LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2016.29

RESEARCH ARTICLE

Gaming Self-Contained Provably Fair Smart Contract Casinos

Piotr J. Piasecki[*†]

Abstract. This paper discusses the game theory behind self-contained smart contract provably fair casinos, how they can be gamed by attackers with a large amount of money and computing power, as well as what are the necessary conditions to assure the system cannot be taken advantage of under various configurations.

1. Introduction

With the advent of smart contracts on Ethereum,[1,2] a number of people have considered creating self-sustained provably fair gambling contracts that derive their random seed from transaction and block hashes. [3-6] However, as with any gambling systems, one can expect someone to try gaining an unfair advantage and turning it into a money faucet. This paper explores whether an attacker with large amount of money and computing power can game the system profitably.

2. Assumptions

We will be looking at a simple "Satoshi Dice-style" game.[7] The game is a simple chance game – it is won if the number derived from a predefined formula is greater than a given threshold. Furthermore, unlike the standard "dice" game, we will be only using randomness generated from block hashes and transactions, rather than the traditional model that relies on server-side secrets.

We will not be discussing the possible 51% attacks on the network as they can skew the coin economy and are generally not a common occurrence for any healthy network; instead, we will focus on cases where the miners do not take over the network completely.

The source of randomness for the game, as assumed above, comes from transaction and block hashes. Transaction hashes combined with an additional source of randomness have been used before in similar games, such as in Satoshi Dice.[7] However, as transaction hashes can be easily brute forced before being submitted (as there is no Proof-of-Work associated with creating them), they are not enough to fully secure a bet. They can, however, be used to give each bet an individual outcome when combined with a better source of randomness – to avoid for example everyone that bet on a given result winning or losing at the same time.

Block hashes can be considered the main deciding factor for the randomness used in a given bet. Any transaction involved in setting or resolving the bet would have to be included in a block before it can take effect, thus the block will always be generated after the transaction. Blocks are also a lot harder to brute force, as each takes a lot of computing effort to create under Proof-of-Work. While not all bits of a block hash are truly random (most hashes will have a lot of

†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

leading zeroes due to the nature of mining), using a portion of the hash, or a hash of the hash would be enough for a pseudorandom number generator. The block hash will be the part of the system the attacker will be attacking with their mining power.

The smart contract used for the game is created by a casino and submitted into a cryptocurrency system such as Ethereum. Beyond creating the contract and providing it with funds, the casino does not take an active role in how the games are played out. The game is initialized by the player sending a transaction to the contract, and the contract is resolved with the transaction / block hash being used as randomness. The player in the scenarios explored is a malicious miner aiming to use their mining power to manipulate the game's random number generator and the bet's outcome to their advantage.

3. Definitions

A miner controlling a certain percentage of mining power of a given Proof-of-Work network has an m chance of mining any given block and receiving the coinbase reward for it. Since all wagers, payoffs and expected values will be expressed in relation to the coinbase reward, the reward will be expressed as 1. Thus, if a wager is 1, it is equal to one block reward. The miners expect to earn 1 every $1/m$ blocks, or m coins per block on average.

The probability of winning a game is denoted by p. It is often chosen by the player in a fairly granular fashion essentially somewhere between 0 and 1. House edge, h, determines the percentage of winnings given to the player. It is usually fixed, close to, but below 1. The player can wager w to play a game. The value of w is in most cases pretty flexible, although a casino can impose lower and upper bounds on it depending on the cost of processing a game and the bank. For our discussions, we will assume it is a positive number without bounds.

The expected value of a game, EV, illustrates how much a player is expected to get out of playing a single game on average. In a fair casino game, $EV = (h-1)w$, the player is expected to lose a small amount of money each game that the casino earns. The individual outcome of a game – payoff win or payoff loss – will be denoted as P_{win} or P_{loss}, respectively.

4. Expected Values of Games

If a game is lost, we expect to lose our wager: $P_{loss} = -w$. Therefore, if a game is won, we expect to earn:

$$EV = (1-p)P_{loss} + pP_{win}$$

$$(h-1)w = (1-p)(-w) + pP_{win}$$

$$wh - w = -w + pw + pP_{win}$$

$$P_{win} = \frac{wh}{p} - w$$

Moreover, we expect to earn money with each win, so,

$$P_{win} > 0$$

$$\frac{wh}{p} - w > 0$$

$$\frac{h}{p} - 1 > 0$$

which leads us to Eq. (1) that imposes a new limit on our domain.

$$h > p \tag{1}$$

5. Scenario 1 – Lottery with an Instant Resolution

The first scenario we will be looking at is a smart contract lottery that resolves in the same block it is played. If the random seed used to determine whether any given transaction is successful or not is a combination of both the transaction hash and the block hash, it can be simplified to essentially being determined only by the block hash if we're looking at only one transaction and we're focused on gaming the system. A losing transaction hash and a block hash cannot turn into a winning combination by changing the transaction hash as that would invalidate the block hash. However, changing the block hash can create a valid combination even for the same transaction hash. An honest miner that does not gamble is expected to earn m per each block.

