LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

Gaming Self-Contained Provably Fair Smart Contract Casinos: Open Review

Author: Piotr J. Piasecki*†

Reviewers: Reviewer A, Reviewer B

Abstract. The final version of the paper "Gaming Self-Contained Provably Fair Smart Contract Casinos" can be found in Ledger Vol. 1 (2016) 99-110, DOI 10.5915/LEDGER.2016.29. There were two reviewers, none of whom have requested to waive their anonymity at present, and are thus listed as A and B. After initial review (1A), the author submitted a revised submission and response (1B). The assigned Ledger editor determined that the author had adequately addressed the reviewer concerns and asked the author for minor revisions which were carried out by the author, completing the peer-review process. Author's responses are in bullet form.

1A. Review

Reviewer A:

This paper explores the intriguing question of whether one can implement fair, probabilistic betting on a blockchain in which miners themselves contribute the wagers. In my opinion, the main contribution of this paper is a suggestion that one can improve the fairness of a gambling system by introducing a delay between the time a bet is placed and the time the payoff bits are revealed. Scenario 2 describes the details of this improvement. The topic of the paper is novel and should appeal to a broad range of mathematicians and computer scientists.

Unfortunately, the analysis in the present work suffers from some technical flaws. For example, the derivative at the bottom of page 3 is computed incorrectly, and the inequality on the last line of page 4 is flipped. In the paragraph on page 2 which concludes with "0=0," I don't understand why there is a "1/p" factor or a "-w" in the formula for EV_win, a quantity propagated throughout the manuscript. I'll detail a few other concerns below. In general the author's long, algebraic calculations include trivial steps and extraneous manipulations. "Scenario 5" rehashes calculations from earlier in the paper verbatim.

†P. J. Piasecki (ThePiachu@gmail.com) is a Master of Computer Science from Technical University of Lodz, Poland. Currently a software developer in Vancouver, Canada.
*1PiachuEVn6sh52Ez7o6Fymvw54qvQ4RBm

The paper's language lacks precision in places and omits certain important details. Here are a few specific questions and comments.

1. In the underlying scenario, who is playing this gambling game? Who is the "casino" party represented by parameter "h," and are there other potential parties besides miners? The story here is mostly told in first person. In Scenarios 1, 3, all cases are either a win or break even for us, and in Scenario 5, we *always* win. Who is the sucker on the other end of these wagers?

2. What are the assumptions about the system? Are we playing on Ethereum, Bitcoin, or some abstraction of one of these systems? What is the mechanism behind the game, and in general what choices of moves do players have at each step? The first line on page 2 reads says that "the probability of winning a game is denoted by p. It is often chosen by the player." Do you mean to say that players choose their own success probabilities?

3. Section 4 makes no distinction between value and expected value for random variables. Please clarify. EV_win is first defined to be the amount the gambler expects to earn if he wins a bet, but later in the section it says "… in a won game, the miner will earn… the winning reward [of] EV_win."

4. The paper's abstract and introduction mention "smart contracts," but the author doesn't explain what smart contracts are or what role they play in the gambling system. How are contracts arranged between parties, and how exactly do they agree on random bits?

5. In the first bullet point of Scenario 1, should this be "c + EV_win" rather than "1 + EV_win"?

6. The probability (m^n)*p in the first bullet of Scenario 3 where "we mine the n blocks and win the bet," is only a lower bound, not equality. The private and public chains need not extend at the same rates. The explicitly say how it arrives at this particular value.

8.[*sic*] I don't understand the assumptions in Scenario 5. Presumably the actors are "irrational" because they ignore their mining rewards, a strange assumption in and of itself. But then the game is set up so that the actors always win, and therefore "the irrational actor will always play the game." Are the irrational actors really irrational after all?

The typesetting in this paper could be improved. The author might consider switching from Microsoft Word to LaTeX in light of the number of mathematical formulas used in the paper. The notation also could be made more standard: normally one uses a dot rather than "*" for multiplication, and the expressions using the symbol "|/" at the end of some lines are not needed. Also, separate-line expressions don't need "=" on both the left and the right; just putting this symbol on the left is sufficient.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

Although the premise of this paper is rather interesting, I don't recommend it for publication in Ledger due to lack of attention to important details in some places and overemphasis of mathematical trivialities in others. It is immediate from the descriptions of the games in this paper what the expected winnings should be, and therefore the formal analyses here offer negligible substance.

