2P live stream social network.

In a real-world social network, some users may be malicious, whose goal is to cause damages to other users. In this chapter, we focus on insider attackers, that is, the attackers also have legitimate identities, and their goal is to prevent the selfish users from getting chunks. In P2P live streaming social networks, there are two ways to attack the system:

1. **Incomplete chunk attack**: The malicious user agrees to send the entire requested chunk to the peer, but sends only portions of it or no data at all. By doing so, the requesting peer wastes his/her request quota in this round, and has to request the same chunk again in the next round.

2. **Pollution attack**: The other kind of attack in peer-to-peer live streaming is pollution [58]. In P2P streaming system, a malicious user corrupts the data chunks,

renders the content unusable, and then makes this polluted content available for sharing with other peers. Unable to distinguish polluted chunks from unpolluted files, unsuspecting users download the polluted chunks into their own buffers, from which others may then download the polluted data. In this manner, polluted data chunks spread through the system.

Instead of forcing all users to act fully cooperatively, our goal is to stimulate cooperation among selfish users as much as possible and minimize the damages caused by malicious users. In general, not all cooperation decisions can be perfectly executed. For example, when a peer decides to send another peer the requested chunk, packets of the chunk may be dropped due to the overloaded routers. It is also possible that the chunk may fail to be completely received in one round due to the significant delay caused by the congested network. In this chapter, we assume that the requesting peer gives up the chunk once it does not arrive in the round, and we use p_{ij} to denote the probability of successful transmission of a chunk from peer i to peer j in one round of τ second. At the beginning of every round, each user will send only one chunk request to one user. Each user will respond to only one request. We assume every chunk request can be received immediately and perfectly.

In order to formally analyze cooperation and security in such peer-to-peer live streaming networks, we model the interactions among peers as the following game:

- **Server**: The video is originally stored at the original streaming server with upload bandwidth W_s, and the server will send chunks in a round-robin fashion to its peers.

- **Players and player type**: There are finite number of users/peers in the peer-to-

peer live streaming social network, denoted by N. Each player $i \in N$ has a type $\theta_i \in \{\text{selfish}, \text{malicious}\}$. Let N_s denote the set of all selfish players and $N_m = N \backslash N_s$ is the set including all insider attackers. A selfish user aims to maximize his/her own payoff, and may cheat other peers if cheating can help increase his/her payoff. A malicious user wishes to exhaust other peers' resources and attack the system.

- **Chunk requesting**: In each round, each player has *one* chunk-request quota, where he/she either *requests a chunk from a peer*, *requests a chunk from the video streaming source*, or *does not request any chunks* in this round.

- **Request answering**: For each player, after receiving a request asking for the upload of a chunk in its buffer, it can either *accept* or *refuse* the request.

- **Cost**: For any player $i \in N$, uploading a chunk to another player incurs cost $c_i = M/W_i\tau$, where W_i is player i's upload bandwidth and $W_i \geq W_{min} \geq M/\tau$, same as in Section 4.1.2.

- **Gain**: For each selfish user $i \in N_s$, if he/she requests a data chunk from another peer j, and if an unpolluted copy is successfully delivered to him/her, his/her gain is g_i where $P_{ji}g_i > c_i$.

- **Utility function**: We first define the following symbols: for each player $i \in N$,

  - $Cr^{(i)}(j,t)$ is the total number of chunks that i has requested from j by time t. Here, j can be either a peer ($j \in N$) or j is the streaming server. $Cr^{(i)}(t) = \sum_{j \in \{N,\, \text{source}\}} Cr^{(i)}(j,t)$ denotes the total number of chunks that i has requested by time t.

– By time t, peer i has successfully received $Cs^{(i)}(j,t)$ chunks from peer j in time (a chunk is received in time if and only if it is received within the same round that it was requested). $Cs^{(i)}(t) = \sum_{j\in\{N,\text{source}\}} Cs^{(i)}(j,t)$ is peer i's total number of successfully received chunks by time t.

– By time t, $C_p^{(i)}(j,t)$ is the total number of polluted chunks that peer i received from peer j. The total number of successively received unpolluted data chunks that peer i received from peer j is $Cs^{(i)}(j,t) - C_p^{(i)}(j,t)$, and each successfully received unpolluted chunk gives peer j a gain of g_i.

– $Cu^{(i)}(j,t)$ denotes the number of chunks that i has uploaded to player j by time t. $Cu^{(i)}(t) = \sum_{j\in\{N,\text{source}\}} Cu^{(i)}(t)$. The cost of uploading each chunk is c_i for peer i.

Let t_f be the lifetime of the peer-to-peer live streaming social network, and $T^{(i)}(t)$ denotes the total time that peer i is in the network by time t. Then, we model the player's utility as follows:

1. For any selfish player $i \in N_s$, its utility $U_s^{(i)}(t_f)$ is defined as

$$U^{(i)}(t_f) = \frac{\left[Cs^{(i)}(t_f) - \sum_{j\in N} C_p^{(i)}(j,t_f)\right] g_i - Cu^{(i)}(t_f)\frac{M}{W_i\tau}}{Cr^{(i)}(t_f)}, \tag{4.16}$$

where the numerator denotes the net profit (i.e., the total gain minus the total cost) that the selfish peer i obtained, and the denominator denotes the total number of chunks that i has requested. This utility function represents the average net profit that i can obtain per requested chunk, which i aims to maximize.

2. For any malicious player $j \in N_m$, its objective is to maximize its utility

$$U_m^{(j)} = \frac{\sum_{i \in N_s} Cu^{(i)}(j,t_f)\frac{M}{W_i\tau} + \sum_{i \in N_s}\left[Cr^{(i)}(j,t_f) - Cs^{(i)}(j,t_f)\right]P_{ji}g_i - Cu^{(j)}(t_f)\frac{M}{W_j\tau}}{T^{(j)}(t_f)}. \tag{4.17}$$

The numerator in (4.17) represents the net damage caused by j: the first term describes the total costs to other peers when sending the requested chunks to the malicious user j; the middle term evaluates other selfish peers' potential loss in gain due to the incomplete chunk attack by peer j; and the last term is peer j's cost by uploading chunks to other peers. We normalize it using the lifetime of peer j, $T^{(j)}(t_f)$. Now, this utility function represents the average net damage that j causes to the other nodes per time unit.

4.2.2 Cheat-Proof and Attack-Resistant Cooperation Stimulation Strategies

Based on the system description in Section 4.2.1, we can see that the multiple player game is much more complicated than the two-person game as in Section 4.1, and pose new challenges. Thus, direct application of the two-player cooperation strategies to multiple player scenarios may not work.

4.2.2.1 Challenges in Multiple User Scenario

For peer-to-peer live streaming networks in heterogeneous Internet traffic environments, user cooperation stimulation has the following challenges:

- **Repeated game model is not applicable.** For example, a peer may request chunks

from different peers at different times to maximize the utility. A direct consequence of such a non-repeated model is that favors cannot be simultaneously granted. This makes cooperation stimulation in peer-to-peer live streaming networks an extremely challenging task.

- **Packet delay is inevitable** in Internet can cause severe trouble. For the two-player cheat-proof cooperation strategy, if the link between users is too busy that some packets of the chunk can not arrive within a round time, the game will be terminated immediately and the performance will be degraded drastically. In addition, the malicious users can claim it was due to the erroneous Internet traffic and pretend to be non-malicious. Distinguishing misbehavior caused by bit errors and packet loss from that caused by malicious intention is a challenging task.

4.2.2.2 Credit Mechanism for Malicious User Detection

To distinguish "intentional" malicious behavior from "innocent" misbehavior caused by packet delay, we introduce the credit mechanism. Addressing the pollution attack, for any two peers $i, j \in N$,

$$Cc^{(i)}(j,t) = Cu^{(i)}(j,t) - C_p^{(j)}(i,t) \tag{4.18}$$

calculates the total number of *unpolluted* chunks that peer i has uploaded to peer j by time t. If the chunk is unpolluted, and is received before its playback time, then the chunk is useful. Note that for a selfish user $i \in N_s$, as discussed in the previous section, he/she has no incentives to intentionally send others polluted data chunks, since doing so will ultimately hurt himself/herself and lower the quality of his/her own video. However,

since peer i cannot identify a chunk as a polluted one until he/she starts decoding and playing that chunk, it is possible that user i *unintentionally* forwards a polluted chunk to other peers. In this chapter, addressing the above issue, we include the term $C_p^{(j)}(i,t)$ in (5.17) and consider the potential unintentional forwarding of polluted data chunks.

Given (5.17), we then define

$$D^{(i)}(j,t) = Cc^{(i)}(j,t) - Cc^{(j)}(i,t) = \left(Cu^{(i)}(j,t) - C_p^{(j)}(i,t)\right) - \left(Cu^{(j)}(i,t) - C_p^{(i)}(j,t)\right), \tag{4.19}$$

which is the difference between the number of *useful* chunks that peer i has sent to peer j and the number of *useful* chunks that peer j uploaded to peer i. Now, similar to the 2-player cooperation-stimulation strategy in Section 4.1.4, we consider the following strategy: each selfish peer $i \in N_s$ limits the number of chunks that he/she sends to any other peer j such that by any time t, the total number of useful(unpolluted) chunks that i has forwarded to j should be no more than $Cu^{(j)}(i,t) - C_p^{(i)}(j,t) + D_{max}^{(i)}(j,t)$, that is,

$$D^{(i)}(j,t) \leq D_{max}^{(i)}(j,t), \quad \forall t \geq 0. \tag{4.20}$$

Here, $D_{max}^{(i)}(j,t)$ is the "credit line" that user i sets for user j at time t. The credit line is set for two purposes: 1) to prevent egoism when favors cannot be simultaneously granted and to stimulate cooperation between i and j, and 2) to limit the possible damages that j can cause to i. By letting $D_{max}^{(i)}(j,t) \geq 0$, i agrees to send some extra, but at most $D_{max}^{(i)}(j,t)$ chunks to j without getting instant payback. Meanwhile, unlike acting fully cooperatively, the extra number of chunks that i forwards to j is bounded to limit the possible damages when j plays non-cooperatively or maliciously.

Player i's goal of setting the credit line is to avoid helping player j much more than

player j helps i in long term's view, and vice versa, since neither of i, j has incentive to send more chunks than the other does. Meanwhile, due to the dynamically changing network conditions, the request rates between i and j may vary from time to time. In this case, the credit line has to be large enough since a small credit line will refuse some requests even when the long-term average request rates between i and j are equal. The ultimate goal of setting the credit line is to make sure that player i and j send asymptotically equal number of unpolluted chunks to each other, and

$$\lim_{t\to\infty} Cc^{(i)}(j,t) = \lim_{t\to\infty} Cc^{(i)}(j,t). \tag{4.21}$$

Combining the definition of $D^{(i)}_{max}(j,t)$ with (4.21), $D^{(i)}_{max}(j,t)$ must satisfy

$$\lim_{t\to\infty} \frac{D^{(i)}_{max}(j,t)}{Cr^{(i)}(t)} = 0, \tag{4.22}$$

which also implies that arbitrarily increasing credit lines cannot always increase the number of accepted requests. (4.22) provides an asymptotic upper bound for $D^{(i)}_{max}(j,t)$. Based on the above analysis, to stimulate cooperation in the first few rounds, $D^{(i)}_{max}(j,t)$ should be large enough in the first few cooperating rounds between user i and j. On the other hand, $D^{(i)}_{max}(j,t)/$[total number of rounds after time t] should be closed to 0 to prevent decreasing the utility of user i. Therefore, when choosing $D^{(i)}_{max}(j,t)$, user i should first estimate the number of remaining rounds for the live streaming, and choose a relatively small number D_{temp}. Then compare D_{temp} with the reciprocal of P_{ij}, so that $D^{(i)}_{max}(j,t)$ should be larger than $1/P_{ij}$ to stimulate the cooperation. A simple solution to this is to set the credit lines to be reasonably large positive constants, as in our simulations in Section 4.4.

4.2.2.3 Malicious User Detection

Malicious attacks, such as the incomplete chunk attack and the pollution attack, exhaust other peers' resources and cause damages to the P2P live streaming system. Thus, it is of critical importance to implement a monitoring system to detect and identify misbehaving users, and a challenging issue is to differentiate "innocent" misbehavior (due to erroneous and congested networks) from "intentional" ones (for example, intentional pollution attacks).

If the credit line is set properly to satisfy (4.22), the damage of the pollution attack can be controlled to 0 asymptotically. Since the pollution attack will not effect honest users' utility by the credit line mechanism, in this section, we propose a malicious user detection algorithm that can differentiate the incomplete information attack to ensure the attack-resistance of the P2P live streaming social network.

P_{ij} is the probability of successful transmitting one chunk within round period τ. Hence when player *i*decides to send a chunk to player j, with probability $1-P_{ij}$, this chunk transmission cannot be completed within one round because of packet dropping or delay caused by high traffic internet. That is, we use a Bernoulli random process to model the unsuccessful transmission of a chunk due to high traffic internet connection. Recall that that $Cu^{(j)}(i,t)$ denote the number of chunks that i has requested from j and j has agreed by time t, and $Cs^{(i)}(j,t)$ is the number of chunks that peer i successfully receives from j in one round. Given the Bernoulli random process, if user j does not intentionally deploy the incomplete chunk attack, based on the Central Limit Theorem [59], for any

positive real number x, we can have

$$\lim_{Cu^{(j)}(i,t)\to\infty} Prob\left(\frac{Cs^{(i)}(j,t)-Cu^{(j)}(i,t)P_{ji}}{\sqrt{Cu^{(j)}(i,t)Pji(1-P_{ji})}}\geq -x\right)=\Phi(x), \tag{4.23}$$

where $\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-t^2/2}dt$ is the Gauss tail function. If user j does not intentionally sends incomplete chunks, (4.23) indicates that when the peer-to-peer live streaming game keeps going and $Cu^{(j)}(i,t)$ is large enough, then $Cs^{(i)}(j,t)-P_{ji}C_u^{(j)}(i,t)$ can be approximated by a Gaussian random variable with zero mean and variance $Cu^{(j)}(i,t)P_{ji}(1-P_{ji})$, that is,

$$Cs^{(i)}(j,t)-Cu^{(j)}(i,t)P_{ji}\sim\mathcal{N}\left(0,Cu^{(j)}(i,t)P'_{ji}(1-P_{ji})\right). \tag{4.24}$$

Therefore, based on (4.24), given a predetermined threshold $h>0$, every selfish peer i can identify peer j as a malicious user by thresholding $Cs^{(i)}(j,t)-Cu^{(j)}(i,t)P_{ji}$ as follows:

$$\begin{aligned} j\in N_m^{(i)}(t) \quad &\text{if and only if} \quad Cs^{(i)}(j,t)-Cu^{(j)}(i,t)P_{ji} \;\leq\; -h\sqrt{Cu^{(j)}(i,t)P_{ji}(1-P_{ji})}, \\ \text{and} \quad j\in N_s^{(i)}(t) \quad &\text{if and only if} \quad Cs^{(i)}(j,t)-Cu^{(j)}(i,t)P_{ji} \;>\; -h\sqrt{Cu^{(j)}(i,t)P_{ji}(1-P_{ji})}, \end{aligned} \tag{4.25}$$

In (4.25), $N_m^{(i)}(t)$ is the set of peers that are marked as malicious by peer i at time t, and $N_s^{(i)}(t)$ is the set of peers that are marked as selfish by peer i at time t. Based on (4.25), if the malicious user is always sending incomplete chunks to other users, then the probability of correctly identify the malicious user (P_d) and the probability of falsely accusing a nonmalicious user as malicious (P_{fa}) can be written as

$$P_d=1-\Phi(h), \quad \text{and } P_{fa}=\Phi(h). \tag{4.26}$$

4.2.2.4 Cooperation-Stimulation Strategies

In reality, the interactions between peers are determined by the Internet topology and the communication pattern in the network. To analyze the effect of internet topology, we define P_{ij} as the probability that peer j successfully receives a chunk from peer i in one round period τ, P_{si} denotes the probability of i successfully receiving a chunk from the streaming server in one round period τ, P_s denotes the percentage of requests that the streaming server can answer in one round. These probabilities, P_{ij}, P_{si}, P_s, can be probed or estimated [27].

Theorem 1 For a selfish peer i, if

$$P_{si} \times P_s > P_{ji}, \quad \forall\, j \in N,\ j \neq i, \tag{4.27}$$

then his/her optimal strategy is to always download the live video from the streaming server and to reject all chunk requests from other peers.

Proof. First, consider the optimal strategy for each round. At each round, peer i has one chunk-request quota by which i can ask a chunk either from the source or one peer $j \in N$. The probability that peer i will successfully receive the requested chunk if i sends request to source is $P_{si} \times P_s$, while the probability that peer i will successfully receive the requested chunk if i sends request to j is $P_{ji} \times$ probability of j agrees to send the chunk. Obviously the probability of j agrees to send the chunk is less than or equal to one, and since (4.27) holds, sending request to the source will give i highest probability of getting the chunk/reward, which is the optimal chunk-request strategy in each round. Therefore, always asking chunks from the source is the optimal chunk-request strategy in the whole game. And since peer i always requests chunks from the original source, it doesn't have

incentive to send any chunks to other peers in the network since it cost $M/W_i\tau$ to send a chunk which decreases peer i's utility as in (4.16). From the above analysis, peer i will always operate non-cooperatively.□

Theorem 1 suggests that if a peer has a very good connection with the original streaming server, which is much better than the connections with all the other peers, then he/she will always refuse to cooperate. Cooperation can not be enforced to these peers. But in real-world case, there are usually very few peers that can meet the above condition since peer-to-peer live streaming social networks are usually very big. Thus, the streaming server is often very busy with low P_s, and makes the condition $P_{si} \times P_s > P_{ji}$ for all $j \in N,\ j \neq i$ in Theorem 1 very difficult to satisfy.

The other extreme scenario is when peer i is has the worst connection with other peers, that is, for every $j \in N,\ j \neq i$, there always exists another peer $k \in N,\ k \neq i, j$ such that $P_{ij} < P_{kj}$. In this scenario, will all the other peers in the network refuse to cooperate with him/her? The answer is no because of the dynamics in peer-to-peer social networks and the assumption of a busy server. Note that in peer-to-peer live streaming, different users have different playback time. If peer i's playback time is earlier than all the other peers in the network, then it is very likely that his/her buffer has chunks that no other peers have, which is the incentive for other peers to cooperate with i under the constraint that $D^{(j)}(i,t) \leq D^{(j)}_{max}(i)$.

4.2.2.5 Multiuser attack-resistant and cheat-proof cooperation strategy

By summarizing the above results, we can arrive at the following cooperation stimulation strategies in peer-to-peer live streaming social networks:

Multiuser attack-resistant and cheat-proof cooperation strategy: *in the peer-to-peer live streaming game, for any selfish peer $i \in N_s$ who does not meet the necessary condition (4.27) of Theorem 1, he/she initially marks every other user $j \in N,\ j \neq i$ as selfish. Then, in each round, i uses the following strategy:*

- *If i has been requested by j to send a chunk, i will accept this request if j has not been marked as malicious by i and (4.20) holds; otherwise, i will reject the request.*
- *When i is requesting a chunk, he/she will send the request to peer j who satisfies*

$$j = \arg \max_{j \in N_s^{(i)}(t), j \neq i} P'_{ji} \tag{4.28}$$

- *Let $1 - \Phi(h)$ be the maximum allowable false positive probability from i's point of view, then, when $Cu^{(j)}(i,t)$ is large enough for any users $j \in N$, i will apply the detection rule (4.25) to detect malicious behavior after each chunk request initiated by i.*

4.2.3 Strategy Analysis under no Attacks

This section analyzes the optimality of the above proposed strategy for peers who do not satisfy the necessary conditions in Theorem 1 when there are no malicious users. We first consider an infinite-lifetime situation with $Cr^{(i)}(t) \to \infty$ as $t \to \infty$, and the finite-lifetime

situation will be discussed later. First, we assume $D_{max}^{(i)}(j,t)$ satisfies (4.22), which is to guarantee at most a finite number of i's requests will be refused by j, and ensure i needs j's help the same as i helps j averagely.

Lemma 1. *In the peer-to-peer live streaming game where some chunks may be dropped or delayed due to high traffic volume in the Internet, for a selfish player j, if all other users follow the multiuser attack-resistant and cheat-proofing cooperation strategy, then playing non-cooperatively and sending only part of the requested chunks will not increase j's payoff.*

Proof. If user j has agreed to upload a chunk to another user $i \in N$, by transmitting only part of the requested chunk will help j reduce his/her cost. However, even though j agrees to upload the chunk, it does not count as a successfully received chunk. In addition, player i follows the multiuser attack-resistant and cheat-proof cooperation strategy, and always tries to let

$$\lim_{t\to\infty} Cs^{(i)}(j,t) \geq \lim_{t\to\infty} Cs^{(i)}(j,t). \tag{4.29}$$

Since (4.22) is satisfied, thus by sending partial of the requested chunk, player j loses one chance to request a chunk from player i. To get this one-chunk-request chance back, player j has to send another chunk completely and successfully to player i. Therefore, intentionally sending partial information of the requested chunks cannot bring any gain to player j. □

Lemma 2. *For a selfish peer $i \in N_s$ in the peer-to-peer live streaming game with no malicious attackers, once i has received a chunk request from another node $j \in N$, if (4.20) holds and if j follows the multiuser attack-resistant and cheat-proofing cooperation*

strategy, then accepting the request is always an optimal decision from player i's point of view.