A dishonest miner would first wager w coins in a transaction they would not transmit yet to create a gambling transaction. Then, they would proceed to create the block with a probability m. Once the block is created, they will only transmit it if they win the game, thus a block will be valid in p cases. If the block hash does result in a won game, the miner will earn their coinbase reward c, plus the winning payoff P_{win}. Thus, the outcomes of the game are:

- We mine the block and win the bet. Probability of that is $m \cdot p$, and we will earn $1 + P_{win}$.
- We mine the block, but lose the bet. Probability of that is $m \cdot (1 - p)$. In this case, we discard the block and the bet, netting 0.
- We don't mine the block. Probability of that is $(1 - m)$. In this case, the bet is not broadcast, thus we net 0 again.

All in all, they are expected to earn:

$$\begin{aligned} EV &= mp(P_{win} + 1) + m(1-p) \cdot 0 + (1-m) \cdot 0 \\ &= mp(P_{win} + 1) \\ &= mp\left(\frac{wh}{p} - w + 1\right) \\ &= m(wh - wp + p) \end{aligned}$$

We are earning more than through mining when:

$$m(wh - wp + p) > m$$
$$wh - wp + p > 1$$
$$w(h - p) > 1 - p$$
$$w > \frac{1 - p}{h - p} \tag{2}$$

When $w = (1 - p)/(h - p)$ we are earning as much being dishonest as mining honestly. For greater values of w, we are earning more. To find the smallest value of w, let us assume:

$$x = \frac{1 - p}{h - p}$$

Therefore the derivative:

$$\frac{dx}{dp} = \frac{(1 - h)}{(h - p)^2}$$

Since $1 - h > 0$ the derivative is positive and the value of x is increasing as p increases. Taking $p \to 0, x \to x_{\min}$, which is $1/h$, meaning that for all values of p,

$$w > \frac{1}{h} \tag{3}$$

which is our absolute lower bound.

6. Scenario 2 – Lottery with a Delayed Resolution

A way to improve the smart contract lottery would be to put a delay between the gambling transaction being sent off and the bet being resolved. This would force the potential attacker to broadcast every transaction they attempt to game and expose them to a risk of losing their wagered amount if they will not be the one creating a block. On the flip side, there is also a chance of getting some of the wager back in blocks that aren't created by the malicious miner by sheer luck. As such, the gaming algorithm would essentially boil down to submitting one wager per block in hopes of gaming them. The attacker would then aim to create a block that wins them the wager – publishing all blocks indiscriminately from them would be no different than honest gambling.

Thus, the game is as follows:

- We mine the block and win the bet. Probability is $m \cdot p$, and we will earn $1 + P_{win}$.
- We mine the block, but lose the bet. Probability of this is $m(1 - p)$. In this case, we discard the deciding block, netting 0, as we are not interested in publishing losing blocks. We can ignore this outcome.
- We do not mine the block but we still win the bet. Probability of this is $(1 - m)p$. In this case, we gain P_{win}.
- We do not mine the block and we lose the bet. Probability of this is $(1 - m)(1 - p)$. In this case, our balance changes by P_{loss}.

$$\begin{aligned} EV &= mp(P_{win}+1)+(1-m)(1-p)(P_{loss})+(1-m)p\cdot P_{win} \\ &= mp\cdot P_{win}+mp+(1-m-p+mp)(P_{loss})+p\cdot P_{win}-mp\cdot P_{win} \\ &= mp+(1-m-p+mp)(P_{loss})+p\cdot P_{win} \\ &= p\cdot P_{win}+mp+(1-m-p+mp)(P_{loss}) \\ &= p\left(\frac{wh}{p}-w\right)+mp+(1-m-p+mp)(-w) \\ &= wh-wp+mp+w(m+p-1-mp) \\ &= w(h-p+m+p-1-mp)+mp \\ &= w\big(h-1+m(1-p)\big)+mp \end{aligned}$$

We are earning more than while mining when:

$$EV > m$$
$$w\big(h-1+m(1-p)\big)+mp > m$$
$$w\big(h-1+m(1-p)\big) > m(1-p)$$
$$w > \frac{m(1-p)}{\big(h-1+m(1-p)\big)}, \text{for } m > \frac{1-h}{1-p} \tag{4}$$
$$w < \frac{m(1-p)}{\big(h-1+m(1-p)\big)}, \text{for } m < \frac{1-h}{1-p} \tag{5}$$

For $\big(h-1+m(1-p)\big)=0$,

$$0 > m(1-p)$$
$$mp > m$$

which is never true.