Reviewer B:

> or c*m coins per block on average

Correction: or c/m coins per block on average.

> Scenario 2 - lottery with a delayed resolution

I think this paragraph should do a better job clarifying that what the attacker is trying to do is potentially not publish the block that _decides_ the outcome of the bet, and it's that block that matters, not the block that the bet was included in.

> Probability of this is m*(1-p). In this case, we discard the block and the bet, netting 0

The logic here feels a bit iffy, as the bet isn't really "discarded" as another miner will create the next block (or possibly yourself) and it will be that same bet that gets resolved. However, I suppose you could view it as being equivalent to "discarding" the bet with zero payoff and immediately publishing a new one so it shouldn't change the calculations.

I also get the instinct that the reason why high-p gambles lead to such seemingly high security levels is actually quite boring: high-p gambles with nonzero house take (ie. $h < 1$) are actually a very bad proposition for a gambler to take in a certain mathematical sense. To see why, consider a p=0.96, h=0.98 gamble. You can consider it being equivalent to the sum of two of the following gambles: (i) putting \$0.5 into a p=0.96, h=1 gamble, (ii) putting \$0.5 into a gamble which 96% of the time just gives you your money back, and 4% of the time takes it all. The first gamble is purely fee-free, the second is purely fee, so in some sense you could view the "fee" of that gamble as _actually_ being 50% (another way of viewing this is that the maximum gains without the fee would be ~0.04, but with the fee the maximum gains are ~0.02, but the losses in both cases are the same, so the "cost" of the "potential upside" is doubled). Of course games that are skewed against the player are hard to exploit, they're so skewed you need to have really really favorable conditions in order for even an exploiter to win at them!

Aside from this, generally solid analysis.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

1B. Author's Response

Aside from the changes described below and deletions annotated by strikethroughs, the document has had the following small changes done that were not highlighted:

- Small grammatical fixes in capitalization, hyphenation, punctuation, etc.
- Changing *s to dots, moved equation signs, etc. as suggested by Reviewer A to simplify the equations' look
- Changing EVwin and EVloss to Pwin and Ploss
- The "c" variable has been removed altogether as it only created confusion

Reviewer A:

This paper explores the intriguing question of whether one can implement fair, probabilistic betting on a blockchain in which miners themselves contribute the wagers. In my opinion, the main contribution of this paper is a suggestion that one can improve the fairness of a gambling system by introducing a delay between the time a bet is placed and the time the payoff bits are revealed. Scenario 2 describes the details of this improvement. The topic of the paper is novel and should appeal to a broad range of mathematicians and computer scientists.

Unfortunately, the analysis in the present work suffers from some technical flaws. For example, the derivative at the bottom of page 3 is computed incorrectly, and the inequality on the last line of page 4 is flipped. In the paragraph on page 2 which concludes with "0=0," I don't understand why there is a "1/p" factor or a "-w" in the formula for EV_win, a quantity propagated throughout the manuscript. I'll detail a few other concerns below. In general the author's long, algebraic calculations include trivial steps and extraneous manipulations. "Scenario 5" rehashes calculations from earlier in the paper verbatim.

- Derivative at page 3 has been corrected.

- The inequality on the last line of page 4 (0>m-mp) is correct - it is taken from the equation "w*(h-1+m*(1-p))>m*(1-p)" for a scenario where the left side of the inequality is equal to 0.

- The "0=0" equation was presented in its form to give a better order of introducing the EVloss, EVwin, EV. The order has been changed in the revision to better illustrate how EVwin was derived instead.

- Scenario 5 appears to be rehashing the calculations, although with a key difference - equation (1) is there to illustrate that the payout for winning always has to be positive and imposes a new boundary condition on p. In Scenario 5, we are looking for cases where EV is positive (rather than EV_win). However, since in that scenario EV = EV_win, the value is the same.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

The paper's language lacks precision in places and omits certain important details. Here are a few specific questions and comments.

1. In the underlying scenario, who is playing this gambling game? Who is the "casino" party represented by parameter "h," and are there other potential parties besides miners? The story here is mostly told in first person. In Scenarios 1, 3, all cases are either a win or break even for us, and in Scenario 5, we *always* win. Who is the sucker on the other end of these wagers?