Proof. From player i's point of view, if (4.22) is satisfied, agreeing to send the requested chunk will not introduce any performance loss, since the average cost of helping j goes to zero when $t \to \infty$. Meanwhile, refusing the request may cause $D^{(j)}(i,t) > D^{(j)}_{max}(i,t)$ and thus forbids user i to request chunks from player j in the future. Therefore, accepting the request is an optimal decision. □

Lemma 3. *In the peer-to-peer live streaming game with no malicious attackers, a selfish peer $i \in N_s$ has no incentive to cheat on his/her buffer map information.*

Proof. From player i's point of view, cheating on his/her buffer information will prevent other peers from requesting chunks from him/her, and thus will decrease the total number of chunks he/she needs to upload ($Cu^{(i)}(t)$). However, since other users always enforce (4.22) and $Cu^{(j)}(i,t) + D^{(j)}_{max}(i,t) < Cu^{(i)}(j,t)$, decreasing $Cu^{(i)}(t)$ will also decrease the chance of getting chunks from other peers and lower player i's overall payoff, similar to the two-player game in Section 4.1.2. Therefore, selfish peers have no incentive to cheat on buffer information. □

Theorem 2. *In the peer-to-peer live streaming game without malicious attackers, if all the selfish players who do not satisfy the necessary conditions in Theorem 1 follow the multiuser attack-resistant and cheat-proofing cooperation strategy forms a equilibrium with following properties: subgame perfect, cheat-proof, and if $0 < \lim_{t\to\infty} \frac{Cr^{(i)}(t)}{Cr^{(j)}(t)} < \infty$ for any $i, j \in N$, this equilibrium is also strongly Pareto optimal.*

Proof. : **1) Cheat-proof**: Similar to the analysis of the two-person game in Section 4.1, since no private information is involved in the game and Lemma 3 says that selfish

users have no incentive to cheat on buffer information, we can conclude that the proposed cooperation-stimulation strategy is cheat-proofing.

2) Nash Equilibrium: To prove that this strategy profile forms a subgame perfect equilibrium, note that this multiuser game can be decomposed into many two-player subgames. Therefore, we only need to consider the two-player subgame between player i and j. Suppose that player i does not follow the above strategy: either i refuses to send chunks to player j when (4.20) is satisfied; or i intentionally sends only part of the chunk requested by player j; or i sends more chunks than it should for player j, that is, j agrees to send the requested chunks even (4.20) is not satisfied. First, from Lemma 1 and 2, neither refusing to sending chunks for other players when (4.20) is satisfied nor intentionally sending incomplete chunks will give player i any performance gain. Secondly, sending many more chunks (i.e., more than $D^{(i)}_{max}(j,t)$) than player j has sent to i will not increase player i's own payoff either. This is because according to the assumption of credit line selections, j will always cooperate with i since j has sent chunks less than i. Therefore, giving j more favor will only cost i more bandwidth. Based on the above analysis, we can conclude that the above multiuser attack-resistant and cheat-proofing cooperation strategy forms a Nash equilibrium.

3) Subgame perfectness: In every subgame of the equilibrium path, the strategies are: if player j is marked malicious by peer i, player j will play non-cooperatively forever, which is a Nash equilibrium; otherwise, player j follows the multiuser attack-resistant and cheat-proofing strategy, which is also a Nash equilibrium. Therefore, the proposed cooperation-stimulation strategy is subgame perfect .

4) Strong Pareto optimality: From the selfish user's utility function in (4.16), a player

i can either try to increase $Cs^{(i)}(t)$ or decrease $Cu^{(i)}(t)$ to increase his/her own payoff. However, from the above analysis, further decreasing of $Cu^{(i)}(t)$ will reduce other peers' successfully received useful chunks and therefore lower their payoff. In order to increase his/her payoff, the only thing that player i can do is to increase $\lim_{t\to\infty} Cs^{(i)}(t)/Cr^{(i)}(t)$, which means that some other players will have to send more chunks to player i. Since all $Cr^{(i)}(t)$s are in the same order, increasing $\lim_{t\to\infty} Cs^{(i)}(t)/Cr^{(i)}(t)$ (and thus improving player i's payoff) will definitely decrease the other players' payoff. Therefore, the above strategy profile is strongly Pareto optimal. □

Until now, we have mainly focused on the situation that the game will be played for an infinite duration. In most situations, a peer will only stay in the network for a finite period of time, for example till the end of the video streaming. Then, for each player i, if $D^{(i)}_{max}(j,t)$ is too large, he/she may have helped other users much more than his/her peers have helped i. Meanwhile, if $D^{(i)}_{max}(j,t)$ is too small, he/she may lack enough peers to send chunks to him/her. How to select a good $D^{(i)}_{max}(j,t)$ is a challenging issue. Section 5.5 will study the trade-off between the value of $D^{(i)}_{max}(j,t)$ and the peers' utility through simulations. It is shows there that under given simulation scenarios, a relatively small $D^{(i)}_{max}(j,t)$ value is good enough to achieve near-optimal performance, when compared to setting $D^{(i)}_{max}(j,t)$ to be infinity. Here, it is also worth mentioning that the optimality of the proposed strategies cannot be guaranteed in finite-duration scenarios. However, we will show in the simulation results that the performance our cheat-proof and attack-resistant cooperation strategies is very closed to optimal.

4.2.4 Strategy Analysis under Malicious Attacks

In this section, we focus on the following two widely used attack models, the incomplete chunks attack and the pollution attack, and analyze the performance of the proposed cooperation-stimulation strategy when there exist malicious users. To simplify our analysis, we assume that $W_i = W$, $g_i = g$, and $g\frac{M}{W\tau} < \infty$ for all $i \in N$.

Pollution attack: We first study the performance of the proposed strategy under the pollution attack. By always accepting selfish users' requests and sending polluted chunks to the selfish nodes, the malicious attackers can waste the selfish users' quota and prevent them from obtaining the gain of receiving useful chunks in that round. Note that every selfish user $i \in N_s$ forces $D^{(i)}(j,t) \leq D^{(i)}_{max}(j,t)$, calculates $D^{(i)}(j,t)$ as in (4.19), and does not include the polluted chunks in $D^{(i)}(j,t)$. Thus, for every selfish peer i, the damage cased by one pollution attacker is upper bounded by $D^{(i)}_{max}(j,t)g$. Since $g < \infty$, as $t \to \infty$,

$$\lim_{t\to\infty} \frac{D^{(i)}_{max}(j,t)g}{Cr^{(i)}(t)} = 0, \tag{4.30}$$

and therefore, the overall damage due to pollution attacks becomes negligible.

Incomplete chunk attack: By sending incomplete chunks to others, malicious users inject trash traffic into the network and waste other peers' limited upload bandwidth. With the proposed attacker detection strategy in (4.25), for a malicious attacker to maximize the damages to the system, always sending incomplete chunks may not be a good strategy since it can be easily detected. Instead, to avoid being detected, attackers should selectively send incomplete chunks and send complete chunks in other time. According to the multiuser attack-resistant and cheat-proofing cooperation strategy in Section 4.2.1, peer j identifies i as malicious if $Cs^{(j)}(i,t) - Cu^{(i)}(j,t)P_{ij} \leq -h\sqrt{Cu^{(j)}(i,t)P_{ij}(1-P_{ij})}$.

Assume that by time t, user i has agreed to upload a total of n chunks to user j. Therefore, to avoid being marked as malicious by j, i has to successfully forward at least $nP_{ij} - h\sqrt{nP_{ij}(1-P_{ij})}$ complete chunks, and the maximum number of incomplete chunks that i can send to j is upper bounded by $n(1-P_{ij}) + h\sqrt{nP_{ij}(1-P_{ij})}$. Note that among these $n(1-P_{ij}) + h\sqrt{nP_{ij}(1-P_{ij})}$ incomplete chunks, $n(1-P_{ij})$ of them are dropped or delayed by the network due the high Internet traffic volume, and the actual number of intentional incomplete chunks sent to j by i is bounded by $h\sqrt{nP_{ij}(1-P_{ij})}$. Therefore, for user j, the extra damaged caused by attacker i's intentional malicious attack is upper bounded by $h\sqrt{nP_{ij}(1-P_{ij})}g$. Furthermore, to avoid being identified as malicious, attacker i has to successfully forward at least $nP_{ij} - h\sqrt{nP_{ij}(1-P_{ij})}$ complete chunks to user j, which costs attacker i a utility of $\left[nP_{ij} - h\sqrt{nP_{ij}(1-P_{ij})}\right]\frac{M}{W\tau}$. Thus, following (4.17), the utility that attacker i receives from intentionally sending incomplete chunks is at most $h\sqrt{nP_{ij}(1-P_{ij})}(\frac{M}{W\tau}+g) - n(1-P_{ij})\frac{M}{W\tau}$. Since for any real positive h,

$$\lim_{t\to\infty} \frac{h\sqrt{nP_{ij}(1-P_{ij})}(\frac{M}{W\tau}+g)}{n(1-P_{ij})\frac{M}{W\tau}} = 0, \tag{4.31}$$

selectively sending incomplete chunks can bring no gain to the attackers if they want to remain being undetected. In other words, if the game will be played for an infinite duration, sending incomplete chunks attack cannot cause damages to selfish nodes.

In summary, when the multiuser attack-resistant and cheat-proofing strategy is used by all selfish users, malicious attackers can only caused limited damage to the system. Further, the relative damage caused by the incomplete chunk attack will asymptotically approach zero when the game will be played for an infinite duration of time. Therefore, except some false alarm of identifying selfish users as malicious, selfish players' overall

payoff will not be affected under attacks. From the above analysis, we can also see that no matter what objectives the attackers have and what attacking strategies that they use, as long as selfish peers apply the multiuser attack-resistant and cheat-proofing cooperation strategy, the selfish users' payoff and the overall system performance can be guaranteed.

Optimal attacking strategy: Based on the above analysis on the pollution attack and the incomplete chunk attack, we can conclude that, for the infinite-duration game, an attacker j's overall payoff is upper bounded by

$$U_m^{(j)} \leq \lim_{t\to\infty} \sum_{i\in N_s} \frac{D_{max}^{(i)}(j,t)}{t} g, \tag{4.32}$$

provided that all selfish users follow the multiuser attack-resistant and cheat-proof cooperation strategy. This upper bound can be achieved by the following *optimal attacking strategy in infinite game model*: in the peer-to-peer live streaming game, upon receiving a request an attacker $j \in Nm$ should always reject the requests; the attackers should always send requests to selfish users, until they do not agree to help.

When the game will only be played for a finite period of time, the above attacking strategy is not optimal any more. In addition to the pollution attack, the attackers can also send incomplete chunks without being detected. This is because the malicious attacker detection algorithm in Section 4.2.2.3 requires that the game has been played for a long time and peer i and j have interacted for a large number of times to provide an accurate estimation, and it will not be initiated unless $Cu^{(j)}(i,t)$ is large enough to avoid high false alarm rate. In such a scenario, different from the asymptotic analysis in (4.31), selfish users' performance will be degraded because of the incomplete chunk attack. However, in this chapter we focus on the scenario the game will be played for a reasonably long

time. Thus the users would have enough rounds to interact with each other and correctly estimate the statistics of chunk transmission, the relative damage is still insignificant.

4.3 P2P Live Streaming Game With Multiple Layered Coding

The previous section discussed the cheat-proofing and attack-resistant multiuser peer-to-peer live streaming cooperation-stimulation strategy with non-scalable video coding. In this section, we will extend the cooperation strategy to the scenario with layered video coding, where different chunks may belong to different layers and thus have different gain to the peers. In this scenario, an important issue is to schedule the chunk requests to maximize each peer's utility. We first investigate the chunk-request algorithm for a two-person P2P live-streaming social network that optimizes three different video quality measures in Section 4.3.2. We then propose a two-person chunk request algorithm considering tradeoff between these measures and extend it to N-person case. Then we will discuss the request-answering strategy when a peer i receives more than one chunk requests at one round, and propose a cheat-proofing and attack-resistant cooperation strategy for P2P live streaming social networks.

4.3.1 P2P Live Streaming with Scalable Video Coding

In P2P live streaming social networks, peers belong to different domains with different upload/download bandwidth, where scalable video coding is widely adopted to accommodate

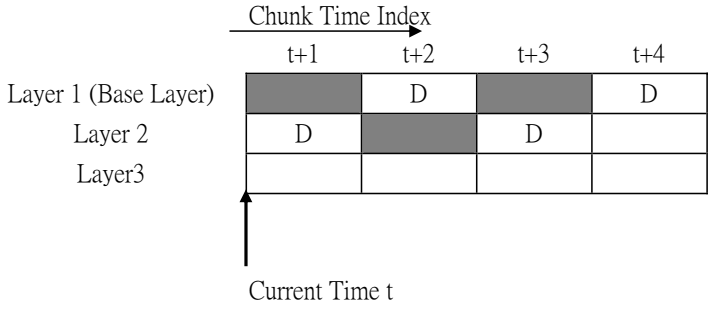


Figure 4.5: Buffer map at a given time t.

heterogenous networks [27], [60]. [27] shows that layered video coding provides higher quality of service in peer-to-peer live streaming social networks than multiple description coding (MDC) [61], thus we only consider the layered video coding. It decomposes the video sequence into different layers of different priority. The base layer contains the most important information of the video and is received by all users, and the enhancement layers gradually refine the reconstructed sequence at the decoder's side.Although scalable video coding provides service depending on peers' bandwidth capacity, it also has its unique challenges when used in P2P live streaming social network: the importance of different layers is unequal since higher layers can not be decoded without successful decoding of lower layers. Therefore, in a P2P live streaming social network with scalable video coding, chunk-request algorithms need to assign higher priorities to the lower layers than to the higher layers.

In this chapter, we encode a video into L layers, and assume that the bit rate of every layer is the same B_s bits/second. We further divide each layer into layer chunks (LCs) of τ seconds. Figure 4.5 shows an example of the buffer map at one user's end. The grey blocks represent the chunks in buffer, while the white blocks denote the chunks that are

not in the buffer, and 'D' stands for layer chunks that are directly decodable after arriving. For example, this user only has the chunks in base layer with time index t+1 and t+3, and the chunk in layer 2 with time index t+2 in buffer. A chunk is *decodable* if and only if all the lower layers in the same chunk time have been decoded correctly.

For user j, we define $a^{(j)}(t)$ as the number of decodable layer chunks at time t. For example, in the example in Figure 4.5, $a^{(j)}(t+1) = 1,\ a^{(j)}(t+2) = 0,\ a^{(j)}(t+3) = 1,$ and $a^{(j)}(t+4) = 0$. Let $T^{(j)}$ denote the duration that peer j is in this network, then we define $A^{(j)} =< a_1^{(j)}, a_2^{(j)}, ...a_{T^{(j)}}^{(j)} >$ as a vector containing all the $\{a^{(j)}(t)\}_{\{t=1,2,...,T^{(j)}\}}$. $N^{(j)}$ is the number of all (decodable and successfully received non-decodable) chunks peer j receives during his stay in the P2P live streaming social networks.

4.3.2 Video Quality Measure

This chapter focuses on investigating the best chunk-request strategy for each user in the peer-to-peer live streaming social network to optimize his/her own received video quality. In the literature, there are many video-quality measures. In this chapter, we consider the following three popular criteria to evaluate our algorithms:

Chunk Decodable Rate: Every member in the P2P live-streaming social network has stringent bandwidth available, and every peer wants to use it as efficiently as possible. The chunk decodable rate $R^{(j)}$ of peer j measures the bandwidth-efficiency of the chunk-request algorithm, and it is defined as

$$R^{(j)} \triangleq \frac{\sum_{i=1}^{T^{(j)}} a_i^{(j)}}{N^{(j)}}. \tag{4.33}$$

Video Smoothness: Intuitively, a video stream with nearly constant quality will be more

pleasant to view than one with large swings in quality. Video smoothness measure (S) is defined as follow:

$$S^{(j)}(A) \triangleq \sum_{i=2}^{T^{(j)}} |(a_i^{(j)} - a_{i-1}^{(j)})|. \tag{4.34}$$

where $|\cdot|$ is the absolute value operator. $S^{(j)}(A)$ increases when the variance of $\{a^{(j)}(t)\}$ goes up, and decreases when the difference between adjacent $\{a^{(j)}(t)\}$ is lowered. To improve the quality and maximize the smoothness of the received video, user j should request the chunks to minimize $S^{(j)}(A)$.

Video Discontinuity Ratio: Discontinuity ratio $\alpha^{(j)}$ of peer j is defined as the percentage of times that a video is undecodable and unplayable. In a scalable video coding scheme, if all frames in the base layer are available, then the video is decodable and playable. Note that $a^{(j)}(t)$ stands for the number of chunks that is decodable at chunk time i. Therefore, if $a^{(j)}(t) = 0$, peer j's video is unplayable at chunk time i. $\alpha^{(j)}$ is defined as:

$$\alpha^{(j)} \triangleq \frac{\sum_{i=1}^{T^{(j)}} \mathbf{U}(a_i^{(j)})}{T^{(j)}}, \tag{4.35}$$

where $\mathbf{U}(a_i^{(j)}) = 1$ when $a_i^{(j)} > 0$, otherwise $\mathbf{U}(a_i^{(j)}) = 0$.

4.3.3 Optimal Chunk-Request Algorithms

In this subsection, we will propose three optimum chunk-request algorithms subject to the three video quality measures discussed in the previous section.

- **Maximizing Chunk-Decodable Rate:** We first discuss the chunk-request algorithm which aims to maximize the chunk decodable rate. According to the definition of chunk-decodable rate in (4.33), chunks that are not decodable do not give

any gain to the player, thus gain of receiving the requested chunk $LC(t',l)$ for player j is

$$g_j = \begin{cases} g & \text{if } LC(t',l) \text{ is decodable} \\ 0 & \text{if } LC(t',l) \text{ is not decodable,} \end{cases} \tag{4.36}$$

where $g > 0$ is a constant, t' is the time index of the requested layered chunk and l stands for the layer index of the requested chunk. Therefore, maximizing payoff function in (4.1) is equivalent to making $g_i = g$, and it is to always requesting chunks that are directly decodable after arriving. In the example in Figure 4.5, at the current state, requesting any one of the "D" chunks, LC(t+1,layer 2), LC(t+2,layer 1), LC(t+3,layer 2), and LC(t+4,layer 1), will maximize the player's payoff.

- **Maximizing Video Smoothness:** If the player concerns more about the video smoothness as defined in (4.34), the gain of receiving a requested chunk $LC(t',l)$ for player j is defined as the increment of smoothness after receiving the requested chunk:

$$g_j = \begin{cases} \sum_{i=t_0}^{i=t_0+L-1} |a_i^{(j)} - a_{i-1}^{(j)}| - |a_i'^{(j)} - a_{i-1}'^{(j)}| & \text{if } LC(t',l) \text{ is decodable} \\ 0 & \text{if } LC(t',l) \text{ is not decodable,} \end{cases} \tag{4.37}$$

where $a_i'^{(j)}$ is the number of decodable layers in chunk time i after receiving the requested chunk $LC(t',l)$, and t_0 is the current playback time. The first term of the summation, $|a_i^{(j)} - a_{i-1}^{(j)}|$ represents the difference between the number of decodable layers in chunk time i and than in $i-1$, hence $\sum_{i=t_0}^{i=t_0+L-1} |a_i^{(j)} - a_{i-1}^{(j)}|$ denotes the smoothness of the buffer map if the chunk LC(t',l) is not received. Similarly, $\sum_{i=t_0}^{i=t_0+L-1} |a_i'^{(j)} - a_{i-1}'^{(j)}|$ denotes the video smoothness if the requested

chunk $LC(t',l)$ is successfully received. Therefore, to maximize the video smoothness, player j should choose the decodable chunk that maximizes the difference $\sum_{i=t'}^{i=t'+1} |a_i^{(j)} - a_{i-1}^{(j)}| - |a_i^{\prime(j)} - a_{i-1}^{\prime(j)}|$ (with maxima greater than 0). If the maxima is less than 0, the peer should always choose undecodable chunks. Using the buffer map in Figure 4.5 as an example, the peer should request $LC(t+4, layer1)$.

- **Minimizing Video Discontinuity Ratio:** If the peer wants to minimize video discontinuity ratio, the base layer is the most important and every chunk in base layer has equal importance according to the discontinuity definition in (4.35). Therefore, the gain of receiving a requested chunk $LC(t',l)$ for player j should be

$$g_j = \begin{cases} g & \text{if} \quad l = 1 \\ 0 & \text{if} \quad l \neq 1. \end{cases} \tag{4.38}$$

  To maximize g_j, the peer should request chunks in base layer. For the example in Figure 4.5, requesting either LC(t+2, layer 1) or LC(t+4, layer 1) will maximize g_j.

The above three algorithms use different video quality measures defined in Section 4.3.2 and select different chunks to maximize each individual criteria. To address the tradeoff between different video quality measure, we combine the above three chunk-request algorithms as follows.

Step 1: For user j, for each chunk $LC(t',l)$ that is not in j's buffer but is available at other peers' buffers, user j assign a score $SC(t',l)$ as follows:

- j first assigns an original score $SC(t',l) = ((t+L)-t')/L$ to the chunk $LC(t',l)$, where t is the current time and L is user j's buffer size. It addresses the stringent

time constraint in video streaming, and gives the chunk $LC(t',l)$ a higher score (thus higher priority for requesting) when it is closer to the playback time.

- If $LC(t',l)$ is decodable after arriving, then the score is updated as $SC(t',l) = SC(t',l) + w_1$.
- If $g_2 = \sum_{i=t'}^{i=t'+1} |a_i^{(j)} - a_{i-1}^{(j)}| - |a_i'^{(j)} - a_{i-1}'^{(j)}| > 0$, then j updates $SC(t',l) = SC(t',l) + w_2 g_2$.
- If $l = 1$, then $SC(t',l) = SC(t',l) + w_3$.

Here, $w_1 \geq 0, w_2 \geq 0, w_3 \geq 0$, $w_1 + w_2 + w_3 = 1$ are the weighs that the peer can adjust depending on the importance of each video-quality measure.

Step 2: Then, for each $LC(t',l)$ that is not in j's buffer but is available at other peers' buffers, let $\Omega(t',l)$ be the set of all users that who are not identified as malicious by user j, those who satisfy (4.20), and those who have $LC(t',l)$ in their buffers. Then, user j further updates the score of each chunk $LC(t',l)$ as

$$SC(t',l) = P_{kj} SC(t',l), \quad \text{where} \quad P_{kj} = \max_{u \in \Omega(t',l)} P_{uj}. \tag{4.39}$$

Step 3: Finally, user j selects the chunk with the highest score, that is, $(t^*, l^*) = \arg\max_{\{t',l\}} SC(t',l)$, and requests the chunk $LC(t^*,l^*)$ from peer k who gives the highest successful transmission probability among all peers in $\Omega(t^*,l^*)$, that is, $k = \arg\max_u P_{uj}, u \in \Omega(t^*,l^*)$.

Since there is no algorithm bring optimal for all the three video quality measures, each peer can choose weights w_1, w_2, w_3 by themselves depending on which video-quality it concerns most.