For the second case, Eq. (5), we are looking for w such that:

$$0 < w < \frac{m(1-p)}{\big(h-1+m(1-p)\big)}$$

Therefore:

$$0 < \frac{m(1-p)}{\big(h-1+m(1-p)\big)}$$

Since in this scenario $\big(h-1+m(1-p)\big)<0$, meaning the denominator is negative. In order to get a positive result, the numerator would also have to be negative, therefore $m(1-p)<0$. Because $p<1$ and $m\geq 0$, there is no solution.

For the first case (4), assuming h=0.98, we get:

$$w > \frac{m-mp}{\big(-0.02+m(1-p)\big)}, \text{for } m > \frac{0.02}{1-p}$$

The graph of $w(m,p)$ is shown in Fig. 1.

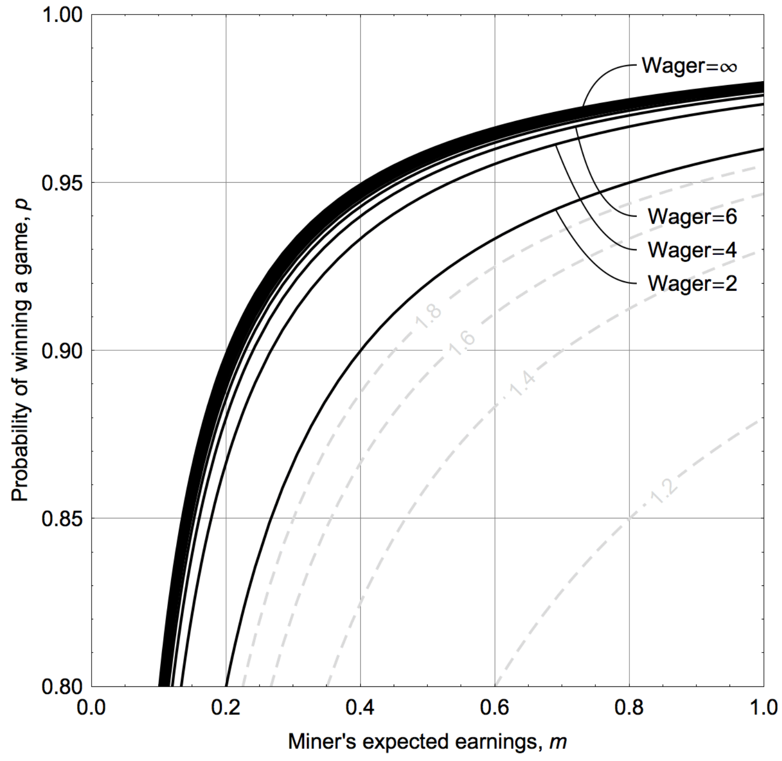

Fig. 1. Minimum wager w required to profitably game the "lottery with a delayed resolution" contract. $h = 0.98$, logarithmic scale for m and p. Black curves at different constant values of w are shown.

Note that in Fig. 1 there is a boundary in the top-left corner for low values of m and high values of p. Even assuming $m = 1$, we get:

$$w > \frac{1-p}{h-1+1-p}$$

$$w > \frac{1-p}{h-p}$$

Meaning our absolute lower bound for w is the same as in Scenario 1, Eq. (3): $w = 1/h$.

7. Scenario 3 – Mining Multiple Blocks to Game Delayed Lottery

A determined attacker might still wish to game the delayed lottery from Scenario 2 by attempting to create a fork in the chain long enough to encapsulate both the betting transaction and the resolution block. They would only submit the chain if it was favourable to them, thus mitigating the constant loss of bets when they are unable to win.

We will denote the probability of the attacker mining n blocks before the entire network mines n blocks together as M. This probability, M, is equal to:[8]

$$M = \sum_{k=n}^{2n-1} \binom{2n-1}{k} \cdot m^k \cdot (1-m)^{2n-1-k}$$

The outcomes of the game for n blocks are as follows:

- We mine the n blocks and win the bet. Probability of that is $M \cdot p$, and we will earn $n + P_{win}$.
- We mine the n blocks, but lose the bet. Probability of this is $M(1-p)$. In this case, we discard the block and the bet, netting 0. We can ignore this outcome.
- We don't mine the block. Probability of this is $(1-M)$. In this case, the bet is not broadcast and nothing happens. We can ignore this outcome.

