

Detecting Bad Mouthing Behavior in Reputation Systems

Kuan-Ta Chen (Chun-Yang Chen)¹, Cheng-Chun Lou², Polly Huang², and Ling-Jyh Chen¹

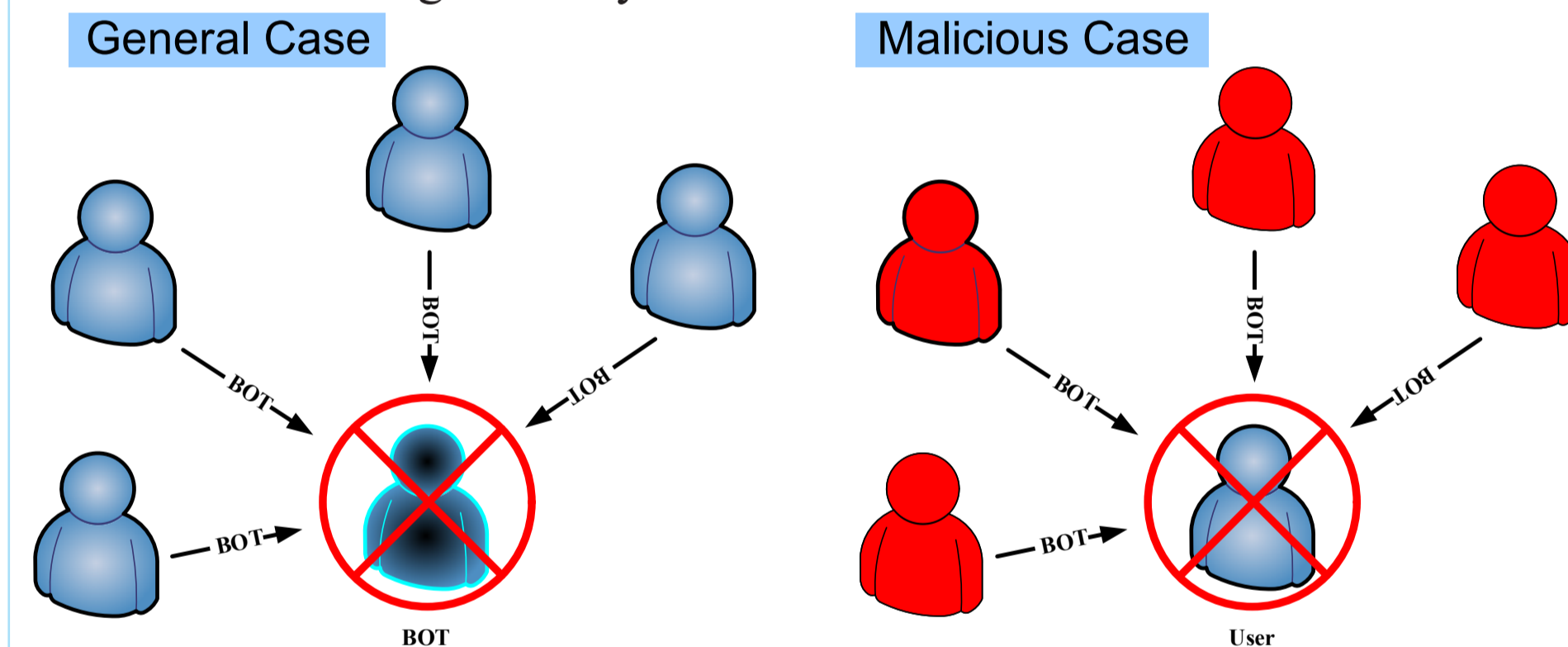
¹Academia Sinica, ²National Taiwan University

Background

- MMORPGs (Massively Multiplayer Online Role-Playing Games) have become extremely popular
- Game bots
 - Auto-playing game clients
 - One of the greatest threats of MMORPGs
- Detection of Game Bots
 - Manual detection (game master) [1]
 - Traffic analysis approach [2]
 - Voting-based system [3]
 - Each player votes the suspicious player as a game bot

Motivation

- Problem in voting-based system



- Collusion
 - A secret agreement between two or more parties for a fraudulent, illegal, or deceitful purpose [4]
 - Unfairly low ratings – bad mouthing
 - Unfairly high ratings – ballot stuffing
- Only can vote negatively (game bot)
 - **This study focuses on bad-mouthing attacks**

Problem Formulation

- Bad-Mouthing
 - A malicious group deliberately vote a legitimate player as a game bot
- Terms
 - Collusion Cluster: a bad-mouthing group
 - Victim: the legitimate players who are under bad-mouthing attacks
- Goal