4.3.4 Request-Answering Algorithm

According to our analysis in Section 4.2.2.4, selfish users who do not satisfy the conditions of Theorem 1 should not reject chunk requests from other selfish peers, some peers may receive several chunk requests in a single round while our P2P environment assume that every user can upload at most one chunk per round. Thus we need a request-answering algorithm to address the above issue.

The peer-to-peer live streaming social network will last till the end of the video and has finite life time, selfish peers tend to consider the contributions from other peers when choosing which request to answer. This situation will encourage the selfish users to be always cooperative in the finite time model. Let $N_r^{(i)}(t) \subseteq N_s^{(i)}(t)$ be the set of users who send a chunk request to peer i in round t and all users in $N_r^{(i)}(t)$ are not marked as malicious by peer i, and also satisfy (4.22). We propose the following request-answering algorithm: for every selfish peer i, when he/she receives multiple chunk requests, he/she randomly chooses one peer j with probability

$$P^{(i)}(j,t) = \frac{(Cs^{(i)}(j,t)+\varepsilon)^{\gamma_i}}{\sum_{k\in N_r^{(i)}(t)}(Cs^{(i)}(k,t)+\varepsilon)^{\gamma_i}}, \tag{4.40}$$

where ε is a small number that gives newcomers who have not sent any chunks to peer i a chance to start cooperation. γ_i is a parameter that controls the sensitivity of peer i to other peers' contribution. If $\gamma_i = 0$, every peer sent a request to peer i has the same probability of being answered. On the contrary, if $\gamma_i \to \infty$, the request from peer who has send most chunks to peer i will definitely be answered.

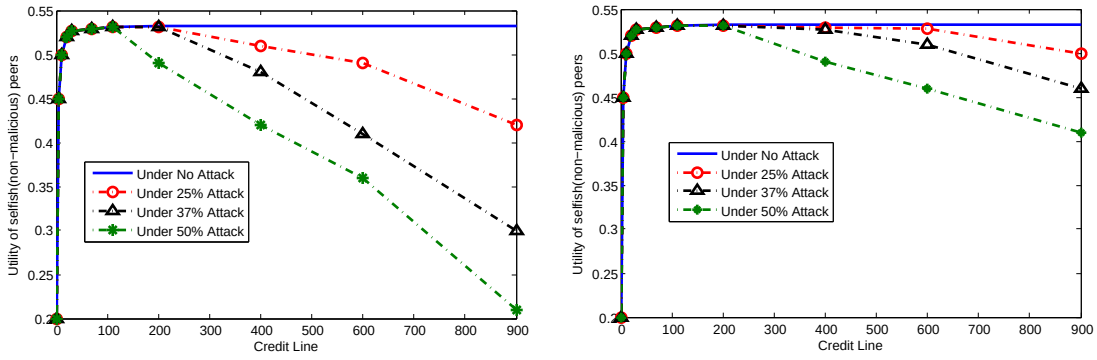
4.3.5 P2P Live Streaming Cooperation Strategy with Layered Video Coding

From the above discussion, the **P2P live streaming cooperation strategy with layered video coding** is as follows: *for any selfish node $i \in N_s$ who does not meet the necessary conditions of Theorem 1, initially i marks every other nodes $j \in N,\ j \neq i$ as selfish. Then, in round t, i uses the following strategy:*

- *In the chunk-requesting stage, i chooses its own (w_1, w_2, w_3), applies the chunk-request algorithm in Section 4.3.3, and sends one chunk request to one peer in $N_s^{(i)}(t)$.*
- *In the request-answering stage, i first identifies the selfish peers that satisfies (4.22). Then, i chooses a peer j among them based on the probability distribution in (5.25), and agrees to send the requested chunk to j.*
- *Let $1-\Phi(h)$ be the maximum allowable false positive probability from i's point of view, then, as long as $Cu^{(j)}(i,t)$ is large enough for any node $j \in N$, i applies the malicious user detection rule (4.25) after each chunk request that is initiated by i.*

4.4 Simulation Results

In our simulation, there are 200 DSL peer with 768 kbps uplink bandwidth and 300 cable peer with 300 kbps uplink bandwidth. The video is initially stored at an original server with upload bandwidth 3 Mbps. We choose the "Foreman" video sequence (352x288) resolution with frame rate 30 frame/sec and by attaching duplicated copies to the original



(a) 3-layer video coding with τ= 0.4 second (b) 4-layer video coding with τ= 0.2 second

Figure 4.6: Selfish peers' performance under proposed strategies with and without attack.

video, make it into a 60 minutes video. Each user has a buffer with length 30 seconds. To exam the influence of different parameters on the performance of the proposed cooperation strategies, we run the simulations under two settings: First, we let the round duration τ is 0.4 second resulting in 9000 rounds in total for the P2P live streaming social network, and the video is coded into 3-layer bitstream with 50 kbps per layer. Then the video is divided into 1 second layered chunks, thus chunk size M= 50 kbits. In our second simulation setup, we let τ=0.2 second and the total number of rounds is 1.8×10^4. The video is encoded into 4-layer bitstream with 37.5kbps per layer. Each chunk is of 1 second length and includes M-37.5K bits. We set the score weighing as $w_1 = 3/6, w_2 = 1/6, w_3 = 2/6$ and the malicious peers can either mount attack by sending incomplete or polluted chunks. The non-malicious (selfish) peers follow the cheat-proof and attack-resistant cooperation strategies in Section 4.3.5.

We first study how different credit lines can affect cooperation stimulation. Figure 4.6 demonstrates the relationship between the credit line when the percentage of attackers are 0, 25%, 37% and 50%, respectively. The attackers are chosen randomly from all the

500 peers. Selfish peers follow the attack-resistant and cheat-proof cooperation strategy in Section 4.3.5, and the attackers follow the attack strategy in Section 4.2.4. From these results, we can see that, in both simulation setups, when the credit line is over 50, the selfish nodes' payoffs are saturated. As the credit line keeps increasing, selfish nodes' utilities start to decrease very fast under attack. The selfish users' utilities remain the same if there are no attackers presented. It is clear from (4.32) that the maximum damage attackers can cause is linearly proportional to the credit line, while total number of rounds is 9000, when credit line is larger than 120 and 50% attackers, by (4.32), the damages are no longer negligible. Also, figure 4.6 suggests that setting credit line of 50 is an optimal choice for both simulation settings since it stimulates the cooperation to the maximum degree. Nevertheless, arbitrarily increasing credit line is dangerous for the selfish users since they do not know how many malicious users are in the network.

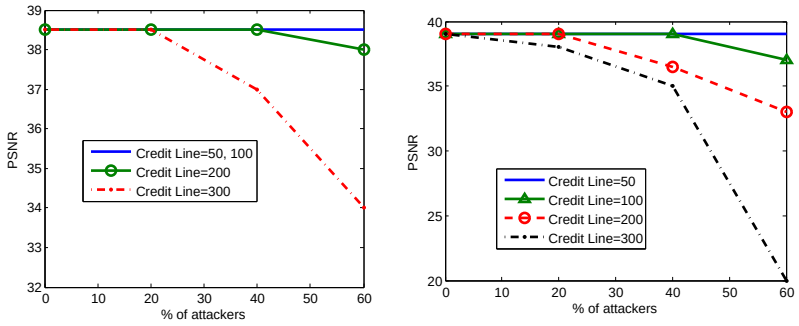
Next, we examine the robustness of our cooperation strategies against attackers and free-riders in terms of PSNR. Since from Fig. 4.6, both simulation settings give similar trends, here we use simulation setting 1 to demonstrate the robustness. Also, to show how the total number of users effects the optimal credit line, we test our proposed cooperation schemes on 500 users and 1000 users with fixed ratio between cable and DSL peers (3:2). We let the credit line equals to 50, 100, 200, or 300, respectively. Selfish peers follow the cooperation strategy in Section 4.3.5. And the malicious peers are randomly selected and follow the optimal attack strategy in Section 4.2.4. Figure 4.7(a) and (b) show the PSNR of a selfish user's video versus the percentage of attackers with different credit lines and different number of users. It is clear that when the credit line is chosen correctly, and is around 50, our cooperation strategies is attack-resistant in both cases. Even the credit line

is too large, around 100, the PSNR of selfish users' video does not degrade too much even there are 60% malicious. From the above discussion, we can conclude that the optimal credit line is the value that just stimulates the cooperation, which should be around several dozen. If there are fewer users in the network, or the total number of rounds is larger, the range of attack-resistant credit line is larger. Although there is no explicit way to choose the credit line, in general, a credit line between 50 an 100 will simulate cooperation among selfish users, resist cheating behavior, and give good performance.

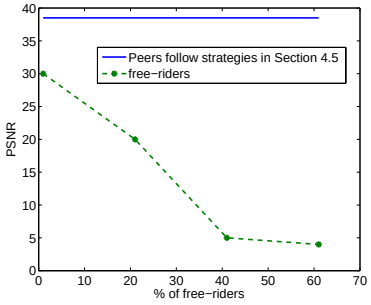
Figure 4.7(c) shows the video quality (PSNR) of peers who follows our cooperation strategy with 500 users in Section 4.3.5 and the free-riders versus % of free riders. The credit line in Figure 4.7(b) is 50. It is clear that there is no incentive for the peers to be free riders since their video quality is very bad, also our attack-resistant and cheat-proof cooperation strategy guarantees the peers' quality of service.

4.5 Chapter Summary

In this chapter, we investigate cooperation stimulation in P2P live streaming social networks under a game theoretic framework. Besides selfish behavior, possible attacks have also been studied, and attack-resistant cooperation stimulations have been devised which can work well under various traffic network and hostile environments. An illustrating two-player game is studied, and different optimality criteria, including Pareto-Optimal, proportional fairness and absolute fairness is performed to refine the obtained Nash Equilibriums. Finally, a unique Nash equilibrium solution is derived, which states that, in the two-person live streaming game, a node should not help its opponent more than its



(a) 500 users (b) 10^3 users



(c) versus free-riders

Figure 4.7: Selfish peers' video quality (PSNR) versus the percentage of attackers and free-riders with 500 users

opponent has helped it.

The results are then extended to stimulate multiuser live streaming, and combing with the chunk-request and request-answering algorithm, a fully-distributed attack-resistant and cheat-proof cooperation stimulation strategy has been devised for P2P live streaming social networks. Simulation results have illustrated that the proposed strategies can effectively stimulate cooperation among selfish peers in internet with various traffic and hostile environments, and the chunk-request algorithm with tradeoffs performs the same as optimal algorithms when the percentage of attackers is lower than 20%.

Chapter 5

Cooperation Stimulation Strategies for Peer-To-Peer Wireless Live Video-Sharing Social Networks

In a wireless live-streaming system, all users directly download the video chunks from the server in the Internet. However, all users share the same link through the access point to the Internet and each user has different playback time and ask for different chunks at the same time. Also there are other users in the wireless network accessing Internet simultaneously. Thus the link might be busy and some chunks can not be received by the end users in time for the playback time. Furthermore, many of the users in the wireless networks have high mobility. Therefore, they would change physical positions from time to time and the quality of network connections may be unstable. All these factors motivate user stimulation in wireless live-streaming social networks to cooperate with each other.

In the literature, the work in [62] proposed an auction-based mechanism for wireless peer-to-peer (P2P) file sharing, and the work in [63] studied the capacity of user-cooperation in wireless network. The live streaming over wireless networks has not been studied. In this chapter, we focus on designing cooperation stimulation strategies for wire-

less live streaming social networks. We first model the cooperation between two users as a Bayesian game and investigate the Bayesian-Nash equilibria. Then, we address the issue of cooperation stimulation among multiple users and investigate cheat-proof and attack-resistant incentive mechanisms. We consider the pollution attack, incomplete-chunk attack, and handwash attack in our model. Our proposed cheat-proof and attack-resistant mechanism rewards users who contribute more with more video chunks (and thus better quality). It includes a request-answering algorithm for the data supplier to upload more to the peers from whom he/she downloads more, and a chunk-request algorithm for the requestor to address the tradeoffs among different quality measure and to optimize the reconstructed video quality.

The rest of this chapter is organized as follows. Section 5.1 introduces the wireless live-streaming system model and the two-player game-theoretical framework. Section 5.2 studies the two-player game and the equilibria. In section 5.3, a cheat-proof and attack-resistant strategy with trust modeling is proposed to stimulate user cooperation among all users in P2P wireless live streaming. Two more issues of wireless live video-sharing, multiple-layered coding and broadcasting nature of wireless channels, are discussed in Section 5.4, and the final wireless live video-sharing cooperation strategy that incorporate these two issues is also studied. Section 5.5 shows simulation results to evaluate the performance of the proposed strategies. Finally, Section 5.6 concludes this chapter.

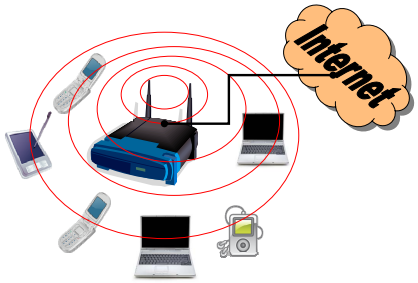


Figure 5.1: Illustration of a wireless live-streaming social network

5.1 System Model and Two-Player Game

In this section, we first describe the model of wireless live streaming systems and how two users in a wireless live streaming social network cooperate with each other. We then define the payoff function and introduce the game-theoretic framework of user dynamics.

5.1.1 Wireless Live Streaming Model

Figure 5.1 shows the architecture of a wireless video live-streaming social network. The wireless network service is provided by an access point connected to the Internet. The video bit stream is divided into media chunks of M' bits in the original server, and are channel-coded to M bits, which is equivalent to t-second piece. All chunks are available at the streaming server in the Internet. Here we assume that there is a dedicated channel of bandwidth BHz for user cooperation and this channel is different from the channel between users and the access point. We assume that the channel for cooperation between users is symmetric and is a slow fading channel with additive white Gaussian noise with

variance σ_n^2. Here we adopt the wireless signal model in [64]

$$Y_i = Z_i + \frac{A_{ij}(t)}{\sqrt{d_{ij}}} X_i, \tag{5.1}$$

where X_i is the signal transmitted to user i, Y_i is the signal that user i receives, Z_i is the additive Gaussian noise, $A_{ij}(t)$ is the channel fading factor at time t, and d_{ij} is the distance between user i and user j. We assume the channel is slow fading and the fading does not change within a round, hence $A_{ij}(t)$ remains constant within a round. We also assume that the fading is non-directional Rayleigh fading.

We assume that two users, u_1 and u_2 try to cooperate with each other by exchanging chunks. Each user has a buffer of length L, which keeps L_f chunks to be played, and $L - L_f$ chunks that have been played. First u_1 and u_2 exchange information about the availability of each chunk in the other's buffer, and the transmission power P_1 and P_2 that u_1 and u_2 use to transmit the chunks, respectively. To ensure quality of cooperation, intuitively, users will not cooperate with people who use too small power for cooperation. Hence we assume that P_1 and P_2 are larger than the minimum transmission power required P_{min}. The chunk exchange is done on a round by round basis. At the beginning of each round, each user sends requests to the other users, and at the same time keeps downloading from the original server. Each user is allowed to send multiple requests in each round, and he/she can also answer multiple requests. Let τ be the duration of each round. Figure 5.1 shows how two users cooperate with each other: At the beginning of each round, every user sends chunk requests to each other. Then, the supplier either replies with the requested chunks and starts transmission or rejects the request. After a round duration τ, the same request-answering process is repeated.

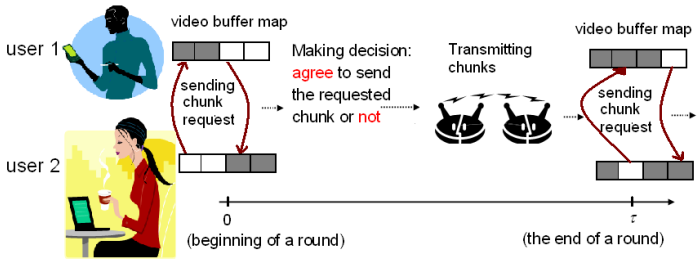


Figure 5.2: Cooperation model for users in the P2P live streaming social network

5.1.2 Two-Player Game Model

To simplify the analysis, we start with modeling the cooperation in each round as a two-person game with single-layer video coding structure. Note that in a mesh-pull live streaming system, although all users watch the same real-time video, the progress at video playback on a peer is determined by how fast the peer collects video chunks from the system. When a new user enters the network, before starting playing the video, he/she waits for a while until he/she has received the first few chunks in the sequence and has buffered enough continuous chunks. Therefore, due to the diverse network conditions and the fact that chunks may arrive out of order, variations in chunk retrieval time result in different *playback time* for different peers and introduce time lags among users. It has been shown [25] that in pplive, one of the most popular IPTV deployments, the maximum time lag among peers fluctuates around 150 seconds within a one hour time period. In this scenario, every chunk has the same value, thus users will always request chunks closest to their playback time. Assume that in the original structure, every user in the wireless live-streaming social network only asks the original server in the Internet for the media chunks, and two of them, u_1 and u_2, want to see if they can cooperate with each other

to get a better-quality video. We model the interactions between u_1 and u_2 using the following game:

- **Players and player types**:There are two players, u_1 and u_2, in this game. Each player u_i has a type $\theta_i \in \{\text{laptop, PDA, PDA2}\}$. Users with different types will have different cost of sharing chunks and gain of obtaining chunks. We assume PDA2 carries weaker battery than PDA, thus the cost per unit energy for PDA2 is higher than PDA.

- **Strategies**: In each round, the two players first exchange their buffer information, and then send the chunk requests to each other. Upon receiving the chunk requests, each player u_i decides how many chunks he/she will send to the other user in this round. We define the number of chunks u_i agrees to send as his/her strategy $a_i \in \mathbb{Z}$. Note that the two users are using the same channel, so the bits to be transmitted within a round can not be larger than the channel capacity, which equals $B \times log(SNR+1)$. Therefore, the constraint of strategy profile (a_1, a_2) at round k is

$$\frac{a_1}{log(1+P_1A_{12}(k)/\sqrt{d_{12}}\sigma_n^2)} + \frac{a_2}{log(1+P_2A_{21}(k)/\sqrt{d_{12}}\sigma_n^2)} \leq \frac{\tau B}{M}. \tag{5.2}$$

  If (5.2) is not satisfied and the users are transmitting chunks above the channel capacity, the probability of transmission would be high and neither will receive any chunks successfully.

- **Utility function**: The utility function π_i of u_i is considered as the gain of receiving chunks (with respect to the opponent's action) minus the cost of sending chunks (his/her own action). Since the members in the wireless live-streaming social network are using mobile devices, the battery energy is the most limited resource.

Hence the cost of cooperation is considered as the transmission energy, and each type of player would give a different weight to the energy cost. For example, clients running on tight energy budget bear a higher cost than those with powerful batteries. Let c_i be the cost per unit energy for u_i, and g_i be u_i's gain of completely receiving one chunk. Every user in the P2P wireless live streaming social network defines his/her own value of g_i depending on how much he/she wants to watch the video. For instance, assume that the NFL final is being broadcasted. An NFL fan would want to try his/her best to receive a high quality video to enjoy the game better, and he/she will set g_i to 1. Another user is watching the game and a movie at the same time. He/she is more interested in the movie, but wants to check the scores/result of the NFL game from time to time. For this user, he/she may give a higher priority to the movie channel, and uses a lower g_i for the streaming of the NFL game.

Based on the above discussion, given the strategy profile (a_1, a_2), the players' payoffs for the k^{th} round are formulated as follows:

$$\begin{aligned}\pi_1(a_1,a_2) &= a_2 g_1 - a_1 c_1 \frac{MP_1}{Blog(1+\frac{P_1 A_{12}(k)}{\sqrt{d_{12}}\sigma_n^2})} \\ \pi_2(a_1,a_2) &= a_1 g_2 - a_2 c_2 \frac{MP_2}{Blog(1+\frac{P_2 A_{21}(k)}{\sqrt{d_{12}}\sigma_n^2})}.\end{aligned} \tag{5.3}$$

Let $\pi(a_1,a_2) = (\pi_1(a_1,a_2), \pi_2(a_1,a_2))$ be the payoff profile. Define $K_1 = MP_1/Blog(1+P_1 A_{\mu_{12}}/\sqrt{a}$ and $K_2 = MP_2 A_{\mu_{21}}(k)/\sqrt{d_{12}}Blog(1+P_2/\sigma_n^2)$. K_i can be considered as the power that user i spends on transmitting a chunk. It is reasonable to assume that $g_i \geq c_i K_i$ and there exists a C_{max} where $c_i K_i \leq C_{max}$. Here c_i and g_i are user i's private information depending on user i's type, and are not known to others. We assume that users do not exchange their private information, i.e., their types. Thus this is a game with incomplete information.

We assume that users have the belief of the probability of the other users' type, which is independent of their own type. Let p_1, p_2, and p_3 be the probability of a user being a laptop, PDA, and PDA2, respectively.

5.2 Optimal Strategies Analysis For Two-Player Game

In this section, we first extend the one-stage game model in Section 5.1.2 into a infinitely repeated game, then apply several optimization criteria such as Pareto optimality and time-sensitive bargaining solution to refine the Bayesian-Nash equilibriums of the game. Furthermore, we discuss the possible cheating behavior which all users may apply to increase their own utility, and design cheat-proof cooperation strategy to stimulate cooperation between two users.

5.2.1 Repeated Game Model

It is easy to show that, if the above game will only be played for one time, the only Bayesian-Nash equilibrium is $(0,0)$, which means no one will answer the other's requests. According to the backward induction principle [55], there will also be no cooperation between the two users when the repeated game will be played for finite times with game termination time known to both players. Therefore, in both circumstances, the only optimal strategy for both players is to always play noncooperatively.