- The game is created by a casino that is taking a passive role in the game. They set up the smart contract, provide it with money, but otherwise take no active part in running the system. Each bet is initialized by the player and resolved by a mined block. The player examined is a malicious miner that aims to skew the game in their favour by strategically creating and withholding blocks. The "sucker on the other end" is the casino that creates the smart contract game. The point of the paper is to explore whether such a contract can run on its own and not lose money if a miner decides to attack it, or whether it is infeasible. While with unlimited bet size, Scenarios 1 and 3 give us a non-negative outcome, the smart contract creator can limit the bet size to prevent the games from being susceptible to an attack. For Scenario 5, the only way to protect the smart contract from an attack would be to reach the limit of how many blocks can feasibly be overwritten by an attacker.

2. What are the assumptions about the system? Are we playing on Ethereum, Bitcoin, or some abstraction of one of these systems? What is the mechanism behind the game, and in general what choices of moves do players have at each step? The first line on page 2 reads says that "the probability of winning a game is denoted by p. It is often chosen by the player." Do you mean to say that players choose their own success probabilities?

- The presented game could be played on a system like Ethereum, or any other blockchain that supports similar turing-complete script.

- A lot of games can be simplified to fit into the scenario. Simplest ones are like Satoshi Dice or Just Dice - choose a target, bet high / low, and see what the RNG rolls. However, more complicated games like the Roulette or Baccarat can also be boiled down to the same game - picking your numbers, place a bet, see what the game rolls. In a lot of those games the probability of winning is flexibly set by the player (in Just Dice you can set any number as your target, in Roulette you can pick from picking a single number to bet on to betting on half of the board), and your payouts for winning adjust accordingly.

- The player is free to choose their probability of winning as well as how much they bet, and the payouts are determined by the game (Usually approximating Payout = Wager * House Edge / Probability)

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

3. Section 4 makes no distinction between value and expected value for random variables. Please clarify. EV_win is first defined to be the amount the gambler expects to earn if he wins a bet, but later in the section it says "… in a won game, the miner will earn… the winning reward [of] EV_win."

- The terminology has been changed to be more in-line with game theory standards - EV_win and EV_loss have been renamed to P_win and P_loss - Payoff, instead of Expected Value.

4. The paper's abstract and introduction mention "smart contracts," but the author doesn't explain what smart contracts are or what role they play in the gambling system. How are contracts arranged between parties, and how exactly do they agree on random bits?

- As the Ledger Journal aims to publish articles on the subject of cryptocurrency and the blockchain technology, the author assumes reader's familiarity with the concept of smart contracts.

- The randomness is derived from block and transaction hashes, as noted in the Assumptions. The part of the paper explaining those assumptions has been expanded to state that more clearly.

- Contracts are created by the casino, and the players play the game by sending transactions to the contract. The resolution is based on transaction / block hashes being used as random bits. In that way, it is similar to SatoshiDice, but instead of using external random numbers, the system relies on block hashes to add more randomness to the system, and it is this part that the attackers will be attacking.

5. In the first bullet point of Scenario 1, should this be "c + EV_win" rather than "1 + EV_win"?

- For all equations, EV and w are expressed in relationship to the mining reward, so "1" in this case does mean "c". As that notation is confusing, the paper has been updated to remove "c" altogether for clarity.

6. The probability (m^n)*p in the first bullet of Scenario 3 where "we mine the n blocks and win the bet," is only a lower bound, not equality. The private and public chains need not extend at the same rates. The explicitly say how it arrives at this particular value.

- My mistake - I didn't consider the chains extending independently. I revisited the scenario and fixed it with a proper binomial probability calculation.

8. I don't understand the assumptions in Scenario 5. Presumably the actors are "irrational" because they ignore their mining rewards, a strange assumption in and of itself. But then the game is set up so that the actors always win, and therefore "the irrational actor will always play the game." Are the irrational actors really irrational after all?

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

- As explained in the first paragraph, those irrational actors do not focus on maximizing their earnings, but rather on maximizing casino's losses. These could for example take a form of a competing casino spending its money to drive competition out of business early on to maximize its long-term profits for example. Those actors were labelled irrational, as their behaviour is not economically rational within the confines of the game - they don't try to maximize their income, but instead are motivated by external factors. An analogy would be 51% attacker in Bitcoin - rational miners would not do a 51% attack since it would destroy their wealth, but irrational actors might want to attack Bitcoin to destroy it at a cost to themselves. We've seen this with Eligius and CoiledCoin.