For a chain of length n, the earnings are as follows:

$$\begin{aligned} EV &= Mp(P_{win} + n) \\ &= Mp\left(\frac{wh}{p} - w + n\right) \\ &= M(wh - wp + np) \end{aligned}$$

It is more profitable to cheat when:

$$\begin{aligned} M(wh - wp + np) &> mn \\ wh - wp + np &> \frac{mn}{M} \\ wh - wp &> \frac{mn}{M} - np \\ w(h-p) &> \frac{mn}{M} - np \end{aligned}$$

Since $h > p$:

$$w > \frac{\frac{mn}{M} - np}{(h-p)}$$

For $n = 1$:

$$M = \sum_{k=1}^{1} \binom{1}{k} \cdot m^k (1-m)^{1-k} = m$$

$$w > \frac{1-p}{h-p}$$

which is identical as in Scenario 1, Eq. (2).

For $n = 2$:

$$\begin{aligned} M &= \sum_{k=2}^{3} \binom{3}{k} \cdot m^k \cdot (1-m)^{3-k} \\ &= \binom{3}{2} m^2 \cdot (1-m)^1 + \binom{3}{3} m^3 \cdot (1-m)^0 \\ &= 3m^2 \cdot (1-m) + m^3 \\ &= 3m^2 - 2m^3 \end{aligned}$$

Leading to the following inequality equation for w:

$$w > \frac{\frac{2m}{3m^2 - 2m^3} - 2p}{h - p}$$

The values of w get very high when $m \to 0$, and $p \to h$. For example, the value of w gets as high as 13,032 when $p = 0.975$ and $m = 0.01$, as shown in Fig. 2, below. However, if we take a more reasonable set of parameters, $p = 0.55$ and $m = 0.03$, we get $w \approx 50$, which is a considerable wager, but not unheard of.[9]

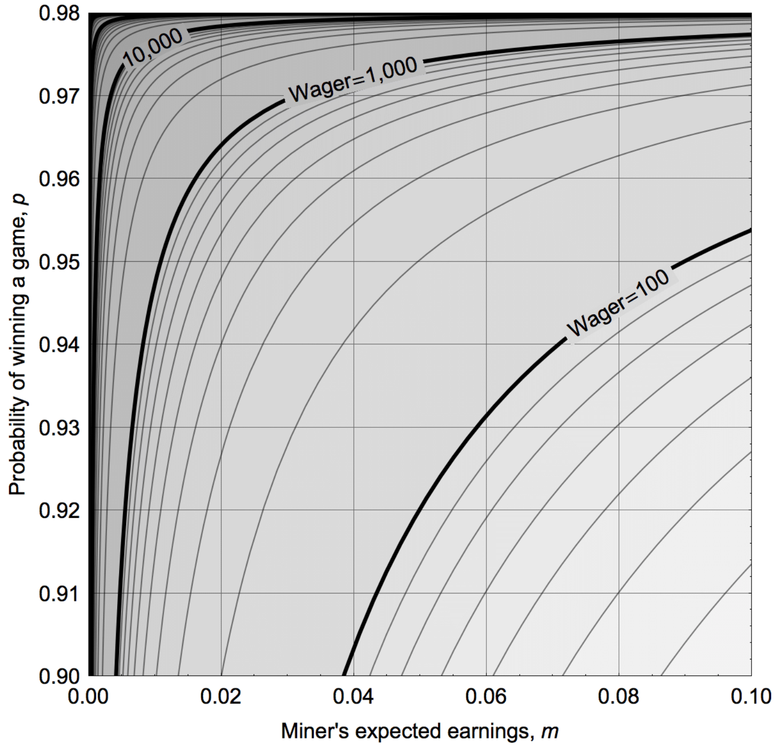

Fig. 2. Minimum wager w required to profitably game the "delayed lottery" contract. Here the values used are $h = 0.98$, $n = 2$, $0.9 < p < 0.975$ with 0.005 steps, and $0.01 < m < 0.1$ with 0.01 steps. Black curves at constant values of w are shown.