However, in live streaming, these two players will interact many rounds and no one can know exactly when the other player will quit the game. Thus we can model the dynamics between u_1 and u_2 as an infinitely repeated game, and we will show in the

following section that cooperative strategies can be obtained in this realistic model. Let s_i denote player i's behavior strategy, and let $\mathbf{s}_1 = (s_1^{(1)}, s_1^{(2)}, ..., s_1^{(T)})$, $\mathbf{s}_2 = (s_2^{(1)}, s_2^{(2)}, ..., s_2^{(T)})$ denote the strategy profile till the T^{th} round. Next, we consider the following utility function of the infinitely repeated game:

$$U_i(\mathbf{s}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} u_i(s^{(t)}) \tag{5.4}$$

Now, we analyze the Bayesian-Nash equilibriums for the infinitely repeated game with the above utility function U_i. According to the Folk theorem [55], there exists at least one Bayesian-Nash equilibrium to achieve every feasible and enforceable payoff profile, where the set of feasible payoff profiles for the above game is:

$$\begin{aligned} V_0 = \text{convex hull}\{v | \exists\ (a_1, a_2) \text{ with } (\pi_1(a_1, a_2), \pi_2(a_1, a_2)) = (v_1, v_2)\} \\ \text{where } a_1, a_2 \text{ satisfy (5.2)} \end{aligned} \tag{5.5}$$

and the set of enforceable payoff, denoted by V_1, can be easily derived:

$$V_1 = \{(v_1, v_2) | (v_1, v_2) \in V_0 \quad \text{and} \quad v_1, v_2 \geq 0\}. \tag{5.6}$$

Figure 5.3 illustrates both the feasible region and the enforceable region: the feasible region is inside the triangle bounded by dashed lines, and the enforceable feasible set V_1 is the shaded region shown in Figure 5.3. It is clear that there exists an infinite number of Bayesian-Nash equilibriums (BNE). To simplify our equations, in this chapter, we use $\mathbf{x} = (x_1, x_2)$ to denote the set of BNE strategies corresponding to the enforceable payoff profile $(x_2 g_1 - x_1 c_1 K_1, x_1 g_2 - x_2 c_2 K_2)$.

From the above analysis, one can see that the infinitely repeated game has infinite number of equilibriums, and apparently, not all of them are simultaneously acceptable.

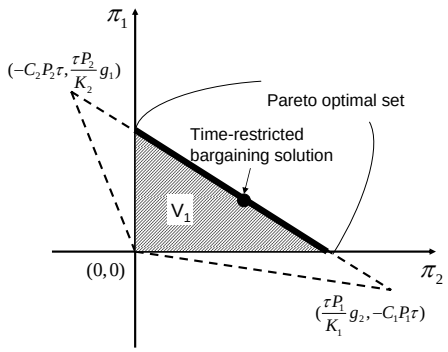


Figure 5.3: Feasible and Enforceable payoff profiles

For example, the payoff profile $(0,0)$ is not acceptable from both players' point of view. Therefore, in this section, we will discuss how to refine the equilibriums based on new optimality criteria to eliminate those less rational and find which equilibrium is cheat-proof.

5.2.2 Nash Equilibrium Refinement

The following optimality criteria will be considered in this section: Pareto optimality, proportional fairness, and absolute fairness.

Pareto Optimality: A payoff profile $v \in V_0$ is Pareto Optimal if and only if there is no $v' \in V_0$ that $v'_i \geq v_i$ for all $i \in N$ [7]. Pareto Optimality means no one can increase his/her payoff without degrading other's, which the rational players will always go to. It is clear from Figure 5.3 that the solid segment between $(-C_2P_2\tau,\, g_1\tau P_2/K_2)$ and $(g_2\tau P_1/K_1, -C_1P_1\tau)$ in the first quadrant is the Pareto Optimal set.

Time-sensitive bargaining solution: Since the players' action pair (a_1, a_2) has to satisfy (5.2), and both players are rational and greedy, they will try to maximize the quality

of their live streaming by asking as many chunks as possible in each round. Every user will request all the chunks that his/her opponent has and that he/she needs. However, according to information theory, the total number of bits being transmitted in within a round has to be less than the channel capacity times chunk duration τ to ensure that the information can be transmitted without bit error. Here we adopt time division multiple access (TDMA) scheme that divide a round time into several time slot, and within a time slot, only one user is occupying the band. Thus users have to bargain for their chunk-request quota for every round to ensure that the total number of bits to be transmitted is not larger than the channel capacity. Also, the gain of receiving a chunk is time-sensitive. For instance, if users cannot reach an agreement on time a user has no gain by receiving that chunk after the playback time.

We model the time-sensitive process for round k as follows: one user offers an action pair $(a_1^{(1)}, a_2^{(1)})$ first, and the other user can decide whether to accept this offer or to reject and offer back another action pair $(a_1^{(2)}, a_2^{(2)})$. This process continues until both players agree on the offer. If users reach agreement at the j^{th} action pair, then g_i decreases to $\delta_i^{j-1}(LC_{k,i})g_i$ for i = 1 or 2, where $\delta_i(LC_{k,i})$ is the discount factor for u_i, $LC_{k,i} = \{I_1, ..., I_q\}$ denotes the indexes of chunks u_i wants to ask in the k^{th} round, and $I(k)$ denotes the index of the chunk playing at the beginning of k^{th} round. Let t be the length of a chunk (in seconds). Suppose the first q' terms in $LC_{k,i}$ are smaller than $I(k) + \tau/t$, which means that among all the chunks that user i needs, there are q' of them have the playback time within the same (k^{th}) round. Therefore, for these q' chunks, if users cannot reach agreement within the k^{th} round, user i gains nothing by receiving them since their playback time has already passed. For the rest $q - q'$ chunks, which would be played after

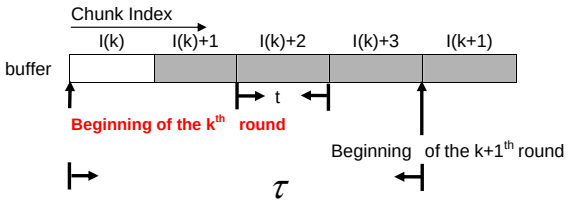


Figure 5.4: Example of a user's buffer with length = 5 chunks

the k^{th} round, user i still receives gain by receiving them still preserve even if bargaining process does not end within a round duration. On the other hand, if one of these $q-q'$ chunks can be received in the k^{th} round, its value is guaranteed to be g_i. However, if the bargaining process in round k takes more time, the number of chunks that can be transmitted in the k^{th} round would decrease. Consequently, a smaller portion of the $q-q'$ chunks can be received in the k^{th} round, thus users receive a small gain. Therefore, even for the chunks which would be played after the k^{th} round, their value would have a higher risk to be dropped if the bargaining time in the k^{th} round is longer.

According to the above analysis, we define the discount factor of gain for user i at round k as follows:

$$\delta_i(LC_{k,i}) = 1 - \frac{\sum_{i=1}^{q'} \frac{\tau}{t} - (I_i - I(k)) + (q-q') * d}{\frac{\tau}{t}(\frac{\tau}{t}+1)/2 + (L_f - \frac{\tau}{t}) * d}, \tag{5.7}$$

where d<1 is the discount constant of the chunks that will be played after the $k+1^{th}$ round begins. For the $q-q'$ chunks that are scheduled to be played after the end of the k^{th} round, it is also better to receive them as soon as possible to prevent their value becomes zero. From such aspect, the value of these $q-q'$ chunks is also decreasing with time and should be counted in δ. However, the value of $q-q'$ does not decrease as fast as the q' chunks that have to be played within this round, and these $q-q'$ chunks should not play

equal roles as the q' chunks that have to be received within this round. So d is the factor to evaluate the less-importance of these $q - q'$ chunks.

For each of the q' chunks whose playback time is within the k^{th} round, the later its playback time, the higher chance that the gain of receiving it can be preserved. We use the chunk index difference to model this phenomena. Thus the first term in the numerator of (5.7) is the sum of the index difference between the requested chunks and the last chunk that can be played in the k^{th} round.

Figure 5.4 gives as an example to illustrate the time-sensitive property for the live-streaming scenario: the white blocks are the chunks that u_1 has in buffer, the grey ones are the chunks he/she needs, and the buffer contains $L = L_f = 5$ chunks. In this example, the number of chunks that u_1 would request, $q = 4$, $q' = 3$, and $\tau/t = 4$. Therefore, $\sum_{i=1}^{q'} \frac{\tau}{t} - (I_i - I(k)) = (4-1) + (4-2) + (4-3) = 6$, and $q - q' = 1$. Let $d = 0.8$, then the discount factor of gain for user i at round k, $\delta_i(LC_{k,i}) = 0.37$.

Since both players' payoffs decrease as the time for bargaining increases, the first mover would seek the equilibrium and offer at the first bargaining round for his/her maximum payoff. Let δ_1 and δ_2 be the averaged discount factor for u_1 and u_2 over all rounds. Note that here we are discussing about the equilibrium of the infinite game, which is the outcome when the game goes to infinity. So at each round, the users do not need to predict δ_i that is averaged over all rounds (including the future). Instead, for each round, the users can calculate the averaged δ_i till the previous round, and find the equilibrium. Such mechanism will result in the equilibrium as follows: The Pareto-optimal equilibrium pair

$((x_1^{(1)},x_2^{(1)}),(x_1^{(2)},x_2^{(2)}))$ for the infinitely repeated game happens when

$$\begin{aligned} x_2^{(2)}g_1 - x_1^{(2)}c_1K_1 &= \delta_1 x_2^{(1)}g_1 - x_1^{(1)}c_1K_1 \\ x_1^{(1)}g_2 - x_2^{(1)}c_2K_2 &= \delta_2 x_1^{(2)}g_2 - x_2^{(2)}c_2K_2, \text{ where } x_1\frac{K_1}{P_1} + x_2\frac{K_2}{P_2} = \tau. \end{aligned} \tag{5.8}$$

Since two users take turn to make the first offer, the time-sensitive bargaining strategy (x_1^*,x_2^*) is

$$\begin{aligned} x_1 &= \frac{1+m}{2} \times \frac{(1-\delta_1)\frac{P_2}{K_2}g_1\tau}{(m-1)K_1c_1 + (m-\delta_1)\frac{K_1P_2}{K_2P_1}g_1} \\ x_2 &= P_2\frac{\tau - x_1\frac{K_1}{P_1}}{K_2}, \text{where } m = \frac{g_2 + c_2K_2\frac{P_2}{P_1}}{\delta_2 g_2 + c_2K_2\frac{P_2}{P_1}}. \end{aligned} \tag{5.9}$$

It is clear that the bargaining solution in (5.9) depends on the knowledge of both users' types, i.e., the private information, which is unavailable. Both players know the discount factors δ_1,δ_2 since the discount factors only depend on the chunks to be requested, which is the information the two users have to exchange. Although at the beginning, users do not know each other's type, they can probe it during the bargaining process using the following mechanism: Let T_1 be u_1's type, which is only known to u_1, let T_2 be u_2's type and $T(j)$ is the j^{th} type. At the first bargaining stage, without loss of generality, let u_1 be the first mover. u_1 calculates all the bargaining equilibriums $(a_1^{(1)}(T_1,T(j)),a_2^{(1)}(T_1,T(j)))$ for $j=1,2,3$ corresponding to the three possible types of u_2. Then u_1 chooses the the equilibrium j' that gives highest $p_{j'}\pi_1(a_1^{(1)}(T_1,T(j')),a_2^{(1)}(T_1,T(j')))$. u_2 will accept the offer if $\pi_2(a_1^{(1)}(T_1,T(j)),a_2^{(1)}(T_1,T(j)))$ is larger than or equal to $\pi_2(a_1^{(1)}(T_1,T_2),a_2^{(1)}(T_1,T_2))$. If not, u_2 will offer back $(a_1^{(2)}(T_1,T_2),a_2^{(2)}(T_1,T_2))$ and reach the agreement. Since u_1 calculates the offer based on the equilibrium in (5.9), which depends on u_1's own type, u_2 can probe u_1's type based on the offer he/she made. Thus

after the first bargaining stage in the first chunk-requesting round, u_2 knows u_1's type, and since u_2 will make the first move in next round, after 2 rounds, the both users have the information of each other's type.

5.2.3 Cheat-Proof Cooperation Strategy

Users in peer-to-peer wireless live streaming social networks would try to maximize their own utility even by cheating. Therefore, to ensure fairness and to give incentives to users, it is crucial that the cooperation strategy is cheat-proof. In this subsection, we will first discuss possible cheating methods, and then propose the two-person cheat-proof cooperation strategy in peer-to-peer wireless live streaming social networks.

5.2.3.1 Cheat On Private Information

Since users know each other's private information (g_i, c_i) by the offers they made, users can cheat by making different offers. First, let us exam whether the time-sensitive bargaining solution in (5.9) is cheat-proof with respect to (g_i, c_i): π_2 increases when x_2 decreases, which can be achieved by increasing x_1 or decreasing P_2.

x_1 is a function of m and

$$\frac{\partial x_1}{\partial m} = -\frac{(1+m)\left(K_1 c_1 + \frac{K_1 P_2}{K_2 P_1} g_1\right)(1-\delta_1)\frac{P_2}{K_2} g_1 \tau}{2\left[(m-1)K_1 c_1 + (m-\delta_1)\frac{K_1 P_2}{K_2 P_1} g_1\right]^2}, \tag{5.10}$$

which is always less than 0 since $m \geq 1 \geq \delta_1$. Thus x_1 is a monotonely decreasing function of m if $\delta_1 < 1$.

Furthermore,

$$\frac{\partial m}{\partial g_2} = \frac{(\delta_2 - 1)c_2 K_2 \frac{P_2}{P_1}}{\left(\delta_2 g_2 + c_2 K_2 \frac{P_2}{P_1}\right)^2} \leq 0 \text{ and } \frac{\partial m}{\partial c_2} = \frac{(1-\delta_2)K_2 \frac{P_2}{P_1}}{\left(\delta_2 g_2 + c_2 K_2 \frac{P_2}{P_1}\right)^2} \geq 0. \tag{5.11}$$

Therefore, m is a monotonely decreasing function of g_2 and is a monotonely increasing function of c_2 if $\delta_2 < 0$. Thus u_2 can have a higher payoff by making the bargain offer using lower g_2, higher c_2, and lower P_2. Similarly, u_1 can also achieve higher utility by offering the equilibrium based on lower g_1, higher c_1, and lower P_1.

As the consequence that both players cheat with respect to c_i and g_i, from the above analysis, both players will bargain based on the minimum value of g_i and maximum value of c_i. Since we have assumed that $g_i \geq c_i K_i$, and $P_i \geq P_{min}$, both players will make the offer based on $g_i = c_i K_i = C_{max}$, and $P_i = P_{min}$, thus the solution (5.9) becomes:

$$\begin{aligned} x_1^* &= \frac{(\delta_2 + 3)(1-\delta_1)}{2(4-(1+\delta_1)(1+\delta_2))} \times \frac{\tau}{M/B\log(1+P_{min}/\sigma_n^2)}, \\ x_2^* &= \frac{\tau}{M/B\log(1+P_{min}/\sigma_n^2)} - x_1^*, \end{aligned} \tag{5.12}$$

which implies that both players should always cooperate with each other. It is clear that solution in (5.12) forms an Nash Equilibrium, is Pareto-Optimal, and is cheat-proof with respect to private information g_i and c_i. Note that the user whose discount factor is closer to 1 has an advantage, and if $\delta_1 = \delta_2$, then $x_1^* = x_2^*$ = half number of chunks can be transmitted in τ seconds.

5.2.3.2 Cheat On Buffer Information

The other way of cheating is to cheat on buffer information, that is, although player i has chunk k in the buffer, he/she does not report it to its opponent. In order to reduce

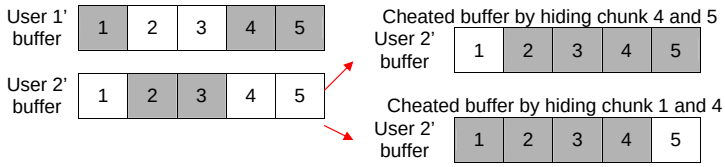


Figure 5.5: An example of how to cheat on buffer information

the number of requests from its opponent. However, hiding the chunk that the other user needs might increase the other user's discount factor based on (5.7).

Take Figure 5.5 as an example. The white blocks are the chunks in buffer, while the grey blocks are the chunks that the user needs. Suppose user 1 always reports his/her buffer information honestly and the time-sensitive bargaining solution gives two chunk-request quota for user 1, and two chunk-request quota for user 2. Apparently, user 1 will ask two of chunk 1, 4, 5 from user 2, and user 2 will ask chunk 2, 3 from user 1. Now if user 2 wants to hide chunks in his/her buffer from user 1, so that the number of chunk requests user 1 will send to user 2 will decrease, and increase user 2's payoff in this round. It is clear that user 2 has to hide at least 2 chunks to increase his/her payoff, since if user 2 only hides one chunk, there are still two chunks in user 2's buffer that user 1 needs. User 2 can choose two of chunk 1, 4, and 5 to hide, and hiding different chunk will lead to different utility. For instance, if user 2 hides chunk 1 and 4, which means chunk 5 is the only chunk that user 1 needs. However, user 2 would ask chunk 2 and 3 from user 1. Since chunk 4 has a later playback time than that of chunk 2 and 3, the discount factor of user 1's gain will be larger than user 2. Thus, user 1 will have more advantage in the time-sensitive bargaining process, and the bargaining solution might be changed to 3

chunk-request quota for user 1 and 1 chunk-request quota for user 2. As a result, user 2's utility decreases because now he/she can only ask one chunk from user 1. Therefore, user 2 has no incentive to cheat on buffer information by hiding chunk 1 and 4.

Although user 2's cheating on buffer information will always increase the the discount factor of user 1's gain (δ_1), it does not necessarily lead to the decrease of chunk-request quota. The reason is the chunk-request quota is always an integer since partial chunk gives no gain for each user and the users would like to round the time-sensitive solution to the closest integers. For instance, if before cheating, the time-sensitive bargaining solution is (1.8, 2.2), and the solution changes to (2.4, 1.6) after cheating. Both solutions round to (2, 2), which means if user 2 hides the chunks properly to keep δ_1 low so that the chunk-request quota does not change after cheating, cheating on buffer information will increase user 2's utility since user 2 can still ask two chunks from user 1, and there is only one chunk in user 2's buffer that user 1 needs.

Therefore, to prevent selfish users gain higher utility by cheating on buffer information, each player should not send chunks more than the other one has sent.

5.2.3.3 Cheat On transmitted power

The power that user 1 and user 2 use for cooperation, P_1 and P_2, are declared in the beginning of the game, and they directly influence the feasible region as in Figure 5.3 and the bargaining solution (5.12). As discussed in Section 5.2.3.1, user i can increase his/her payoff by decreasing P_i, thus both users will declare that they use the minimum power P_{min}. However, if the user declares that he/she transmit the chunks using P_{min} but

the actual power used for transmission is less than P_{min}, he/she can have higher utility by paying less cost for cooperation.

Given the signal model in (5.1), the receiver has to estimate the attenuation term $A_{ij}(k)/\sqrt{d_{ij}}$ before estimating the transmitted power. Suppose user i wants to estimate $A_{ij}(k)/\sqrt{d_{ij}}$. If user j is honest, user i can simply ask user j to transmit a probing signal using P_2 to estimate the attenuation. However, in the fully distributed system, user j might be cheating and transmit the probing signal with power lower than P_2, and the estimated attenuation that user i estimated will be more serious than the real attenuation. To solve this problem, we propose that user i sends the probing signal that user j cannot decode to user j and ask user j to transmit back the received signal, and user i can investigate the attenuation from the replied signal.

If user i send the probing signal X to user j, then the signal Y_j that user j receives is $Z_j + A_{ij}(k)/\sqrt{d_{ij}}X$. Suppose the selfish user j wants to manipulate the signal, he/she can secretly amplify Y_j with a constant $\alpha < 1$ and then send αY_j back to user i. Then the replied signal Y_i that user i receive will be

$$Y_i = Z_i + \alpha Z_j + \alpha \frac{A_{ij}^2(k)}{d_{ij}} X. \tag{5.13}$$

Since user i knows X and the noise power σ_n^2, he/she can easily extract $\alpha \frac{A_{ij}^2(k)}{d_{ij}} X$ from Y_i, divide the energy of the residue with σ_n^2, and get the estimation of $1+\alpha^2$. Given α, the attenuation term $\frac{A_{ij}^2(k)}{d_{ij}}$ can be estimated easily. From the above analysis, such probing procedure is cheat-proof since no matter how user j manipulates the signal, the estimation of the attenuation term is independent of α.

After estimating $\frac{A_{ij}^2(k)}{d_{ij}}$, the transmitted power can be easily to be estimated by calcu-

lating the averaged power of the signal at the receiver's side. Therefore, for user i, he/she can compute the estimated transmitted power $P'_j(k)$ for user j at the k^{th} round by

$$P'_j(k) = \frac{d_{ij}}{A^2_{ij}(k)} \frac{1}{\tau_j} \int_{t=t_k}^{t=t_k+\tau_j} \left[y^2(t) - \sigma_n^2 \right], \tag{5.14}$$

where $y(t)$ is the received signal, t_k is the beginning of user j's transmission in the k^{th} round, and τ_j is the duration of user j's transmission in the k^{th} round.

Thus we design a mechanism to prevent cheating on transmitted power based on $P'_j(k)$ in (5.14):

- For each user i at each round k, he/she estimates the transmitted power of the other user j by (5.14). If $P'_j(k)$ is less than P_{min}, then at the round (the $k+1^{th}$ round), user i transmit the chunks using $P'_j(k)$. If $P'_j(k) \geq P_{min}$, user i uses P_{min} power for cooperation.

- Each user estimates the transmitted power at every round and follow the same decision above.

Using the above mechanism, if user i decides to cheat by transmitting chunks with power $P'_i \leq P_{min}$, then the other user j can estimate P'_i and use P'_i to transmit the chunks for user i in the next round. Therefore, although user i increases his/her payoff in this current round, his/her payoff will be decreased in the next round, thus the actual channel capacity is less than the users' estimation using P_{min}. Therefore, the probability of successfully receiving the request chunks for both users would decrease and lead to no gain since they cannot receive the extra chunks by cooperation. Therefore, neither of the users has the incentive to cheat on transmission power if both follow the above mechanism.

5.2.3.4 Two-Player Cheat-Proof Cooperation Strategy

Based on the above analysis, we can conclude that, in the two-player wireless live-streaming game, in order to maximize each user's own payoff and resist possible cheating behavior, for each player in each round, he/she should always agree to send the requested chunks up to the bargained chunk-requesting quota as in (5.9) and should not send more chunks than his/her opponent has sent to him/her. Also, each user should estimate the transmitted power of the other user in every round, and use the estimated power for transmission if it is less than P_{min}. We refer to the above strategy as *two-player cheat-proof wireless live streaming cooperation strategy*.