The typesetting in this paper could be improved. The author might consider switching from Microsoft Word to LaTeX in light of the number of mathematical formulas used in the paper. The notation also could be made more standard: normally one uses a dot rather than "*" for multiplication, and the expressions using the symbol "|/" at the end of some lines are not needed. Also, separate-line expressions don't need "=" on both the left and the right; just putting this symbol on the left is sufficient.

- The formatting issue has been brought up to the Ledger Journal editors. While it is possible to correct the in-line equations not displaying properly, doing so would require changing the font or otherwise changing the provided template, which would be going against the submission guidelines.

- Changed the "*" to a dot for the equations. Removed the "|/" expressions. Moved all "="s to the left side, removed them from the right sides.

Although the premise of this paper is rather interesting, I don't recommend it for publication in Ledger due to lack of attention to important details in some places and overemphasis of mathematical trivialities in others. It is immediate from the descriptions of the games in this paper what the expected winnings should be, and therefore the formal analyses here offer negligible substance.

Reviewer B:

> or c*m coins per block on average

Correction: or c/m coins per block on average.

- No, c*m is a correct value. m exists in a range between 0 and 1 indicating how big of a "market share" the miner has - 0.1 is 10%, 0.51 is 51%, etc. A miner owning 10% of the network's mining power earns on average 10% of the block reward for each block - c*m, or 0.1c. If we used c/m, they would earn 10c, or 10x the block reward per each

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

block. The confusion stemmed from the fact that the miners were expected to create one out of every 1/m blocks, which would give them c/(1/m), or c*m.

> Scenario 2 - lottery with a delayed resolution

I think this paragraph should do a better job clarifying that what the attacker is trying to do is potentially not publish the block that _decides_ the outcome of the bet, and it's that block that matters, not the block that the bet was included in.

- Clarified the paragraph to explain that the attacker is trying to create a winning block and discarding the deciding block if it does not guarantee a win. Publishing all blocks would be no different than honest gambling.

> Probability of this is m*(1-p). In this case, we discard the block and the bet, netting 0

The logic here feels a bit iffy, as the bet isn't really "discarded" as another miner will create the next block (or possibly yourself) and it will be that same bet that gets resolved. However, I suppose you could view it as being equivalent to "discarding" the bet with zero payoff and immediately publishing a new one so it shouldn't change the calculations.

- The logic here is as follows - if we mine a block that wins us the bet, we publish it. If we mine a block that loses us the bet, we discard it. We do this to avoid a guaranteed loss in hopes that the block that will eventually be generated, either by us or some other miner, will win us the bet. For all intents and purposes, the block discarded block does not happen, so the net result is 0. The outcome was put in there to account for all 4 outcomes from combining “win bet”, “lose bet” with “attacker mines the block”, “someone else mines the block”. If the attacker broadcasted all of the blocks they mine, they would just be engaged in honest gambling.

I also get the instinct that the reason why high-p gambles lead to such seemingly high security levels is actually quite boring: high-p gambles with nonzero house take (ie. h < 1) are actually a very bad proposition for a gambler to take in a certain mathematical sense. To see why, consider a p=0.96, h=0.98 gamble. You can consider it being equivalent to the sum of two of the following gambles: (i) putting $0.5 into a p=0.96, h=1 gamble, (ii) putting $0.5 into a gamble which 96% of the time just gives you your money back, and 4% of the time takes it all. The first gamble is purely fee-free, the second is purely fee, so in some sense you could view the "fee" of that gamble as _actually_ being 50% (another way of viewing this is that the maximum gains without the fee would be ~0.04, but with the fee the maximum gains are ~0.02, but the losses in both cases are the same, so the "cost" of the "potential upside" is doubled). Of course games that are skewed against the player are hard to exploit, they're so skewed you need to have really really favorable conditions in order for even an exploiter to win at them!

- Yes, it does look like games where p->h are rather skewed. For most scenarios discussed, lower p games were much easier to game with modest mining power and monetary resources.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29

- Indeed, games that are skewed against the player are hard to exploit, which is by design for casino games - having no house edge means the casino is not earning money, while having a game favouring the player creates a money faucet. The entire paper was focusing on starting at an unfavourable condition and skewing it in the player's favour with the use of mining power.

Aside from this, generally solid analysis.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2016.29