DI Philipp Schosteritsch
Amschlgasse 29a, 8010 Graz
s.philipp@schoste.com

Fighting an Almost Perfect Crime

Online Poker Fraud Detection

Philipp Schosteritsch, MSc
Consultant for Software Development and Information Management
Technical Advisor on Cyber Security & Privacy Foundation
Graz, 2014

Contents

Abstract

Online-Poker sites are a great place for gambling at any time. One advantage of playing poker over the Internet is that game rules are strictly enforced by the game software. The great pit fall is that you don't see the people you play with. And since you don't see them you don't know if they cheat.

This paper addresses the danger of players who collude to win over unknown individual players. This means some players exchange information of their hands during a game to minimize their own losses and to maximize their wins.

Hosts (gaming sites/service providers) who do nothing against cheaters will lose clients as soon as more and more players feel betrayed. These hosts have no long term perspective. Host who actively fight cheaters and fraud signal commitment and trustworthiness to their clients and create an environment for regular clients.

This paper will propose an algorithm to discover such collusions by the host.

DI Philipp Schosteritsch
Amschlgasse 29a, 8010 Graz
s.philipp@schoste.com

Introduction

At least since the James Bond movie "Casino Royale" playing Poker became popular and Online-Poker sites appeared en-masse. Links to dodgy sites can be found in your Spam-Mail folder and links to more trustworthy sites can be found in the top results of the search engines.

Playing Poker requires money. An Online-Poker site therefore receives money from its clients (or the players) and therefore these sites are attractive targets for criminal individuals. The question for most of the criminals is "How do I transfer the players' money to my account without being caught?"

The attempt of a black hat hacker or a criminal with an affinity to information technology is most likely looking for a flaw in server software or the host's network.

A less technical approach is exploiting a vulnerability of how the game is implemented. In an Online-Poker game you cannot have aces up your sleeves and there is no dealer who may lack attention. That's one side of the table. The other side is that in an Online-Poker game you don't know what your play mates are doing (besides playing the online game). You have no control about what they are talking, if they are consulting or using software to compute the chances of their hands.

And exactly that's the vulnerability: The ability for players to cooperate. Players on a table can gang up and play together against other players. They can exchange information about their hands via Skype, Phones or other side channels. These are called side channels because Online-Poker sites usually provide a channel for players to chat. But the official way to chat may be public and is control- and observable by the host. Using these side channels is undetectable by the host. Even if you know people are communicating with each others, how can you know or even prove they are cheating by talking about their hands? It is improvable; it is hard to detect so it is almost perfect!

Collusion is among other poker frauds a known problem [1]. Poker fraud in general is already a domain that gets attention from specialists [2] as well as from communities [3].

In this paper an attack scenario will be shown where collusion minimizes the risk of losing (and the amount of money lost) and a scenario will be shown where collusion can be used to maximize the amount of money won. Finally an algorithm will be presented to detect possible collusions of players.

DI Philipp Schosteritsch
Amschlgasse 29a, 8010 Graz
s.philipp@schoste.com

Collusion

In two examples it will be shown how collusion can be used to exploit a game and other players.

Using Collusion to Minimize Losses

Collusion of players can be used to minimize losses by detecting situations where it is impossible to ever get a winning hand. Let’s take a look on the following scenario:

Five players play Texas Hold’em. Player 1, 2 and 3 are working together. Player 4 and 5 are regular players. The flop shows five spades, six hearts, eight diamonds (see **Fehler! Verweisquelle konnte nicht gefunden werden.**). The turn and the river are yet to come.

Player one holds seven diamonds and eight clubs. She or he needs either a nine or a four in any color to have a straight. (see **Fehler! Verweisquelle konnte nicht gefunden werden.**)

Player two has a pair: Nine in diamonds and nine spades (see Figure 3).

Player three also has a pair: four in hearts and a four diamonds.

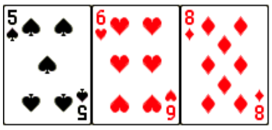

Figure 2:
Flop cards

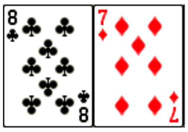

Figure 1:
Player one's hand

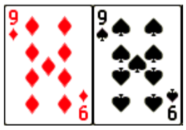

Figure 3:
Player two’s hand

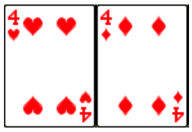

Figure 4:
Player three's hand

If player one did not know the cards of his or her co-players the odds to hit the right card was 12 / 47 (which equals to roughly 25.5 %). There are 52 cards in total. Five cards are known to player one, so 47 remain unknown. There are six possible cards for player one and two more turns to receive one of them. The overall chance for player one to win would be 41.6 % [1].

But since player one knows the cards of her or his co-players the computation of the odds bases on new facts. There are now only 43 cards unknown. Since player two holds a pair of nines and player three holds a pair of fours, only two cards remains as possible hits. The odds to hit a straight change to 2 / 43 (which equals to roughly 4.7 %).

By knowing the low odds to hit a straight player one will now rather check or fold than hit or raise. Therefore player one (and the colluding players) can minimize the losses she or he would have when betting on the small odds.

Using Collusion to Maximize Wins

Collusion of players can also be used to maximize winds (most likely more effectively than reducing losses). This happens when a player learns he or she has the best hand on the table and neither the river nor the turn card will change the odds.