5.3 Multiuser P2P Wireless Live Streaming Game

In this section, we first introduce the multi-user game formulation to model the behavior of all users in a peer-to-peer live streaming social network. Then we propose a cheat-proof and attack-resistant cooperation strategy for the infinitely repeated game model, and show that the cooperation strategy is a Pareto optimal and subgame perfect Nash equilibrium. We further discuss the impact of handwash attack to the system and design the strategy to against handwash attack.

First, we will try to extend the two-player cooperation strategy derived from the previous section into the multiple-user scenario.

The two-player cooperation strategy suggests that users should be fully cooperative and refuse to cooperate with a user who behaves uncooperatively before. However, transmission errors are inevitable in fading and noisy wireless channels, and the errors can

cause severe troubles. For the two-player cheat-proof cooperation strategy, there exists a positive probability that one packet and a packet cannot be decoded successfully due to transmission errors and has to be retransmitted. Retransmission may cause delay, and some packets can not arrive within one round. In such scenario, the game will be terminated immediately since the two-person cheat-proof cooperation strategy asks for equal contribution between users, and the performance will be degraded drastically. Therefore, the malicious users can claim it was due to the erroneous Internet traffic and pretend to be non-malicious. Distinguishing misbehavior caused by bit errors and packet loss from that caused by malicious intention is a challenging task.

Also, in the multi-user scenario, the repeated game model is not applicable. For example, a peer may request chunks from different peers at different time slots to maximize his/her utility. A direct consequence of such a non-repeated model is that favors cannot be simultaneously granted. When favors cannot be granted simultaneously, players falls into the dilemma of egoism or altruism, where egoism is an intuitive choice but will stop others from giving favors. Meanwhile, altruism may not guarantee satisfactory future payback, especially when future is unpredictable. Hence the two-player cheat-proof and attack-resistant solution cannot be directly applied to the multi-user scenario.

5.3.1 Multi-user Game Model

Next, we will investigate how to stimulate cooperation for all users in peer-to-peer wireless live streaming over noisy channels, and analyze users' behavior dynamics. We focus on the scenario that video streaming will keep alive for a relatively long time, and there

exist a finite number of users (for example, people watch live Super Bowl over the Internet). Each user will stay in the social network for a reasonably long time (for instance, from the beginning to the end of the game). They are allowed to leave and reconnect to the network when necessary. Each user has an unique user ID registered at the first time he/she joins this network for identification purpose, and he/she uses the same ID whenever he/she reconnects to the same network. We consider an information-pull model, where the streaming server has no duty to guarantee the successful delivery of chunks and it only sends out chunks upon users' demand.

For each user, uploading chunks to other users will incur cost, and successfully receiving chunks can improve the quality of his/her video and thus brings some gain. To simplify the analysis, in this section, we assume that the video stream is encoded using a non-scalable video codec. Therefore, for each user i, each received chunk gives the same gain g_i, whose value is specified by the user individually and independently. As discussed in Section 5.2, g_i, the gain of receiving a chunk for the live video, is evaluated by user i depending on how much he/she wants to watch the video.

In a real-world social network, some users may be malicious, whose goal is to cause damages to other users. In this chapter, we focus on inside attackers, that is, the attackers also have legitimate identities, and their goal is to prevent selfish users from getting chunks. In P2P wireless live streaming social networks, there are three ways to attack the system:

1. **Handwash Attack**: Since peer-to-peer system has a pure anonymous nature that each user is identified by the ID they registered, if a malicious user is detected

and cannot cause damage to the system anymore, he/she can delete his/her ID and register for a new one to come back to the social network. By handwashing, the attacker can keep causing damages to the system as a new comer.

2. **Incomplete chunk attack**: A malicious user agrees to send the entire requested chunk to the peer, but sends only portions of it or no data at all. By doing so, the requesting user wastes his/her request quota in this round, and has to request the same chunk again in the next round.

3. **Pollution attack**: The other kind of attack in peer-to-peer wireless live streaming is pollution [58]. In P2P wireless streaming system, a malicious user corrupts the data chunks, renders the content unusable, and then makes this polluted content available for sharing with other peers. Unable to distinguish polluted chunks from unpolluted files, unsuspecting users download the polluted chunks into their own buffers, from which others may then download the polluted data. In this manner, polluted data chunks spread through the system.

Instead of forcing all users to act fully cooperatively, our goal is to stimulate cooperation among selfish (non-malicious) users as much as possible and minimize the damages caused by malicious users. In general, not all cooperation decisions can be perfectly executed. For example, when a user decides to send another peer the requested chunks, packets of the chunk may not be correctly decoded at the receiver's side. In this chapter, we assume that the requesting peer gives up the chunk if it does not arrive in one round, and we use P_{ij} to denote the probability of successful transmission of a chunk from peer i to peer j in one round of τ second. At the beginning of every round, each user will first

bargain the chunk-request quota, and then send chunk requests to others. We assume that every chunk request can be received immediately and perfectly.

In order to formally analyze cooperation and security in such peer-to-peer live streaming networks, we model the interactions among peers as the following game:

- **Server**: The video is originally stored at the original streaming server with upload bandwidth W_s, and the server will send chunks in a round-robin fashion to its peers. All players are connected via the same access point to the Internet. This backbone connection has download bandwidth W_d.

- **Players and player type**: There are finite number of users in the peer-to-peer wireless live streaming social network, denoted by N. Each player $i \in N$ has a type $\theta_i \in \{\text{selfish}, \text{malicious}\}$. Let N_s denote the set of all selfish players and $N_m = N \backslash N_s$ is the set including all inside attackers. A selfish(non-malicious) user aims to maximize his/her own payoff, and may cheat to others if cheating can help increase his/her payoff. A malicious user wishes to exhaust other peers' resources and attack the system.

- **Chunk requesting**: In each round, users bargain for chunk-request quota based on the time-sensitive bargaining solution since the channel dedicated for user cooperation has limited bandwidth B. For each chunk-request quota, the user can send multiple chunk-request to one user. Users can use their chunk-request quota either *requests chunks from other users* or *does not request any chunks* in this round. On the other hand, since the user-cooperation channel is different from the channel between users and the access point, the users can ask the server for chunks at the same

time.

- **Request answering**: For each player, after receiving a request, it can either *accept* or *refuse* the requests.

- **Cost**: For any player $i \in N$, uploading a chunk to another player incurs cost $c_i MP_i / Blog(1+\frac{P_i}{\sigma_n^2})$, where c_i is the user-defined cost per unit energy, P_i is the transmission power that player i uses for cooperation and $P_i \geq P_{min}$, same as in Section 5.2.

- **Gain**: For each selfish user $i \in N_s$, if he/she requests a data chunk from another peer j, and if a clean copy is successfully delivered to him/her, his/her gain is g_i where $g_i > c_i MP_i / Blog(1+\frac{P_i}{\sigma_n^2})$.

- **Utility function**: We first define the following symbols: for each player $i \in N$,

  - $Cr^{(i)}(j,t)$ is the total number of chunks that i has requested from j by time t. Here, j can be either a peer ($j \in N$) or j is the streaming server. $Cr^{(i)}(t) = \sum_{j \in \{N,\text{ source}\}} Cr^{(i)}(j,t)$ denotes the total number of chunks that i has requested by time t.

  - By time t, peer i has successfully received $Cs^{(i)}(j,t)$ chunks from peer j in time (a chunk is received in time if and only if it is received within the same round that it was requested). $Cs^{(i)}(t) = \sum_{j \in \{N,\text{source}\}} Cs^{(i)}(j,t)$ is peer i's total number of successfully received chunks by time t.

  - By time t, $C_p^{(i)}(j,t)$ is the total number of polluted chunks that peer i received from peer j. The total number of successively received unpolluted data chunks

that peer i received from peer j is $Cs^{(i)}(j,t) - C_p^{(i)}(j,t)$, and each successfully received unpolluted chunk gives peer j a gain of g_i.

– $Cu^{(i)}(j,t)$ denotes the number of chunks that i has uploaded to player j by time t. $Cu^{(i)}(t) = \sum_{j\in\{N,\text{source}\}} Cu^{(i)}(t)$. The cost of uploading each chunk is c_i for peer i.

Let t_f be the lifetime of the peer-to-peer live streaming social network, and $T^{(i)}(t)$ denotes the total time that peer i is in the network by time t. Then, we model the player's utility as follows:

1. For any selfish player $i \in N_s$, its utility $U_s^{(i)}(t_f)$ is defined as

$$U^{(i)}(t_f) = \frac{\left[Cs^{(i)}(t_f) - \sum_{j\in N} C_p^{(i)}(j,t_f)\right] g_i - Cu^{(i)}(t_f)\frac{MP_i}{Blog(1+P_i/\sigma^2)}}{Cr^{(i)}(t_f)}, \quad (5.15)$$

where the numerator denotes the net profit (i.e., the total gain minus the total cost) that the selfish peer i obtained, and the denominator denotes the total number of chunks that i has requested. This utility function represents the average net profit that i can obtain per requested chunk, which i aims to maximize.

2. For any malicious player $j \in N_m$, its objective is to maximize its utility

$$U_m^{(j)} = \frac{\sum_{i\in N_s} Cu^{(i)}(j,t_f)\frac{MP_i}{Blog(1+\frac{P_i}{\sigma^2})} + \sum_{i\in N_s}\left[Cr^{(i)}(j,t_f) - Cs^{(i)}(j,t_f)\right] g_i - Cu^{(j)}(t_f)\frac{MP_j}{Blog(1+\frac{P_j}{\sigma^2})}}{T^{(j)}(t_f)}. \quad (5.16)$$

The numerator in (5.16) represents the net damage caused by j: the first term describes the total costs to other peers when sending the requested chunks to

the malicious user j; the middle term evaluates other selfish peers' potential loss in gain due to the incomplete chunk attack by peer j; and the last term is peer j's cost by uploading chunks to other peers. We normalize it using the lifetime of peer j, $T^{(j)}(t_f)$. Now, this utility function represents the average net damage that j causes to the other nodes per time unit.

5.3.2 Cheat-Proof and Attack-Resistant Cooperation Stimulation Strategies

Based on the system description in Section 5.3.1, we can see that the multiple player game is much more complicated than the two-person game in Section 5.2, and pose new challenges. Thus, direct application of the two-player cooperation strategies to multiple player scenarios may not work.

5.3.2.1 Challenges in Multiple User Scenario

For peer-to-peer live streaming networks in heterogeneous Internet traffic environments, user cooperation stimulation has the following challenges: First, transmission errors are inevitable in the wireless network and the repeated game model is not applicable as discussed int the previous subsection. Second, Malicious users make cooperation stimulation extremely challenging. Misbehavior can result in the decrease of video quality experienced by other peers, which may consequently decrease the quality of service provided by the affected peers. This quality degradation will then be propagated back to the misbehaving peers. Therefore, selfish nodes have no incentives to intentionally behave

maliciously in order to enjoy a high quality video. However, the malicious attackers' goal is to degrade the live streaming network performance, and such quality degradation is exactly what they want to see. Unfortunately, malicious behaviors have been heavily overlooked when designing cooperation stimulation strategies.

5.3.2.2 Malicious User Detection

To distinguish "intentional" malicious behavior from "innocent" misbehavior caused by packet delay, we adopt the credit mechanism and the statistical-based malicious user detection in Chapter 4 and introduce trust modelling to resist handwash attack . In this chapter, we incorporate the trust modelling into the attacker detection mechanism from Chapter 4, and will prove by simulation result that the combined anti-attack mechanism can resist handwash attack.

1. **Credit mechanism for pollution attack**: Addressing the pollution attack, for any two peers $i, j \in N$,

$$Cc^{(i)}(j,t) = Cu^{(i)}(j,t) - C_p^{(j)}(i,t) \tag{5.17}$$

calculates the total number of *unpolluted* chunks that user i has uploaded to user j by round t, where $C_p^{(j)}(i,t)$ is the number of polluted chunks that user i has uploaded to user j.

Since peer i cannot identify a chunk as a polluted one until he/she starts decoding and playing that chunk, it is possible that user i *unintentionally* forwards a polluted chunk to other peers. Thus to distinguish the malicious behavior and the unintentionally pollution by non-malicious users, we adapt the credit-line mechanism as in

Chapter 4 that

$$D^{(i)}(j,t) \leq D^{(i)}_{max}(j,t), \quad \forall t \geq 0, \text{where}$$

$$D^{(i)}(j,t) = Cc^{(i)}(j,t) - Cc^{(j)}(i,t) = \left(Cu^{(i)}(j,t) - C^{(j)}_p(i,t)\right) - \left(Cu^{(j)}(i,t) - C^{(i)}_p(j,t)\right) \quad (5.18)$$

Here, $D^{(i)}_{max}(j,t)$ is the "credit line" that user i sets for user j at time t. The credit line is set for two purposes: 1) to prevent egoism when favors cannot be simultaneously granted and to stimulate cooperation between i and j, and 2) to limit the possible damages that j can cause to i. By letting $D^{(i)}_{max}(j,t) \geq 0$, i agrees to send some extra, but at most $D^{(i)}_{max}(j,t)$ chunks to j without getting instant payback. Meanwhile, unlike acting fully cooperatively, the extra number of chunks that i forwards to j is bounded to limit the possible damages when j plays non-cooperatively or maliciously.

To stimulate cooperation in the first few rounds, $D^{(i)}_{max}(j,t)$ should be large enough in the first few cooperating rounds between user i and j. On the other hand, $D^{(i)}_{max}(j,t)$/[total number of rounds after time t] should be closed to 0 to prevent decreasing the utility of user i. Therefore, when choosing $D^{(i)}_{max}(j,t)$, user i should first estimate the number of remaining rounds for the live streaming, and choose a relatively small number D_{temp}. Then make D_{temp} with the reciprocal of the probability of successful transmitting a chunk from user j to user i to stimulate the cooperation. A simple solution to this is to set the credit lines to be reasonably large positive constants, as in our simulations in Section 5.5.

2. **Statistical-based malicious user detection**:

Since the users have to know the transmission protocol of each other to cooperate,

given the signal to noise ratio in the k^{th} round, $P_j A_{ij}(k)/\sqrt{d_{ij}(k)}\sigma^2$, the probability of user j successfully transmit a chunk to user i without retransmission in the k^{th} round, $P_{ij}(k)$, can be estimated. We assume the users use TDMA to share the wireless channel, so there is only one user occupying the band in one time slot with no interference. Under such scenario, $P_{ij}(k)$ can be calculated by the probability successfully transmitting all symbols in a chunk. First, the symbol error rate $e_s(k)$ of each information block given the modulation type, channel coding scheme, and the signal to noise ratio can be analytically calculated according to [65]. Assume there are b_s bits per symbol. Then the $P_{ij}(k)$ can be estimated as $(1-e_s(k))^{M'/b_s}$. The other way of probing $p_{ij}(k)$ is user i sends probing request to ask user j send the probing package. However, such method is not appropriate in wireless live streaming social network since user j can also intentionally send the incomplete probing package to reduce $P_{ij}(k)$.

After Hence when player i decides to send a chunk to player j in round k, with probability $1-P_{ij}(k)$, this chunk transmission cannot be completed without retransmission because of the fading channel. That is, we use a Bernoulli random process to model the unsuccessful transmission of a chunk due to high traffic internet connection. Let $\overline{P_{ji}(t)}$ equals to the averaged $P_{ji}(k)$ within all the rounds that user j has sent chunks to user i by time t. Given the Bernoulli random processes and $P_{ij}(k)$ being the probability of successfully receive a chunk in round k, then by time t, user i is supposed to receive $P_{ji} \times Cu^{(j)(i,t)}$ chunks from user j, but the actual number is $C_s^{(i)}(j,t)$. Hence if user j does not intentionally deploy the incomplete

chunk attack, based on the Lyapunov's Central Limit Theorem [59], if t goes to infinity, then $C_s^{(i)}(j,t) - \overline{P_{ji}(t)} \times Cu^{(j)(i,t)}$ should follow normal distribution.

for any positive real number x, we can have

$$\lim_{Cu^{(j)}(i,t)\to\infty} Prob\left(\frac{Cs^{(i)}(j,t) - Cu^{(j)}(i,t)\overline{P_{ji}(t)}}{\sqrt{Cu^{(j)}(i,t)\overline{P_{ji}(t)}(1-\overline{P_{ji}(t)})}} \geq -x\right) = \Phi(x), \tag{5.19}$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt$ is the Gauss tail function.

Therefore, based on (5.19), given a predetermined threshold $h > 0$, every selfish peer i can identify peer j as a malicious user by thresholding $Cs^{(i)}(j,t) - Cu^{(j)}(i,t)\overline{P_{ji}(t)}$ as follows:

$$\begin{aligned} j \in N_m^{(i)}(t) \quad \text{iff} \quad Cs^{(i)}(j,t) - Cu^{(j)}(i,t)\overline{P_{ji}(t)} \quad &\leq \quad -h\sqrt{Cu^{(j)}(i,t)\overline{P_{ji}(t)}(1-\overline{P_{ji}(t)})}, \\ \text{and} \quad j \in N_s^{(i)}(t) \quad \text{iff} \quad Cs^{(i)}(j,t) - Cu^{(j)}(i,t)\overline{P_{ji}(t)} \quad &> \quad -h\sqrt{Cu^{(j)}(i,t)\overline{P_{ji}(t)}(1-\overline{P_{ji}(t)})} \end{aligned} \tag{5.20}$$

In (5.20), $N_m^{(i)}(t)$ is the set of peers that are marked as malicious by peer i at time t, and $N_s^{(i)}(t)$ is the set of peers that are marked as selfish by peer i at time t.

3. **Trust Modelling for handwash attack**: In an environment where malicious users might mount the hand-wash attack, selfish users suffer badly from the hand-wash attack, thus the unknown risk of interacting with untrustworthy users will reduce the incentive for cooperation in P2P wireless live streaming social networks. With handwash, malicious users can pretend to be innocent until being detected again. The malicious user detection method above is statistic-based, which means the selfish users have to wait for enough rounds to interact with the malicious user before detection. This statistics collection process allows the handwashed malicious user to cause extra damage to the system. Thus to reduce the influence of handwash

attack, selfish users have to identify malicious users as soon as possible in order to reduce their losses. A straightforward solution is to reduce the credit line $D^{(i)}_{max}(j,t)$ defined in (5.18) or adjust the threshold in (5.20). However, an arbitrary decrease of the credit line or detection threshold will prevent users from cooperation, resulting in the failure of the whole social network. For instance, if user j is not malicious but just polluted by other malicious users, user i will loose the extra gain by cooperating with user j if user i decreases $D^{(i)}_{max}(j,t)$ arbitrarily.

Therefore, to provide a guideline of setting the credit line and calculating the detection statistics for malicious users, we introduce the idea of *trust* among selfish users. If a selfish user choose several trusted users to share the information of interaction with other intrusted users, the malicious user detection can be faster thus decrease the damage by handwash attack. Also, by taking the damage of the intrusted user j caused to other trusted users into credit line $D^{(i)}_{max}(j,t)$ can also stop cooperation with malicious users earlier. It is well known that trust is the driving force for cooperation in social networks [66]. In the following we will discuss how to utilize the trust model to against handwash attack.

A selfish user i establishes direct trust with another user j upon observations on whether the previous interactions between user i and j are successful. We adopt the beta-function-based method in [67], where user i trusts in user j at time t with value $Tr^{(i)}(j,t)$, which is defined as

$$Tr^{(i)}(j,t) = \frac{Cs^{(i)}(j,t) - Cp^{(i)}(j,t) + 1}{Cr^{(i)}(j,t) + 2}. \tag{5.21}$$

If user j is not malicious and also not serious polluted, based on the definition,

$Tr^{(i)}(j,t)$ should be closed to P_{ij}. If user j mounts pollution attack, $Cp^{(i)}(j,t)$ will increase and if he/she mounts incomplete-chunk attack, $Cs^{(i)}(j,t)$ will decrease. Thus both types of attack decrease the numerator in (5.21), resulting in low trust value for malicious users. Also, the trust is directional, which means user i trusts user j does not imply that user j also trusts user i.

Since the trusted selfish users would like to identify the malicious users together, the damage caused by intrusted users to the trusted users are considered collectively. For example, if user i trusts another user j at round t, user i consider the damage that malicious user k has caused to user j as his/her own damage. This scenario is equivalent to reduce the credit line $D^{(i)}(k,t)$ in (5.18) to $D^{(i)}(k,t) - Tr^{(i)}(j,t) \times D^{(j)}(k,t)$. There is an effective bad-mouthing attack against the trust system, where malicious users provide dishonest recommendations to frame up good parties and/or boost trust values of malicious users [66]. To resist such bad-mouthing attack, selfish users should only trust users who have sent them certain number of unpolluted chunks. Assume that selfish user i will only trust user j at time t if user j has sent i more than $Ch^{(i)}(t)$ useful chunks, that is, if $Cs^{(i)}(j,t) > Ch^{(i)}(t)$. The idea for setting $Ch^{(i)}(t)$ is that even the malicious user badmouthes on other selfish users, he/she has to be cooperative and pay enough cost to be trusted, by which the malicious user causes no damage, even contributes, to the system to be trusted. Another advantage of a peer-to-peer cooperation in wireless network is, everyone can listen to the chunk requests and chunk answering of all the users in the network, so the malicious user cannot arbitrarily badmouth the users that he/she has no interaction

with.

In summary, the credit line $D_{max}^{(i)}(j,t)$ in (5.18) is updated in each round as follows:

$$\begin{aligned} D_{max}^{(i)}(j,t+1) &= \max\left\{1, D_{max}^{(i)}(j,t) - \sum_{k\in N_{Tr}^{(i)}(t)} Tr^{(i)}(k,t)\times D^{(k)}(j,t)\right\} \\ \text{where} \quad N_{Tr}^{(i)}(t) &= \left\{k | k\in N_s^{(i)}(t) \quad \text{and} \quad Cs^{(i)}(k,t) > Ch^{(i)}(t)\right\}. \end{aligned} \tag{5.22}$$

And the malicious user detection is done at each round by

$$j \in N_m^{(i)}(t) \text{ iff } Cs'^{(i)}(j,t) - Cu'^{(j)}(i,t)p_{ji} \le -h\sqrt{Cu'^{(j)}(i,t)p_{ji}(1-p_{ji})}, \text{ and}$$

$$j \in N_s^{(i)}(t) \text{ iff } Cs'^{(i)}(j,t) - Cu'^{(j)}(i,t)p_{ji} > -h\sqrt{Cu'^{(j)}(i,t)p_{ji}(1-p_{ji})}, \text{ where}$$

$$Cs'^{(i)}(j,t) = \sum_{k\in N_{Tr}^{(i)}(t)} Cs^{(k)}(j,t),$$

$$Cu'^{(i)}(j,t) = \sum_{k\in N_{Tr}^{(i)}(t)} Cu^{(k)}(j,t), \text{ and } p_{ji} = \frac{1}{\text{size of } N_{Tr}^{(i)}(t)} \sum_{k\in N_{Tr}^{(i)}(t)} (5.23)$$

if $Cu'^{(i)}(j,t)$ is large enough.