To Detect the Collusion Clusters

Hypothesis

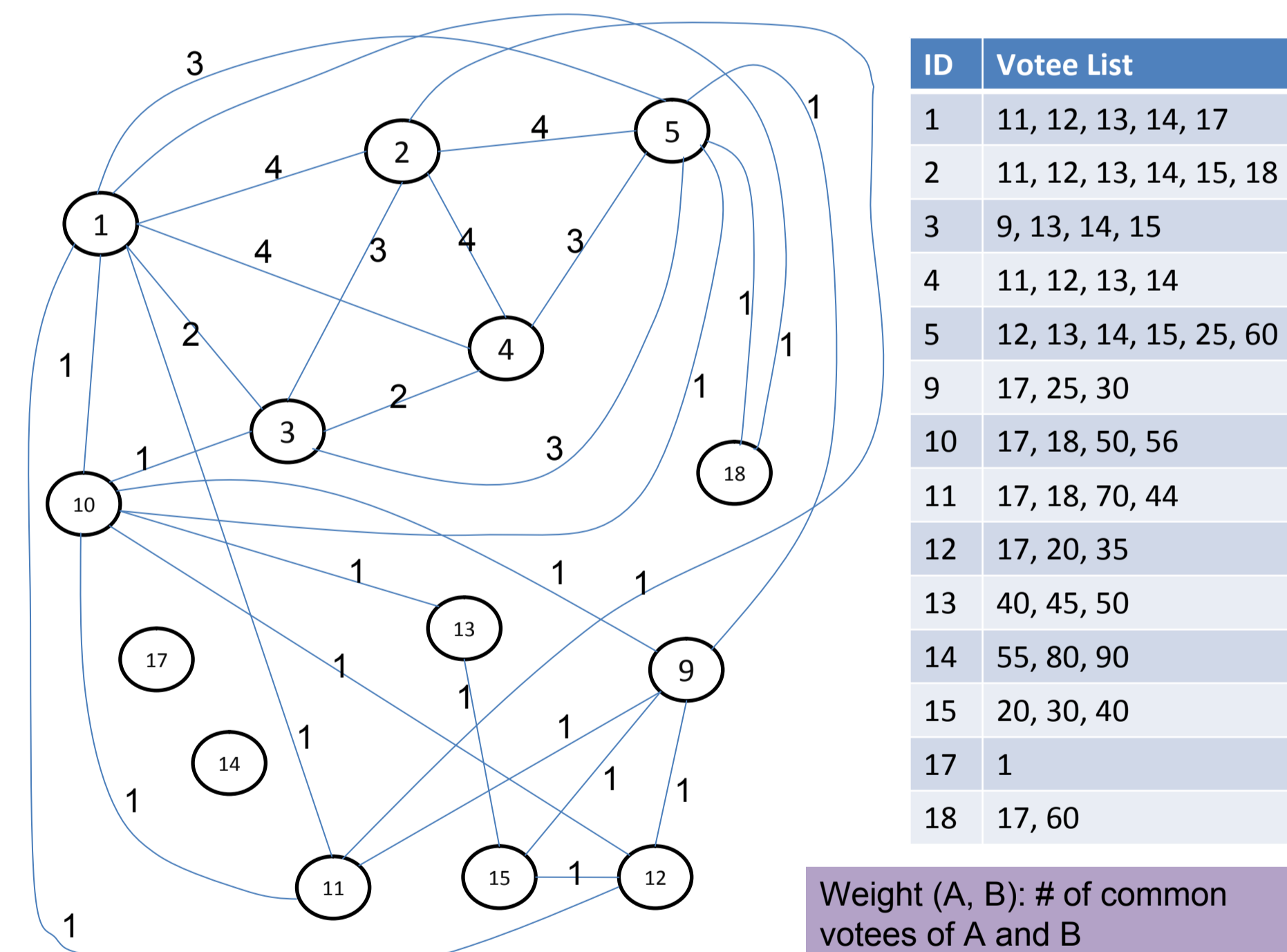
- The most voters of legitimate players are likely collusion cluster
- In case if a collusion cluster attacks for several times
 1. Collusion Cluster has more **common** votees than random voter cluster
 2. Victims have more **common** voters than random votee cluster
- Based-on
 - The voters & votees id of each player
- Note
 - When not attack, a collusive player acts as a legitimate player

Voter-Based Collusion Cluster Detection

The relationship of collusion cluster is stronger than random voter cluster

Take the voters with **more common votees** between each other

- form the collusion cluster.



Weight (A, B): # of common votees of A and B

```

< Algorithm >
while (edge(G) ≠ ∅)
    Take the edge with the largest weight
    while (not termination condition)
        Take the outlier edge with the largest weight
    end while
    output
end while
    
```

Terminal Condition:

$$\frac{\sum \{weight(u,v) | u, v \in S'\} - \sum \{weight(u,v) | u, v \in S\}}{|S|} < a$$

Votee-Voter-based Collusion Cluster Detection

The relationship of victim group is stronger than random votee cluster

First take the votees with more common voters

- form the victim group

Then take union of the voters of the victim group

- form the collusion cluster candidate

Apply Voter-based Collusion Cluster Detection

Performance Evaluation Results

	Voter-Based	Votee-Voter-Based	Comparison
# of Attack (Single CC or Multiple CC)	More attacks → higher accuracy		Votee-Voter > Voter
Collusion Cluster Size	No influence → high accuracy	Need more than 3 → high accuracy	Voter > Votee-Voter
Prob. of Collusive Player Attack Vote	Higher Prob. → higher accuracy		Votee-Voter > Voter

Voter-Based scheme:

Robust to the size of collusion cluster

Votee-Voter-Based scheme:

Robust to the number of attacks

Conclusion

- Two mechanisms to detect the collusion cluster
 - Based on the voting history
 - Single cluster or multiple clusters
- Accuracy
 - Attack more than three times: 83%+
 - Attack more than five times: 97%+
- Adjust other experimental factors
 - Only collusion cluster size and prob. of collusive player attack vote have the obvious influence to the accuracy

Future Work

Detecting players who participate in multiple collusion clusters

- [1] I. MacInnes and L. Hu, "Business models and operational issues in the chinese online game industry," Telematics and Informatics, vol. 24, no. 2, pp. 130-144, 2007
- [2] K.-T. Chen, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen, "Identifying mmorpg bots: A traffic analysis approach," in Proceedings of ACM SIGCHI ACE, 2006
- [3] Blizzard, <http://www.blizzard.com/war3/>
- [4] <http://www.answers.com/collusion?cat=biz-fin>