- The flop shows a king in diamonds and two aces - shades and hearts (see Figure 5).
- Player one holds a pair of kings - hearts and shades (see Figure 6).
- Player two holds an ace and a two in clubs (see Figure 7).
- Player three holds an ace in diamonds and a five in shades.
- Player one, player two and player three are working together.

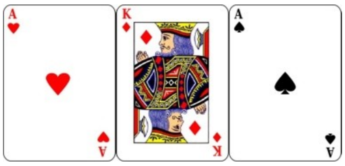

Figure 7:
The Flop cards (maximize win)

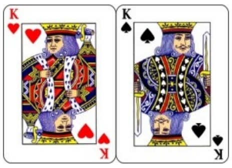

Figure 5:
Player one's hand

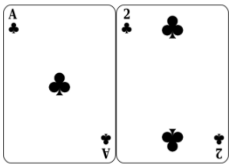

Figure 6:
Player two's hand

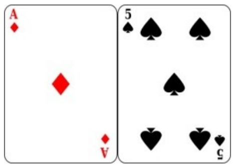

Figure 8:
Player three's hand

The situation above shows a very good position for player one. She already has a (small) Full House. Player two has a good position too – he already has three of a kind. So does player three.

Player one can be only beaten by another player who has four of a kind. This requires such a player to have a pair in her hands and the river and turn to show the remaining pair. A player who holds a pair has a winning chance of less than one percent after the flop [1]. If there is no other pair than the aces in the shared cards, player one will definitely win. No higher full house than two aces and three kings is possible.

So before show down player one knows if she will win or lose. Therefore she can safely bet as much as she wants and player two and player three may raise the pot by calling each raise.

DI Philipp Schosteritsch
Amschlgasse 29a, 8010 Graz
s.philipp@schoste.com

Detecting Collusion

To detect collusion it is necessary to identify players who cooperate. For players who want to cooperate it is necessary to play on the same table. So the idea behind the algorithm to detect collusion is to create a model for games that highlights reoccurring constellations of players.

Suitable for such a model are adjacency matrices [2]. In an adjacency matrix the connections of nodes in a graph are stored. If a node may have several edges to the same node, the number of edges can be used as value. In this algorithm players are the nodes and each encounter in a game is an edge.

The algorithm work with one global (player comprehensive) matrix and with multiple local (player limited) matrices. A global matrix would contain all registered players (this could be more than several thousand). A local matrix is kept for each player. It would contain all players the player encountered in games. In the local matrix the diagonal values are the number of games a player played in total (the diagonal values are how often a player encountered him- or herself). In a local matrix it is the number of times a player played in from log in to log out.

The local matrix is created when a player logs in and it is abandoned when the player logs out. The local matrix is initialized to only one node – the player him- or herself – and a value of zero. The local matrix is rewritten each time the player encounters a new player he or she didn't play with before. The local matrix is updated before each game starts.

The global matrix is created when the server(s) start(s). It contains all registered players and the edges are initialized to zero. The global matrix is rewritten each time a new player logs in for the very first time. The global matrix is updated every time a player logs out with the value of his or her local matrix.

The global matrix displays the evolution of the usage trend. In the main diagonal the absolute number of games a user played is stored. In the cells of the matrix the number of times a user encountered another user in all of his or her games is stored.

The local matrix displays the number of games the users played since logging-in in the main diagonal. In the cells of the matrix the number of times a user encountered another user in his or her games since logging-in is stored.

From the encounter rates in the global matrix ordinary statistic indicators like mean encounter rate and standard deviation can be computed. To those players who have significantly high encounter rates special attention should be plaid when they play on the same table again. These players might collude. To get further confidence about the situation further analysis on their behavior in the games needs to be done (e.g. winning rates at show down compared to winning rates without show down, etc.)

DI Philipp Schosteritsch
Amschlgasse 29a, 8010 Graz
s.philipp@schoste.com

Conclusion

By updating the global matrix from the local matrices and computing statistical indicators is a fast and simple step to reveal players who are possible playing together. This document does not explain which indicators should be used. There is also no recommendation for any pre- or post processing of the data. These steps can be individual for each host. E.g. if there is a high rate of user who sign up, only play a few games and then never log in again it may make sense to start counting users after a certain numbers of log-ins (as pre-processing step). Simple statistical indicators could be mean value and standard deviation.

The algorithm can be implemented at relatively low costs of memory and computing power. More sophisticated and detail fraud detection analysis can be done for players who are detected by this algorithm.

Further, the created adjacency matrices can be used for further analytics and statistics.

References

[1] Poker Tournament Terminiator, "Cheating in Online Poker," [Online]. Available: http://www.tournamentterminator.com/tournament-strategy/online-poker/cheating-collusion-fraud/. [Accessed 2 April 2014].

[2] M. Bruley, "Full Tilt Poker: Big Data Fraud Detection and Prevention," 6 May 2013. [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/full-tilt-poker-big-data-fraud-detection-and-prevention. [Accessed 2 April 2014].

[3] T. Gullen, "Detecting an online poker cheat," 3 September 2010. [Online]. Available: http://stackoverflow.com/questions/3635216/detecting-an-online-poker-cheat. [Accessed 2 April 2014].

[4] pokernews.com, "Poker News, Online Poker Reviews," 2013. [Online]. Available: http://de.pokernews.com/poker-tools/poker-odds-calculator.htm. [Accessed 04 April 2014].

[5] Wikimedia Foundation, Inc., "Adjacency matrix," 13 March 2014. [Online]. Available: https://en.wikipedia.org/wiki/Adjacency_matrix. [Accessed 30 April 2014].