As will be demonstrated in Section 5.5, employing the trust model in (5.21) and replacing the modified credit line as in (5.22) will help improve the system's robustness against the handwash attack by malicious users and significantly increase selfish users' utility.

5.3.2.3 Multiuser cheat-proof and attack-resistant cooperation strategy

In summary, the cheat-proof cooperation stimulation strategies in peer-to-peer wireless live streaming social networks are:

Multiuser cheat-proof and attack-resistant cooperation strategy: *In the peer-to-peer wireless live streaming game, for any selfish peer $i \in N_s$, he/she initially marks every other user $j \in N,\ j \neq i$ as selfish. Then, in each round t, i uses the following strategy:*

- *First bargain the chunk-request quota with other users in the network*
- *Update the credit line $D_{max}^{(i)}(j,t)$ by (5.22) and identify malicious users by (5.23)*
- *If i has been requested by j to send chunks, i will accept this request if j has not been marked as malicious by i and (5.18) holds; otherwise, i will reject the request.*
- *When i is requesting a chunk, he/she will send the request to peer j who satisfies*

$$j = \arg \max_{j \in N_s^{(i)}(t), j \neq i} P'_{ji}, \tag{5.24}$$

where $P'_{ji} = P_{ji} \times Cc^{(i)}(j,t)/Cs^{(i)}(j,t)$ is the probability that user i successfully receives an unpolluted chunk from user j

5.3.3 Strategy Analysis

Using the same analysis as in Chapter 4, the above multiuser cheat-proof cooperation strategy can be proven to be a subgame-perfect and Pareto-Optimal Nash equilibrium of the multiuser wireless live streaming game if there exists no attackers. It can also be shown by the proof in [28] that the cooperation strategy is attack-resistant to pollution attack and incomplete chunk attack.

Here we will analyze the optimal attacking strategy with handwash attack.

5.3.3.1 Optimal attacking strategy:

As discussed in [28], the damage that each attacker by pollution attack and incomplete-chunk attack can cause to selfish user i is bounded by $D^{(i)}_{max}$, which is negligible if the P2P wireless network has infinite lifetime. In this scenario, peer i will still waste his/her resource on the hand-washed malicious user j since i does not recognize j's new identity and every user is marked as non-malicious at the beginning. Therefore, with the hand-wash attack, malicious users can increase their payoff dramatically. To simplify the analysis, we assume the attackers will only apply the hand-wash attack at the beginning of each round. For every (selfish or malicious) user in P2P wireless live streaming, at the beginning of each round, besides the strategies discussed in Section 5.3.1, he/she can also choose to hand wash.

Theorem 1. In the P2P wireless live streaming game where every selfish user follows the cheat-proof cooperation strategy proposed in Section 5.3.2.3, if a malicious user i is not detected by any other users and if $D^{(j)}(i,t) < D^{(j)}_{max}(i,t)$ for all other users $j \in N$, hand wash will not provide the malicious user i any further gain. If the malicious user i is detected by another user j, or if there exists another user $j \in N$ where $D^{(j)}(i,t) \geq D^{(j)}_{max}(i,t)$, then the hand-wash attack will increase the malicious attacker i's payoff.

Proof. If the malicious user i is not detected by any other user and (5.18) is satisfied for all $j \in N$, then all the selfish users will still cooperate with the malicious user i. Using the original identity, i receives the same utility as he/she mounts the hand-wash attack and therefore, hand-wash will not bring the malicious user any extra gain. In the scenario where i is detected by a selfish user j as malicious and j refuses to cooperate with i

any longer, if i chooses to hand-wash and reenters the game with a new ID, then j will cooperate with i until (5.18) is not satisfied or i is detected again. Therefore, in this case, i's payoff is increased by causing extra damage to the selfish user j.

From Theorem 1 and [28], *the optimal attacking strategy for a malicious user* is: Upon receiving a request an attacker $j \in Nm$ should always reject the requests; the attackers should always send requests to selfish users, until they do not agree to help, and hand-wash once he/she is identified malicious by one user in the social network. For a malicious use i, to determine whether it has been detected, he/she observes other users' behavior: a selfish user j will always reject the malicious user i's request if and only if i has been identified as malicious by j.

5.4 P2P wireless live video-sharing cooperation strategy

In this section, we consider two more issues for P2P wireless live video-sharing social networks: coding the live stream into different layers and giving extra chunk-request quota to utilize the broadcast nature of wireless channels. In this chapter, we improve the efficiency of cooperation by taking the advantage of the broadcasting nature of the wireless network. Then we present the P2P wireless live video-sharing cooperation strategy.

5.4.1 Multiple Layered Coding

Since different users in the P2P wireless live streaming social network use different devices, their demand of video quality is different. For instance, for devices with smaller screen as PDA or cell phones, the spatial resolution of the video can be lower than lap-

tops but still have the same visual quality. Under this circumstances, spatial video coding, which encode the video into bitstreams with different priorities can provide better quality of service. The base layer provide the most important information while the enhancement layers gradually refine the reconstructed video at the decoder's side. Higher layers cannot be decoded without all the lower layers. Therefore, receiving chunks in different layers gives the user different gains, depending on which video quality the user addresses most.

In addition, suppose that the video is encoded into V_L layers, and based on user i's device, he/she is satisfied with the video with $V(i)$ layers, then user i has no incentives to ask for chunks in layer higher than $V(i)$. The reason is that since chunks in layer higher than $V(i)$ do not increase visual quality for small-screen device, receiving those chunks gives no gain for user i. Therefore, for each user i, upon deciding which chunks to ask in the round, he/she will first determine how many layers he/she needs based on the device. Then he/she requests chunks that give him/her the highest video quality depending on which quality measure user i values most. For the later part of chunk-requesting, we adopt the chunk-request algorithm with tradeoff in Chapter 4.

5.4.2 Over-Request For Broadcast Nature

According to the cooperation strategy in Section 5.3, users will first bargain for chunk-request quota to ensure the total bits to be transmitted in one round does not exceed the channel capacity. On the other hand, the bargained quota also ensures that every user is capable of answering all the requests that he/she receives. Thus based on the above analysis, selfish users have incentives to answer all the requests in every round.

However, since all users in the peer-to-peer wireless live streaming social network share the same wireless cooperation channel, which has the broadcasting nature that allows the users to listen to others' signals, every selfish user will tend to broadcast the requested chunks to all the users that ask the same chunk to reduce the cost of cooperation. As a result, the overall number of bits transmitted in one round will be much less than the channel capacity since some chunk-requests are combined by one transmission. Therefore, we propose the *over-request* mechanism to fully utilize the channel capacity:

- After bargaining for chunk-request quota, allow each user to send up to K times the bargained quota. $K > 1 \in N$ is is a pre-defined constant which is agreed by all the users.

- During chunk-requesting stage, users mark the chunk requests with 1 (for the requests use the bargained quota) or 0 (for the requests use the extra quota).

- Then in the request-answering stage, all the users first choose $q = 1$ chunk to be transmitted, and exchange this information to confirm the total bits to be transmitted do not exceed the channel capacity. Increase q until fully utilizing the channel capacity. If when $q = 1$, the total bits to be transmitted exceed the channel capacity, then all the selfish users answer the chunk requests marked with 1.

Although the over-request mechanism can increase the usage of the cooperation channel, the users might not agree to all the chunk requests that are sent to them. Therefore, an algorithm is needed for choosing which chunk requests to answer during cooperation.

Since the live streaming social network will last till the end of the video and has finite life time, selfish users tend to consider the contributions from other peers when choosing which request to answer. This situation will not only encourage the selfish users to be always cooperative in the finite time model but also reduce the damage of handwash attack. Let $Ch^{(i)}(t)$ be the set of chunk indexes that other users request from user i in round t. The users who request chunks from user i must be not marked as malicious by peer i, and also satisfy (5.18) to make their requested chunks included in $Ch^{(i)}(t)$. We propose the following request-answering algorithm: for every selfish peer i, when he/she receives multiple chunk requests from multiple users and has decided to send q chunks by the above over-request mechanism. Then user i chooses q chunks based on the probability

$$P^{(i)}(I_j,t) = \sum_{R(I_k,t)\in Ch^{(i)}(t)} \frac{\sum_{m\in R(I_k,t)}(Cs^{(i)}(m,t)+\varepsilon)^{\gamma_i}}{\sum_{R(I_k,t)\in Ch^{(i)}(t)}\sum_{m\in R(I_k,t)}(Cs^{(i)}(m,t)+\varepsilon)^{\gamma_i}}, \tag{5.25}$$

where $R(I_k,t)$ is the set of users that request chunk I_k from user i at round t and ε is a small number that gives newcomers who have not sent any chunks to peer i a chance to start cooperation. γ_i is a parameter that controls the sensitivity of user i to other peers' contribution. If $\gamma_i = 0$, every peer sent a request to peer i has the same probability of being answered. On the contrary, if $\gamma_i \to \infty$, the request from user who has send most chunks to peer i will definitely be answered.

5.4.3 P2P wireless live video-sharing Cooperation Strategy with Layered Video Coding and Over-Request

From the above discussion, the **P2P wireless live video-sharing cooperation strategy** is as follows: *for any selfish node $i \in N_s$, he/she initially i marks every other nodes $j \in$*

N, $j \neq i$ as selfish. Then, in round t, i uses the following strategy:

- *Identify malicious users by (5.20) and update $D_{max}^{(i)}(i,t)$ by (5.22).*
- *Bargain with other users and get the chunk-request quota which is K times the time-sensitive bargaining solution*
- *In the chunk-requesting stage, i chooses its own maximum number of video layers $C(i)$ and desired video quality measure, applies the chunk-request algorithm (5.24), and sends chunk requests to the users in $N_s^{(i)}(t)$.*
- *Decide q, the number of chunks to transmit in this round by exchanging information with other users in the social network*
- *In the request-answering stage, i first identifies the selfish users that satisfy (5.18). Then, i chooses the chunks to transmit based on the probability distribution in (5.25), and agrees to send the requested chunks to all the selfish users that ask for the chunks and satisfy (5.18).*

5.5 Simulation Results

5.5.1 Simulation Settings

We use ns2 and C as the simulation platform. In our simulation, we assume the users communicate with the access point using IEEE 802.11 within the diameter of 15 meters, and users build their own wireless network that uses a different band dedicated to co-operation. ns2 is used to simulate the wired network from the live-streaming server to

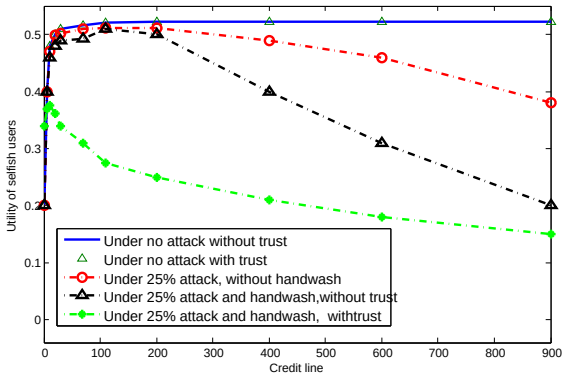


Figure 5.6: Utility of selfish (non-malicious) users under attack versus the initial credit line

the access point, and the communication between the access point and the users. The cooperation among users are simulate by C simulator. ns2 and the C program exchange the real-time simulated results by the log files. The link from the wireless router to the Internet is a DSL link with 1.5Mbits download bandwidth. There are totally 30 users in the network using live-streaming service, and another 5 users using Internet resources at the same time. For the 5 Internet users, we assume the traffic generated from them is a Poission process. The 30 live-streaming users will cooperate by sharing one channel, and we assume every one in the network can connect with any other user in the network via the dedicated cooperation channel. The location of users are randomly distributed within the circle of 15-meter diameter. The users access the channel by TDMA, and users transmit in a round sequence, which is, user 1 transmit first then user 2 and so on, and in the next round, user 2 transmit first and user 1 transmit last.

We fix the ration between the laptop, PDA, and PDA2 users as 3:1:1. The video is initially stored at the original streaming server with an upload bandwidth of 3 Mbps, and

there are other 800 users in the Internet watching the same live stream. The request round is 0.4 second and the buffer length is 10 seconds with $Lf = 20$ and $L = 20$. We choose the "Foreman" and "Akiyo" video sequences with frame rate 30 frames/sec. We encode the video into a 3-layer bitstream with 25 kbps per layer, and divide each layer into chunks of 0.1 second. Thus the layered chunk size is M'= 2.5 kbits. In the wireless network, the chunks are channel coded using BCH code with rate 15/31, thus the chunk size in the wireless live video-sharing social network is M = 5.15 kbits. The 30 live-streaming users in the wireless network can either follow the wireless live streaming cooperation strategy in Section 5.4.3 if they are selfish users, and they follow the optimal attack strategy in Section 5.3.3 if they are malicious attackers. We set $g_i = 1 = C_{max} = 0.8c_{cellphone} * K_i$, $c_{cellphone} : c_{PDA} : c_{laptop}$ = 1:0.9:0.4, $P_{min} = 100mW$, noise power = $20mW$, and bandwidth B = $200kHz$. Discount measure d in (5.7) is set to be 0.7, γ_i in (5.25) is set to be 2 and PDA2 and PDA users are satisfied with only receiving the quality of base layer of the video.

The performance of the cooperation strategies is evaluated by the utilities of the users and the PSNR of the video. The PSNR is calculated by first calculating the mean square error between the original video (Foreman or Akiyo) and the received video, and then divided by the peak pixel value. If a frame is not received or not decodable, it will introduce the square error equals to the sum of all pixel-value square in the frame.

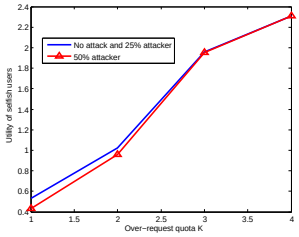


Figure 5.7: Utility of averaged selfish (non-malicious) users with or without attack versus the amount of over-request quota

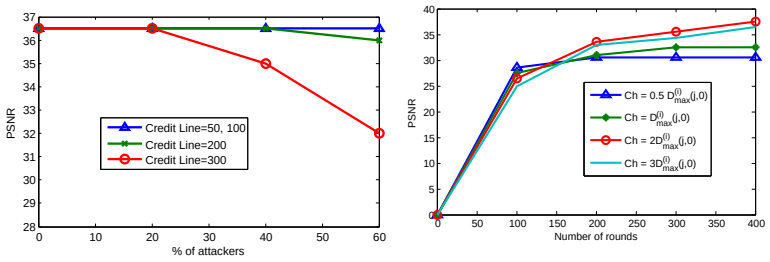
5.5.2 Performance Evaluations

We first study how different initial setting of credit lines can affect cooperation stimulation. Figure 5.6 demonstrates the relationship between credit line and the averaged utility of selfish peers under no attack or 25% of peers being attackers, where the attackers are chosen randomly from all the 20 users. Compare the selfish user's utility when there are no attackers and 25% attackers. In Figure 5.6, the attackers do not mount hand-wash attack, and the trust concept in Section 5.3.2.2 was not incorporated. In both cases, when the credit line is over 50, the selfish nodes' payoff is saturated, and as the credit line keeps increasing, to more than 40s, selfish nodes' utility starts to decrease very fast under attacks, while the utility keeps the same if there are no attackers presented. It is clear that the maximum damage attackers can cause is linearly proportional to the credit line, while the total number of rounds is 9000. When credit line is larger than 400, the damages are no longer negligible. Also, from this plot, it shows that a credit line of 50 is an optimal choice and an arbitrary large credit line will only lower selfish users' utilities is there are attackers presented.

If the attackers mount the hand-wash attack, and the selfish users do not trust each

other, the selfish users' utility will be very small no matter which credit line they choose. This case is shown as black circle dashed line in Figure 5.6. However, the star line in Figure 5.6 shows the selfish users' utility if they trust each other, which is much better than without trust. Here we set the minimum number of successfully received chunks $Ch(i)$ from the trusted users as two times the initial credit line. An intuitive explanation of choosing $Ch(i)$ is that since the initial setting of credit line $D^{(i)}_{max}(j,0)$ can be considered as user i's tolerance of the damage that others cause to him/her. On the other hand, $D^{(i)}_{max}(j,0)$ is the number of chunks that user i thinks an usual non-malicious user should interact with him/her. Thus users who have sent more than two times $D(i)_{max}(j,0)$ chunks successfully to him/her should be trusted. And if the credit line is chosen carefully between 50 and 200, the highest utility can be achieved even the attackers mount the hand-wash attack. The performance of the cooperation strategy with trust when there are no attackers is also presented as red triangle in Figure 5.6, showing that trust concept will not degrade selfish users' utility if every one is non malicious.

Figure 5.7 illustrates the averaged selfish users' utility of the over-request algorithm with or without attack. Here we choose the initial credit line as 50 from the observation drawn in Figure 5.6, and set $Ch(i)$ as 100. When there are 50% of attackers and the users do not over request as in Section 5.4.2, then the utility for selfish users will drop 20% when there are 50% attackers. However, if the users over request to 3 times of the bargained quota, then the utility of the selfish users when there are 50% and 25% attackers will be the same. Thus it is clear that the over-request algorithm can effectively increase the selfish users utility, and the contribution-based chunk-answering algorithm can also help against attack to 50% malicious users.



(a) versus percentage of attackers (b) versus number of rounds

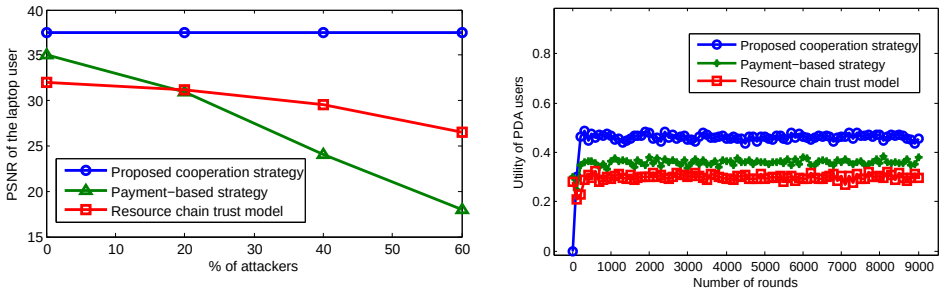
Figure 5.8: PSNR of the selfish laptop users

Figure 5.8 shows the averaged PSNR of the selfish laptop users under different parameter setting. Here the attackers will mount hand-wash attack and the selfish users apply the cooperation strategy as in Section 5.4.3. The PSNR is calculated by the received video given the maximal number of layers of different users. For instance, if the user's device is PDA, then the PSNR is calculated using 2-layer video only. Figure 5.8(a) shows the robustness of different credit line setting versus the percentage of attackers. When the percentage of attackers increases, higher credit line setting will give lower PSNR for the selfish users since the credit line mechanism only ensures the maximal damage of each attacker, and the total damage caused by the attackers can increase if there are more malicious users in the system. Thus, this phenomenon again suggests the credit line should be set as the minimal number that can stimulate cooperation, which is 50 in this case. Figure 5.8(b) shows the selfish user's averaged PSNR under different trust thresholds Ch in (5.22) versus the number of rounds. It is clear after 400 rounds that the selfish user's PSNR is saturated and $Ch = 0.5D_{max}^{(i)}(j,0)$ or $Ch = D_{max}^{(i)}(j,0)$ gives lower PSNR than $Ch = 2D_{max}^{(i)}(j,0)$. These results imply that setting trust threshold Ch too small will cause damage to the system since the selfish users might trust the malicious users

also. On the other hand, from Figure 5.8(b), higher *Ch* needs more number of rounds to saturate the selfish user's PSNR, which means the selfish users need to wait more rounds to trust other users.

Furthermore, we compare our cooperation strategy with the payment-based incentive schemes [26] and the resource chain trust model for P2P security [68]. The credit line is set to 100, and the users over request the chunk by 3 times. We first compare the attack-resistance of the three algorithm as shown in Figure 5.9(a). It is clear that our cooperation strategy is attack-resistant when the percentage of attackers is less than 60%, and the resource chain trust model can resist up to 30% of attackers. The payment-based method is not resistant to the attack, while under no attack the payment-based method can achieve 35 dB but still lower than the proposed cooperation strategy since the payment-based method does not consider the issues of wireless channels.

We also compare the utility for the PDA user versus number of rounds for the three algorithms without attack in Figure 5.9(b). First, the proposed algorithm converge to steady payoff as quick as the payment-based method, while the resource chain trust modelling takes longer time. On the other hand, our proposed scheme gives the PDA users higher utility by taking into account the desired resolution of the user. The PDA user will not request higher-layer chunks and thus he/she will dedicate his/her chunk-request quota to the base-layer chunks and get higher utility.



(a) PSNR of laptop users versus percentage of attackers

(b) Utility of PDA users versus number of rounds

Figure 5.9: Performance comparison of the proposed cheat-proof and attack-resistant cooperation strategies and the payment-based cooperation strategy and the resource chain trust model

5.6 Chapter Summary

In this chapter, we investigate cooperation stimulation in wireless live-streaming social networks under a game theoretic framework. An illustrating two-player Bayesian game is studied, and different optimality criteria, including Pareto-Optimal and time-sensitive bargaining solution is performed to refine the obtained equilibriums. Finally, a cheat-proof cooperation strategy is derived which provides the users in wireless live streaming social network an secured inventive to cooperate.

The results are then extended to stimulate multiuser live streaming, and combing with the chunk-request and request-answering algorithm, a fully-distributed attack-resistant and cheat-proof cooperation stimulation strategy has been devised for peer-to-peer wireless live streaming social networks. Simulation results have illustrated that the proposed strategies can effectively stimulate cooperation among selfish peers in a wireless network, and the incentive-based cooperation strategies are attack-resistant to pollution

attack and handwash attack when the percentage of attackers is less than 25%.

Chapter 6

Optimal Price Setting for Mobile Live Video Service

With the explosive advance of communication technologies and multimedia signal processing, nowadays users can watch live tv program over mobile phones legally by subscribing to the data plans. A recent study [69] shows that there are 97 cellphone users per 100 inhabitants at the end of year 2007 in developed countries. Since almost every person has at least one cellphone, and the phone-to-phone communication is available, some users who have subscribed to the live tv program can try to pirate the live video and sell to the non-subscribers by lower price. Due to the high mobility, high time-sensitiveness, and small transmission range of the mobile devices, each pirate action only exists for a short time. Thus, such pirating market is very difficult to track. Consequently, the better way to prevent the pirating action and protect the copyright of the content owner is to set a price that no subscribers will have incentives to pirate the live video.

The pirates and the non-subscribers who are interested in the live video interact with each other and form a live-video marketing social network. Users influence each others' decisions and performance and both groups of users will reach agreement at the equilibrium price that all users have no incentive to deviate. Hence, such equilibrium

price will serve as the upper bound for the price set by the original service provider. Due to the small coverage of each mobile device, a pirate can only sell the content to the non-subscribers within his/her transmission range. On the other hand, the non-subscribers can only buy the content from the close pirate. If there are multiple non-subscribers within a pirate's transmission range, they have to bid for the live video since one pirate can only transmit to one non-subscriber at a time to avoid interference. If there is only one non-subscriber that the pirate can sell to, then the dynamics between them is a traditional seller-buyer game. To solve this hybrid user dynamics in the live-video marketing social network, we propose a Auction-Stackelburg game to solve the problem.

The rest of the chapter is organized as follows. We introduce the system model and define the problem and the utility functions for the pirate and the non-subscriber in Section 6.1. We then analyze the optimal strategies for all users and provide the solutions in Section 6.2. Simulation results are shown in Section 6.3 and conclusions are drawn in Section 6.4.

6.1 System Model and Problem Formulations

In this section, we first introduce the channel, transmission, and rate-distortion model for the video transmission. Then, we formulate the optimization problem of pirate and power selection using an Auction-Stackelberg game.

6.1.1 System Model

The system diagram is shown in Figure 6.1. Suppose the i^{th} pirate S_i is transmitting the video chunks to the j^{th} non-subscriber B_j using power P_i, the channel between them is slow fading channel with channel gain G_{ij}, the distance between them is d_{ij} and the variance of the additive white gaussian noise at the receiver side is σ_{ij}^2, then the signal-to-noise ratio (SNR) between S_i and B_j can be expressed by

$$SNR_{ij} = \frac{P_i G_{ij}}{\sqrt{d_{ij}}\sigma_{ij}^2}, \tag{6.1}$$

thus bit rate of the video stream, which is the the channel capacity is

$$R_{ij} = W \log_2 \left(1 + \frac{SNR_{ij}}{\gamma}\right), \tag{6.2}$$

where W is the bandwidth of the for transmission, and γ is the capacity gap.

For a video streaming service, a common objective quality measure is the video's peak signal-to-noise-ratio (PSNR). The PSNR of the video stream between S_i and B_j is

$$PSNR_{ij} = 10 \log_{10} \frac{255^2}{MSE_{ij}}, \tag{6.3}$$

where MSE_{ij} is the mean square error which is the distortion of the video. Without loss of generality, in this paper, we use a simple two-parameter distortion-rate model, which is widely employed in a medium or high bit-rate situation, and other models can be similarly analyzed. Since the video bit rate is formulated in (6.2), MSE_{ij} can be expressed by

$$MSE_{ij} = \alpha e^{-\beta W \log_2\left(1+\frac{SNR_{ij}}{\gamma}\right)}, \tag{6.4}$$

where α and β are two positive parameters determined by the characteristics of the video content. If the pirates available to sell the live video to the non-subscriber B_j at a certain

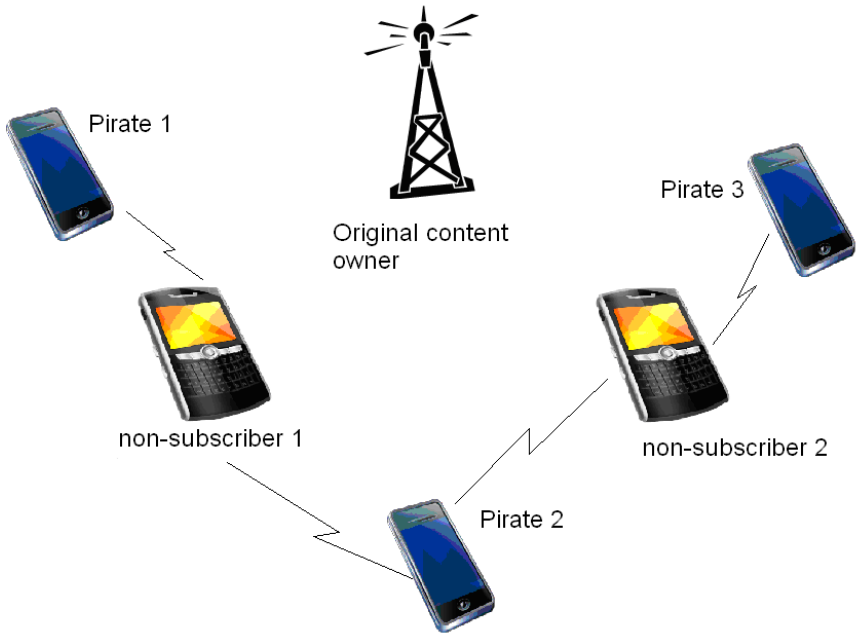


Figure 6.1: An example of a live-streaming marketing social network

time constitute a set, denoted by $L = S_1, S_2, ..., S_N$, then the distortion of the video MSE_j is

$$MSE_j = \alpha e^{-\beta \frac{W}{N'+1} \Sigma_{S_i \in L} \log_2\left(1+\frac{SNR_{ij}}{\gamma}\right)}, \tag{6.5}$$

where N' is the number of pirates that the non-subscriber buy the video stream from.

6.1.2 Problem Formulation

Since the live-video marketing social network is with high mobility, it is very difficult to control the user behavior by a centralized authority. Since each user can only communicate to the users within a certain distance, local information is enough for the users to make decision. Hence, we propose a fully distributed Auction-Stackelburg-game-theoretical model to analyze how the non-subscribers provide incentives for the pirates to

sell the video content, and what is the optimal price and quantity that the non-subscribers should offer. The goal of this analysis is to help the original content owner set the price such that the equilibrium of the game between pirates and the non-subscribers leads to negative payoff, which means the users have no incentive to pirate the video. We start the analysis by the defining the stages of the game and the utility functions of both types of users in the social network.

- **Game Stages** Before the game starts, the pirate and the non-subscribers will declare their existence to let all user within their transmission range know their coverage areas.

  The first stage for the game is the pirates' (auctioneers'/sellers') move. For each pirate who has only one non-subscriber within their coverage areas, the pirate will set the unit price p_i' for his/her transmission power as well as the maximal power that he/she can use for transmission. For every pirate who has more than one non-subscribers within the transmission diameter, he/she will declare the reserve price p_i' per unit power as well as the maximal power. Since there are multiple non-subscribers, they have to bid for the video, and the reserve price is the minimal price for bidding.

  Then in the second stage of the game, the non-subscribers (bidders/buyers) will decide whom to buy the video from and how much power they want the pirate to transmit. The subscriber will offer each pirate the price per unit power p_i and the quantity of power P_i. If a pirate receives more than one offer, he/she will choose the one that maximize his/her utility.

- **Actions of non-subscribers/bidder/buyer** We first discuss the utility function and the optimal action for the non-subscribers. A non-subscriber B_j gain the reward by successfully receiving the live video with a certain quality. On the other hand, B_i has to pay for the power that the pirates use for transmission. Therefore, given the rate-distortion model, the utility function of a non-subscriber B_i can be defined as

$$\pi_{B_j} = a \times (f(PSNR_j) - f(PSNR_{max})) - \sum_{S_i \in L} p_{ij} P_i + p_o, \tag{6.6}$$

where $f(PSNR_j)$ is the reward, $PSNR_{max}$ is the maximal PSNR of the video which can be obtained by buying the video from the original content owner, p_o is the price set by the content owner, P_i is the power that pirate i used for transmission, p_{ij} is the price per unity power paid from non-subscriber j to pirate i, and a is a parameter controlling the balance between the gain and cost. According to the human visual system (HVS) model, the quality difference in the low PSNR region is easier to be distinguished than that in the high PSNR region. Therefore, we define the $f(.)$ function as:

$$f(PSNR) = \ln(PSNR). \tag{6.7}$$

According to the rate distortion model in (6.4) and PSNR formulation in (6.3), the reward term can be formulated as a function of the video bit rate as

$$f(R_{1i}, R_{2i}, ..., R_{Ni}) = \ln(\eta + \beta \frac{W}{N'+1} \sum_{S_i \in L} \log_2 \left(1 + \frac{SNR_{ij}}{\gamma}\right)) \tag{6.8}$$

where $\eta = 2\ln 255 - \ln \alpha$.

Combining (6.1) and (6.6) with the above equation, we can formulate the utility function of B_i as a function of $\{P_i, p_i \; \forall i \in L\}$. Note that p_i is the price that B_i paid

for S_j, not the price p_i' that pirate i asks for. Hence, $p_i \geq p_i'$ and the optimal action for the non-subscriber is

$$\max_{P_i, p_{ij}} a \times \ln(\eta + \beta \frac{W}{N'+1} \sum_{S_i \in L} \log_2 \left(1 + \frac{P_i G_{ij}}{\sqrt{(}d_{ij})\sigma^2 \gamma}\right)) - (\ln(\eta + \beta R_{max})) - (\sum_{S_i \in L} p_i P_{ij} - p_o),$$
$$s.t. p_{ij} \geq p_i', \frac{W}{N'+1} \sum_{S_i \in L} \log_2 \left(1 + \frac{SNR_{ij}}{\gamma}\right) \leq R_{max}, S_i \in L. \quad (6.9)$$

The choice of the optimal auction price p_{ij} and the power quantity P_i not only influenced by the channel conditions between S_i and B_j, but also depends on how many non-subscribers are auctioning for the same pirate and how many pirates that the non-subscriber B_j can buy the video from. For instance, if B_j is the only non-subscriber within the transmission range of S_i, B_j has no incentive to offer $p_{ij} > p_i'$. On the other hand, if B_j has to increase the offer p_{ij} in order to compete with other non-subscribers, B_j might buy more power from other pirates instead.

- **Actions of pirates/auctioneers/seller**

Each pirate S_j can be seen as a seller or auctioneer and aims to not only earn the payment that covers his/her transmission cost but also gain as many extra rewards as possible. We introduce a parameter c_i, the cost of power for relaying data, in our formulation. c_i is determined by the characteristics of the device that pirate S_i uses. Hence, the utility of S_i can be defined as

$$\pi_{S_i} = \max_j (p_{ij} - c_i) P_i, \quad (6.10)$$

where P_i is the power that pirate i used for transmission and $p_{ij}l \leq p_i' \forall j$. Thus, the pirate i will choose the reverse price p_i' such that

$$\max_{p_{ij} \leq p_i'} \pi_{S_i} = (p_{ij} - c_i) P_i \quad \forall i. \quad (6.11)$$

The choice of the optimal reserve price p_i' is affected not only by each pirate's own channel conditions to each non-subscriber but also by the other pirates' prices. This is because the seller-level game is noncooperative, and the relay nodes compete to get selected by source node s. If a certain pirate S_i asks such a high price that makes it less beneficial than the other pirate to the non-subscriber, then non-subscriber will buy less from pirate S_j or even discard it. It is worth noticing that the only signaling required to exchange between the source node and the relay nodes are the price pi and the information about how much power P_i to buy. Consequently, the proposed two-level gametheoretical approach can be implemented in a distributed way. The outcome of the proposed games will be shown in detail in the following section.

6.2 Equilibrium Analysis

First, we obtain closed-form solutions to the outcomes of the proposed games. Then, we prove that these solutions are the global optima.

6.2.1 Analysis of the non-subscribers' actions

We analyze the game from the backward manner by analyzing the optimal strategy for the non-subscribers first. The goal of each non-subscriber B_j is to find the optimal bidding price p_{ij} to bid for the video from pirates who can offer the live video service for multiple non-subscribers and also determine the optimal bit rate that B_j should buy from each pirate in L. Let L_c be the set of pirates that have multiple non-subscribers within their coverage areas. We will answer these two questions by first investigating the maximal

utility $\pi^*_{B_j}(L \setminus L_c)$ by excluding all pirates in L_c and then compare with the maximal utility $\pi^*_{B_j}(L \setminus L_c + S_i)$ with the pirate set $L \setminus L_c + S_i$, in which $S_i \in L_c$ to find the distribution of the bidding function. Then based on the bidding function, we can solve the optimal bid for the pirates in L_c.

- **Solving $\pi^*_{B_j}(L \setminus L_c)$**: For the pirates in $L \setminus L_c$, we can solve the optimal power P_i by taking the derivatives of π_{B_j} in (6.6) with respect to P_i:

$$\frac{\partial \pi_{B_j}}{\partial P_i} = a \frac{\partial \ln(\eta + \beta W \sum_{S_i \in L} \log_2 \left(1 + \frac{SNR_{ij}}{\gamma}\right))}{\partial P_i} - p_{ij} = 0 \; \forall S_i \in L \setminus L_c \tag{6.12}$$

Let $C = BW / \ln 2$, and $A_i = \sqrt{d_{ij}} \sigma^2 \gamma / G_{ij}$ then

$$\frac{C}{(P_i + A - i)\eta + \beta W \sum_{S_i \in L \setminus L_c} \log_2 \left(1 + \frac{P_i G_{ij}}{\sqrt{(d_{ij})} \sigma^2 \gamma}\right)} = p_{ij}. \tag{6.13}$$

Since for any $S_n \in L \setminus L_c$, $n \neq i$,

$$\frac{p_{ij}}{p_{nj}} = \frac{P_n + A_n}{P_i + A_i}, \tag{6.14}$$

$$P_n = \frac{p_{ij}(P_i + A_i) - A_i p_{ij}}{p_{nj}}. \tag{6.15}$$

By substituting the above equation into the denominator of (6.13), we have

$$\sum_{S_n \in L \setminus L_c} \log_2 \left(1 + \frac{P_n G_{nj}}{\sqrt{(d_{nj})} \sigma^2 \gamma}\right) = \sum_{S_n \in L \setminus L_c} \log_2 \left(1 + \frac{p_{ij}(P_i + A_i) - A_n p_{nj}}{A_i p_{nj}}\right), \tag{6.16}$$

thus the optimizer

$$P_i^* = \frac{\eta + \beta W \sum_{S_n \in L \setminus L_c} \log_2 \left(1 + \frac{p_{ij}(P_i^* + A_i) - A_n p_{nj}}{A_i p_{nj}}\right)}{p_{ij}} - A_i. \tag{6.17}$$

The above equation can be solved by numerical method and the unique solution is denoted as $\mathbf{P}_i^*$.

- **Private valuation of the non-subscribers**: Private valuation is the value of the resource that each bidder evaluates by himself/herslf. In the live-video marketing social network, the resource that the non-subscribers are competing for is the live video stored in a pirate S_i. Hence, the valuation of such video is a function of the biding price p_{ij} and can be defined as

$$v(p_{ij}) = \pi^*_{B_j}(L \setminus L_c + S_i, p_{ij}) - \pi^*_{B_j}(L \setminus L_c), \tag{6.18}$$

where $p_{ij} \geq p'_i$, and p'_i is the reserve price of the pirate S_i. Since $\{L \setminus L_c\} \subset \{L \setminus L_c + S_i\}$, $v(p_{ij})$ is always positive for any $p_{ij} \geq p'_i$. On the other hand, $v(p_{ij})$ is upper bounded by the optimal price p^*_{ij} which is the maximizer of $\pi^*_{B_j}(L \setminus L_c + S_i, p_{ij})$ and can be sound by solving

$$\frac{\partial \partial \pi_{B_j}}{\partial p_{ij} \partial P_i} \mid_{p_{ij}=p^*_{ij}} = 0. \tag{6.19}$$

Hence, $v(p_{ij})$ takes value between 0 and $v^*(p_{ij})$ and is a function of p_{ij}. Since p_{ij} can take any value greater or equal to p_i, and the bidder B_j only has local information which is the number of competitors but does not have the information about the channel condition between his/her competitor and the pirate S_i, the best choice of B_j is to uniformly randomize the bidding price p_{ij}, such that p_{ij} takes value in $[p_j, p^*_{ij}]$. As a result, the private valuation $v(p_{ij})$ is also a random variable. Since the bidding rule is that the highest bid wins, the optimal bid will be p^b_{ij} that satisfies

$$\frac{\partial v(p^b_{ij})}{\partial p_{ij}} F(p^b_{ij}) = v(p^b_{ij}) \times f(p^b_{ij}), \tag{6.20}$$

where $f(p^b_{ij})$ is the probability distribution function of p_{ij}, and $F(p^b_{ij})$ is the cumulative distribution function.

Hence, the bidder B_j will bid the price p_{ij}^b and use $p_{ij} = p_{ij}^b$ to calculate the optimal $\mathbf{P}^*{}_i$ in (6.17).

6.2.2 Analysis of the pirates' actions

Given the solution $\mathbf{P}_i^*$ of (6.13), each pirate $S_j \in L \setminus L_c$ seek to maximize their utility by setting the optimal price p_i' that

$$max_{\{p_i'\}} \pi_{S_i}(p_i' - c_i)P_i^*. \tag{6.21}$$

The optimal price $(p')_i^*(\mathbf{G}_j, \mathbf{d}_j)$ should satisfy

$$\frac{\partial \pi_{S_i}}{\partial p_i'} = P_i^* + (p_i' - c_i)\frac{\partial P_i^*}{\partial p_i'}. \tag{6.22}$$

6.2.3 Equilibrium Analysis

Now we will prove that (6.22), (6.17), and form the equilibrium of the Auction-Stackelburg game.

First, $\mathbf{P}_i$ and $\mathbf{p}_j = \{p_{ij} \forall j \in L\}$ form an Auction-Stackelburg equilibrium if

1. For every pirate $S_i \in L_c$, $(p_{ij} \forall j)$ forms an auction equilibrium.
2. When $\mathbf{p}_j$ is fixed, for every non-subscriber B_j, $\pi_{B_j}(\mathbf{P}_i) = \sup_{p_{ij} \geq 0} \pi_{B_j}$.
3. For every pirate $S_i \in L \setminus L_c$ when $\mathbf{P}_i$ is fixed, $\pi_{S_i}(\mathbf{p}_j) = \sup_{p_{ij} \geq 0} \pi_{S_i}$

Then we will discuss the existence of the equilibrium which are the solutions in (6.22) and (6.17). Let $\underline{v}$ be the infimum of the private valuation v and $\bar{v}$ be the supremum of v.

Property 1: $f(v) > 0$ on $(\underline{v}, \overline{v}]$

Proof: According the the definition of $v(p_{ij})$ in (6.18) and given p_{ij} is uniformly distributed in which $f(p_{ij}) > 0 \forall p_{ij} that v(p_{ij}) \in [\underline{v}, \overline{v}]$. Therefore, $f(v(p_{ij})) > 0$ on $(\underline{v}, \overline{v}]$.

Property 2: The payoff of every bidder is less than 0 if the bidding value is less than the reserve value of the auctioneer.

Proof: The payoff of bidder B_j by joining the bid is $\pi_{B_j}(L \setminus L_c + S_i, \mathbf{P}_i) - \pi_{B_j}(L \setminus L_c, \mathbf{P}_i)$ with optimal power vectors $\mathbf{P}_i^{(j)}$ and $\mathbf{P}_i$, respectively. If the bidder offers a bid $p_{ij} < p_i'$ where p_i' is the reserve price of the auctioneer S_i, then S_i will reject the offer and $P_i^{(j)} = 0$ which is the power that S_i sells to B_j. Let $\mathbf{P}_i^{(j)}$ be optimal the power vector of $\pi^*_{B_j}(L \setminus L_c + S_i, p_{ij})$, and $\mathbf{P}_i$ be the optimal power vector of$P_i^{(j)} = 0$, $\pi_{B_j}(L \setminus L_c, \mathbf{P}_i^{(j)})$ $= \pi_{B_j}(L \setminus L_c + S_i, \mathbf{P}_i) < \pi_{B_j}(L \setminus L_c, \mathbf{P}_i)$ since $\mathbf{P}_i$ is the optimizer.

According to [70], any auction satisfies property 1 and property 2 has a unique equilibrium which is composed of the optimal auction of each bidder. Hence, the solution in (6.20) exists and satisfy the definition of the Auction-Stackelburg equilibrium.

6.3 Simulation Results

In this section, we will show the equilibrium of the Auction-Stackelburg game under different scenario as well as the optimal price for the content owner.

6.3.1 Single non-subscriber with multiple pirates

We first set up multiple-pirate single non-subscriber simulations to test the proposed game. We set the coordinate of the non-subscriber as (0 m, 0m), and the pirates are

uniformly located within the range of [50 m, -50 m] in both x-axis and y-axis. The maximal transmit power P_{max} is 100 mW, the noise level is 10^{-8} W, and we select the capacity gap γ =1, bandwidth W = 1 MHz, the gain per $\ln PSNR$ a=0.01 , and the cost per unit of power for each pirate is $c_i = 0.1$. We use the video sequence "Akiyo" in QCIF format and H.264 vodeio codec. The resulted rate-distortion parameter β = 0.0416, and α = 6.8449. We set the maximal PSNR which is provided by the original content owner be 40dB, and the corresponding maximal bit-rate for Akiyo is 84 kB/sec. The subscription price P_o for the video sequence is set to be 0.

In Fig. 6.2, we can observe that as the total number of the available pirates increases, the competitions among the pirates become more severe, so the average payment per pirate decreases. Although the total payment for the non-subscriber increases slowly with the number of pirates, the utility keeps increasing since the video quality is better. This is the nature of free market with more sellers. The optimal price for the content owner, which equals to the negative value of the non-subscriber's utility is also decreasing with the number of pirates.

6.3.2 Multiple non-subscriber with multiple pirates

We then set up multiple-non-subscriber with multiple-pirates simulations to test the proposed game. We will discuss two factors that influence the optimal price P_o. One is the number of non-subscribers in the network, the other is the distance between the non-subscribers.

For the first simulation, we let both pirates and non-subscribers be uniformly lo-

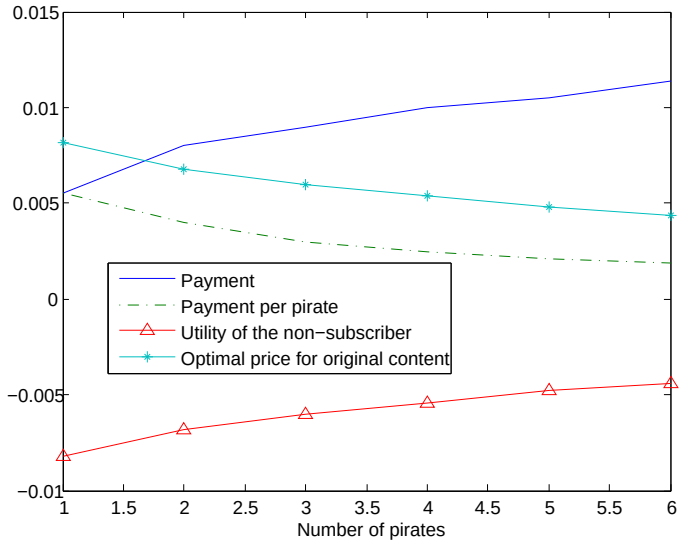


Figure 6.2: Single non-subscriber case with different number of pirates

cated within the range of [40 m, -40 m] in both x-axis and y-axis. The number of pirates is fixed to be 10. Other settings are the same as the previous section, and the subscription price P_o for the video sequence is set to be 0. Fig. 6.3 shows the average payment each non-subscriber pays, the averaged utility of the non-subscribers and the optimal subscription price P_o. We can observe that as the total number of the non-subscriber increases, the higher the number of pirates that can sell the video to multiple non-subscribers. So the competitions among the non-subscribers become more severe, so the average payment per non-subscriber increases. Note that when there are less non-subscribers, say less than 3, the utility and the payment keeps the same. But as the number of non-subscribers increases, the non-subscribers' utilities decrease dramatically, which means the congestion influence the performance of the system a lot. The optimal price for the content owner, which equals to the negative value of the non-subscriber's utility is also increasing with the number of non-subscribers. Such phenomenon implies if there is no cooperation

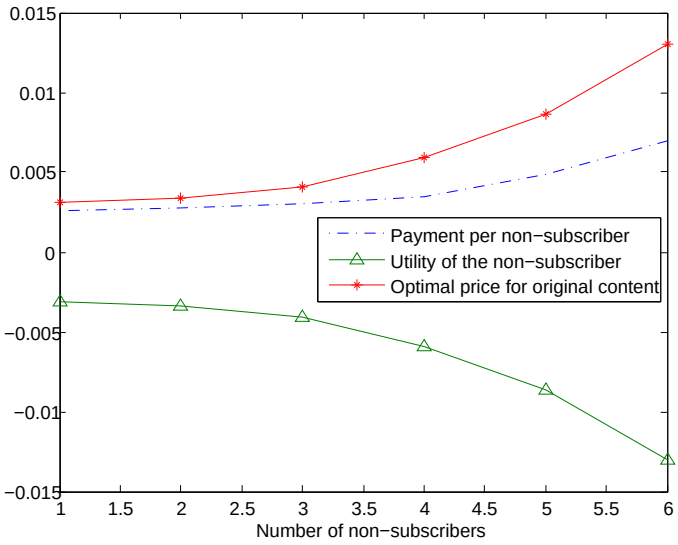


Figure 6.3: Multiple non-subscriber case with different numbers of non-subscribers

among non-subscribers, the live-video marketing social networks can only serve small number of users.

Then we will examine the relationship between the averaged distance among the non-subscribers and the optimal content price. We let the pirates be uniformly located within the range of [50 m, -50 m] in both x-axis and y-axis. The two non-subscribers are located at [40m, 0m] and [-40m, 0m] at the beginning, and gradually move toward each other. The number of pirates is fixed to be 10. Other settings are the same as the previous simulation and the subscription price P_o for the video sequence is set to be 0. Fig. 6.4 shows the optimal subscription price P_o versus the distance between the two non-subscribers. We can observe that as the distance between the non-subscribers increases, the higher the number of pirates that can sell the video to multiple non-subscribers. So the competitions among the non-subscribers become more severe, so the average payment per non-subscriber increases. The optimal price is almost doubled when the two non-

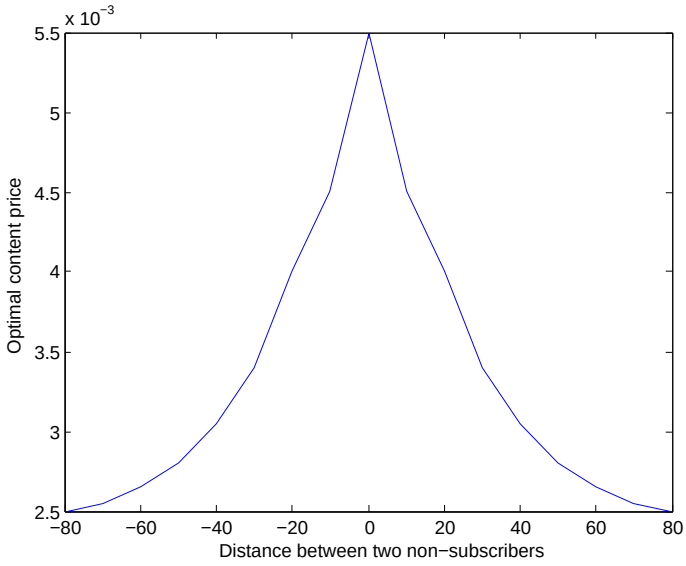


Figure 6.4: Multiple non-subscriber case versus distance between non-subscribers

subscribers are at the same position. Such phenomenons is because when the two non-subscribers are at the same position, they are completely competing over all resources.

6.4 Chapter Summary

In this chapter, we propose a game-theoretical approach for the optimal price setting over mobile live-video streaming social networks. We target to find the optimal price of the mobile live tv service by answering two questions: Which non-subscriber which buy buy the video from which pirate, and how much power should the pirates use for transmission? We propose a Auction-Stackelberg game to jointly consider the benefits of the pirates and the non-subscribers. The proposed scheme not only helps the non-subscribers optimally choose the pirates at better locations but also helps the competing pirate nodes ask optimal prices to maximize their utilities. Consequently, the results provide a guideline for the

content owner to set the price of the video content and prevent the pirate behavior. From the simulation results, the non-subscriber will tend to buy more power from the closer pirates to increase the his/her utility, and if the total number of the pirate increases, the non-subscriber can obtain a larger utility value, and the average payment to the pirates shrinks, due to more severe competitions among the sellers. On the other hand, if there are more non-subscribers within the range of the pirates, the average payment to the pirates will increase.

Chapter 7

Conclusions and Future Research

In this thesis, we have modelled and analyzed human behavior for multimedia social networks which involve a large number of users of different types with different objectives. Such an analysis helps to understand the impact and importance of human behavior factors on multimedia security and communications. Our study aims to stimulate user cooperation, facilitates the implementation of misbehavior monitoring mechanisms, and provides important guidelines on the design of cheat-proof and attack-resistant strategies. All these are essential factors to maximize the overall system performance and minimize the damage caused by malicious users.

We first took the multimedia fingerprinting social network as an example to study the behavior forensics and user dynamics. We answered the question of when and how will the collusion happen by analyzing the colluder social network. We defined a general utility function for the colluders and investigated the necessary conditions for multi-user collusion. Then, by modelling the colluders' behavior as a non-cooperative game, we found the fair collusion under different types of bargaining model, either the market value of the multimedia content is time-sensitive or not, to further reduce the possible collusion set and improved the traitor-tracing performance.

We then investigated the side information about the mean values of the detection

statistics can help the detector significantly improve the collusion resistance and proposed the self-probing detector to utilize such side information for detection. The simulation results demonstrate that the self-probing detector has approximately the same performance as the optimal fingerprint detector, and the difference between these two can be ignored. Since the self-probing detector can be considered optimal, it breaks the collective fairness equilibrium between the colluders and the fingerprint detector, and the colluders have to choose different strategies to achieve fairness. We model the colluder-detector dynamics with side information as a Stackelburg game and show that under the assumption that colluders demand absolute fairness of the attack, the min-max solution achieves the equilibrium which is the optimal strategy of all users in the multimedia fingerprint social network.

We also studied how to stimulate cheat-proof and attack-resistant cooperation that can ensure the efficiency under various traffic network and hostile environments in P2P live streaming social networks. An illustrating two-player game was studied, and different optimality criteria, including Pareto-Optimal, proportional fairness and absolute fairness is performed to refine the obtained Nash Equilibriums. Finally, a unique Nash equilibrium solution is derived, which states that, in the two-person live streaming game, a node should not help its opponent more than its opponent has helped it. Then the results were extended to stimulate multiuser live streaming, and combing with the chunk-request and request-answering algorithm, a fully-distributed attack-resistant and cheat-proof cooperation stimulation strategy has been devised for P2P live streaming social networks. Simulation results have illustrated that the proposed strategies can effectively stimulate cooperation among selfish peers in internet with various traffic and hostile environments.

We then analyzed the user dynamics in wireless P2P live video sharing systems. The trust modelling was incorporated to against hand-wash attack and the wireless fading channel. Both simulation and analytical results demonstrated that the trust modelling significantly reduced the damage caused by malicious attack, and did not influence the non-malicious users' utility when there are no attackers in the system.

In this thesis, we also addressed the optimal price setting for wireless video streaming to prevent users from reselling the video content on mobile devices. We proposed a mixed Auction-Stackelburg game and proved the equilibrium price between the re-sellers and the non-subscribers.

We hope that the frameworks presented in this thesis will encourage and stimulate researchers from different areas to further explore behavior modeling and forensics for multimedia social networks and beyond. It is an emerging research field with much uncharted territory remains unexplored. We envision that insights from a wide range of disciplines, such as signal processing, game theory, sociology, networking, communications and economics, will help improve our understanding of human dynamics and its impact on multimedia social networks, and ultimately lead to systems with more secure, efficient and personalized services.

Behavior analysis is at its young age, and there are many more interesting research directions that need to be further investigated.

First, our current work on wireless peer-to-peer live video sharing social networks consider a simple scenario that all users are watching the one and only one video, and each user can connect to all other users in the network directly. As we pointed out in Chapter 5, users in a wireless network are using different types of devices. Hence, the more powerful

nodes, such as laptops, can serve as supernodes to work as a small access point to help connect users that are far away from each other and increase the social warefare. Utilizing supernodes has been proven to significantly increase the system performance, therefore, it is important to stimulate powerful devices to serve as supernodes and investigate the optimal strategies for these users. Furthermore, due to the limitation of transmission power, the mobile devices has much smaller coverage than the access point. Therefore, in some cases, not all users within a wireless live video social network can communicate with all users in the network directly. In such scenario, some users must serve as relays or routers to complete the transmission. Consequently, designing incentives and optimal strategies for users to forward other users' packets as well as exchanging video chunks is of ample importance.

Also, our current work on live-video marketing social networks assume all users are honest and rational. Which means the pirate will transmit the live video to the non-subscribers using the power that he/she has agreed with. However, the pirate can increase his/her payoff by transmitting the video using less energy. Also, some attackers can maliciously bidding the live video for arbitrarily high price but refuse to pay the money by pretending not receiving the video. Such cheating or malicious behavior will definitely influence the system performance and the equilibrium price. Consequently, the original content owner's profit will also be influenced. Furthermore, the results showed if there are no cooperation among the non-subscribers, the live-video marketing social networks can only serve small number of users. Such phenomenon will stimulate cooperation among the non-subscribers to pay less for the pirated live video.

Finally, the online social networks such as youtube, wikipedia, and facebook are

becoming very popular and have millions of users worldwide. Users upload articles or videos to gain the reputation of the society as well as the attention of entertainment industries. However, the reputation is rated by other users in the network and the cheating and malicious behavior is very common in the online societies. It will be fruitful to investigate the user dynamics for online social networks and design the strategies to against attack and cheating. This investigation will lead to a more secured personal service, and provide a basis for the design of online platforms.

Bibliography

[1] H. Zhao, W. Sabrina Lin, and K. J. R. Liu, "Behavior modeling and forensics for multimedia social networks: A case study in multimedia fingerprinting," *IEEE Signal Processing Magazine*, January 2009.

[2] G. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M Levy, and J. Zahorjan, "Measurement, modeling and analysis of a Peer-to-Peer file-sharing workload," *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19)*, pp. 314 – 329, Oct. 2003.

[3] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in P2P file sharing systems," *IEEE InfoCom*, vol. 2, pp. 1174 – 1185, March 2005.

[4] Z. Liu, H. Yu, D. Kundur, and M. Merabti, "On Peer-to-Peer multimedia content access and distribution," *IEEE Int. Conference on Multimedia and Expo*, pp. 557–560, July 2006.

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

[6] S. Saroiu, G. P. Gummadi, and S. Gribble, "A measurement study of Peer-to-Peer file sharing systems," *Proceedings of Multimedia Computing and Networking (MMCN)*, Jan. 2002.

[7] G. Owen, *Game Theory*, Academic Press, 3rd edition, 1995.

[8] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, 1991.

[9] N. Naoumov and K. Ross, "Exploiting P2P systems for DDoS attacks," *Proceedings of the 1st international conference on Scalable information systems*, 2006.

[10] K. J. R. Liu, W. Trappe, Z. J. Wang, M. Wu, and H. Zhao, *Multimedia Fingerprinting Forensics for Traitor Tracing*, EURASIP Book Series on Signal Processing and Communications, Hindawi Publishing Corporation, 2005.

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

[12] F. Zane, "Efficient watermark detection and collusion security," *Proc. of Financial Cryptography, Lecture of Notes in Computer Science*, vol. 1962, pp. 21–32, Feb. 2000.

[13] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Tran. on Multimedia*, vol. 7, no. 1, pp. 43–51, Feb. 2005.

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

[15] H. Zhao and K. J. R. Liu, "Behavior forensics for scalable multiuser collusion: fairness versus effectiveness," *IEEE Tran. on Information Forensics and Security*, vol. 1, no. 3, pp. 311– 329, Sept. 2006.

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

[17] W. S. Lin, H. V. Zhao, and K. J. R. Liu, "A game theoretic framework for colluder-detector behavior forensics," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. II, pp. 721–724, April 2007.

[18] W. S. Lin, H. V. Zhao, and K. J. R. Liu, "Multi-user collusion behavior forensics: game-theoretic formulation of fairness dynamics," *to appear, IEEE Int. Conf. on Image Processing*, Sept. 2007.

[19] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transaction on Multimedia*, vol. 9, December 2007.

[20] "The software is available at http://www.pplive.com/en/index.html," .

[21] "The software is available at http://www.ppstream.com/," .

[22] "The software is available at http://www.uusee.com/," .

[23] X. Zhang, J. Liu, B. Li, and P. Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," *IEEE INFOCOM*, vol. 3, pp. 2102–2111, March 2005.

[24] Ahsan Habib and John Chuang, "Incentive mechanism for peer-to-peer media streaming," *International Workshop on Quality of Service (IWQoS)*, pp. 171–180, June 2004.

[25] X. Hei, Y. Liu, and K.W. Ross, "Inferring network-wide quality in P2P live streaming systems," *IEEE Journal on Selected Areas in Communications*, vol. 25, Sep 2003.

[26] G. Tan and S. A. Jarvis, "A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast," *In Proceedings of International Workshop on Quality of Service (IWQoS)*, June 2006.

[27] Z. Liu, Y. Shen, S. Panwar, K. Ross, and Y. Wang, "Using layered video to provide incentives in P2P live streaming," *ACM Special Interest Group on Data Communication*, August 2007.

[28] W. Sabrina Lin, H. Vicky Zhao, and K. J. Ray Liu, "Incentive cooperation strategies for peer-to-peer live streaming social networks," *IEEE Transaction on Multimedia*, vol. 11, no. 3, pp. 396–412, April 2009.

[29] W.S. Lin, H.V. Zhao, and K.J.R. Liu, "A game theoretic framework for incentive-based peer-to-peer live-streaming social networks," *Proc. IEEE Int'l Conf. Acoustic, Speech, and Signal Processing (ICASSP)*, 2008.

[30] S.C Kim, M. G. Kim, and B. H. Rhee, "Seamless connection for mobile p2p and conventional wireless network," *In Proceedings of International Conference on Advanced Communication Technology*, vol. 3, pp. 1602–1605, Feb 2007.

[31] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Key agreement in peer-to-peer wireless networks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 467–478, Feb. 2006.

[32] S. Ghandeharizadeh, B. Krishnamachari, and Shanshan Song, "Placement of continuous media in wireless peer-to-peer networks," *Multimedia, IEEE Transactions on*, vol. 6, no. 2, pp. 335–342, April 2004.

[33] S. Ghandeharizadeh and T. Helmi, "An evaluation of alternative continuous media replication techniques in wireless peer-to-peer networks," in *MobiDe '03: Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, New York, NY, USA, 2003, pp. 77–84, ACM.

[34] W. Sabrina Lin, H. Vicky Zhao, and K. J. Ray Liu, "Cheat-proof cooperation strategies for wireless live streaming social networks," *Proc. IEEE Int'l Conf. Acoustic, Speech, and Signal Processing (ICASSP)*, 2009.

[35] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, pp. 931-933, 1972.

[36] R. Nickalls, "A new approach to solving the cubic: Cardan's solution revealed," *The Mathematical Gazette*, vol. 77, pp. 354–359, 1993.

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

[38] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2001.

[39] 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.

[40] I. Cox and J.P Linnartz, "Some general methods for tampering with watermarking," *IEEE Journal on sel. areas of Comm.*, vol. 16, no. 4, pp. 587–593, May 1998.

[41] G. Doerr, J. L. Dugelay, and L. Grange, "Exploiting self-similarities to defeat digital watermarking systems: A case study on still images," *Proceedings of the 2004 ACM Multimedia and Security Workshop*, 2004.

[42] D. Kiroski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1045–1053, 2003.

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

[44] Z. J. Wang, M. Wu, W. Trappe, and K. J. R. Liu, "Group-oriented fingerprinting for multimedia forensics," *EURASIP Journal on Applied Signal Processing, Special Issue on Multimedia Security and Rights Management*, vol. 2004, no. 14, pp. 2142–2162, Nov. 2004.

[45] S. He and M. Wu, "Joint coding and embedding techniques for multimedia fingerprinting," *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, pp. 231–247, June 2006.

[46] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*, Prentice Hall, 1st edition, 2001.

[47] 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.

[48] W.S. Lin, H. V. Zhao, and K. J. R. Liu, "Scalable multimedia fingerprinting forensics with side information," *IEEE Int. Conf. on Image Processing*, Oct. 2006.

[49] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer Verlag, 2nd edition, 1999.

[50] W.S. Lin, H. V. Zhao, and K. J. R. Liu, "A game theoretic framework for colluder-detector behavior forensics," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, May. 2007.

[51] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for sharing in peer-to-peer networks," *ACM Electronic Commerce (EC 01)*, Oct. 2001.

[52] C. Buragohain, D. Agrawal, and S. Suri, "A game theoretic framework for incentives in P2P systems," *In Proceeding of the International Conference on Peer-to-Peer Computing*, pp. 48–56, Sep 2003.

[53] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," *ACM Conference on Electronic Commerce (EC-04)*, May 2004.

[54] W. Yu and K.J.R. Liu, "Game theoretic analysis of cooperation and security in autonomous ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 5, pp. 507–521, 2007.

[55] M.J. Osborne and A. Rubinste, *A Course in Game Theory*, The MIT Press, 1994.

[56] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Publishing, 2nd edition, 2000.

[57] A. Tourapis, K. SAuhring, and G. Sullivan, "Revised H.264/MPEG-4 AVC reference software manual," *Joint Video Team, Doc. JVT-Q042*, October 2005.

[58] J. Liang, R Kumar, Y Xi, and K. Ross, "Pollution in P2P file sharing systems," *In Proceeding of IEEE Conference on Computer Communications (INFOCOM)*, vol. 2, December 2005.

[59] O. Kallenberg, *Foundations of Modern Probability.*, Springer-Verlag, 1977.

[60] D. Marpe H. Schwarz and T. Wiegand, "Joint scalable video model (JSVM) 2," *Joint Video Team, Doc. JVT-O202*, April 2005.

[61] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," *In Proceeding of 33rd Asilomar Conf. Signals, Systems and Computers*, October 1999.

[62] O.Karonen and J.K.Nurminen, "Cooperation incentives and enablers for wireless peers in heterogeneous networks," *IEEE International Conference on Communications*, , no. 134-138, May 2008.

[63] T. Ozbilgin and M.O. Sunay, "On the capacity of wireless peer-to-peer networks with transmitter and receiver cooperation," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, , no. 1-5, Sept 2007.

[64] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *Information Theory, IEEE Transactions on*, vol. 51, no. 9, pp. 3037–3063, Sept. 2005.

[65] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications, Englewood Cliffs*, Prentice Hall, 1983.

[66] Y. Sun, Z. Han, and K.J. Ray Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE Communication Magazine*, vol. 42, no. 6, pp. 112–119, Feb. 2008.

[67] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 42, no. 2, pp. 618–644, 2005.

[68] Sinjae Lee, Shaojian Zhu, and Yanggon Kim, "P2p trust model: The resource chain model," 30 2007-Aug. 1 2007, vol. 2, pp. 357–362.

[69] International Telecommunication Union, “”http://www.itu.int/itu-d/ict/statistics/ict/graphs/mobile.jpg”,” .

[70] Alessandro Lizzeri and Nicola Persico, “Uniqueness and existence of equilibrium in auctions with a reserve price,” *Games and Economic Behavior*, vol. 30, no. 1, pp. 83–114, January 2000.