For $n = 3$, the function gets steeper:

$$\begin{aligned} M &= \sum_{k=3}^{5} \binom{5}{k} \cdot m^k \cdot (1-m)^{5-k} \\ &= \binom{5}{3} \cdot m^3 \cdot (1-m)^2 + \binom{5}{4} \cdot m^4 \cdot (1-m)^1 + \binom{5}{5} \cdot m^5 \cdot (1-m)^0 \\ &= 10m^3 \cdot (1-m)^2 + 5m^4 \cdot (1-m)^1 + m^5 \\ &= 10m^3 \cdot (1 - 2m + m^2) + 5m^4 - 5m^5 + m^5 \\ &= 10m^3 - 20m^4 + 10m^5 + 5m^4 - 4m^5 \\ &= 6m^5 - 15m^4 + 10m^3 \end{aligned}$$

Leading to the following inequality equation for w:

$$w > \frac{2}{h-p}\left(\frac{1}{6m^4 - 15m^3 + 10m^2} - p\right)$$

For the same parameters $p = 0.975$ and $m = 0.01$ we considered for $n = 2$, we get $w = 405{,}676$ (see Fig. 3). To get a wager under 50 for $p = 0.55$, we would need m greater than 0.1.

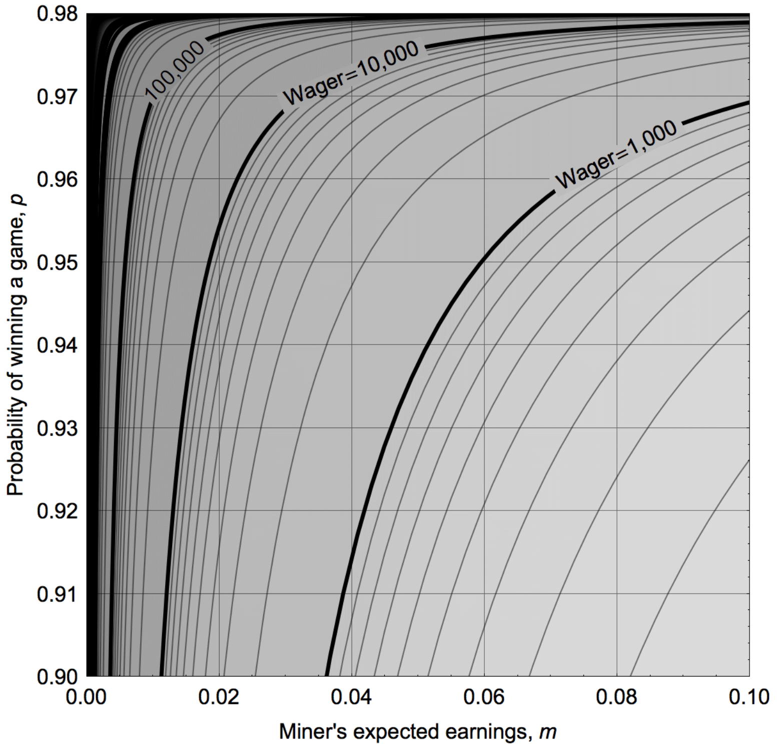

Fig. 3. Minimum wager w required to profitably game the "delayed lottery" contract. Here we used the parameters, $h = 0.98, n = 3, 0.9 < p < 0.975$ in 0.005 steps, and $0.01 < m < 0.1$ in 0.01 steps. Black curves at constant values of w are shown.

For higher values of n, it takes significantly more money or computing power to be profitable.

8. Scenario 4 – Proof-of-Stake Blockchain

In Proof-of-Stake,[10-12] it is significantly easier to create a valid block with a desirable hash since one only requires a valid signature, not a sufficiently low value of hash.[13] Assuming an attacker can brute force through blocks on the level of GPU Bitcoin block hashing,[14] we can expect to cycle through 865 million combinations using a single 6990 ATI graphics card per second. This means being able to produce a winning hash for a bet of $1{:}2^{29}$ every second. Clearly, relying on a combination of transaction and block hashes is not a useful random number generator in a Proof-of-Stake scenario.

9. Scenario 5 – Irrational Actors

The previous scenarios assumed actors trying to attack the game were rational - they would only play or cheat the game if it would result in a positive net gain for them. However, a smart contract can also face actors that are not motivated by their own profit, but rather by the loss of money from the contract, no matter how much it would cost them. They could take a form of competing casinos or other malicious entities that are motivated by the casino running out of money or going out of business. Their motivation is not to maximize their earnings, but to maximize the casino's losses.

In this scenario, we ignore the mining reward completely (the irrational actors don't care about mining rewards or losses). All other probabilities and earnings are the same as in their respective scenarios.

Our decision whether or not to play from Scenario 1 now looks like:

$$\begin{aligned} EV &= mp(P_{win}) \\ &= mp\left(\frac{wh}{p} - w\right) \\ &= m(wh - wp) \\ EV &> 0 \\ m(wh - wp) &> 0 \\ wh - wp &> 0 \\ h - p &> 0 \\ h &> p \end{aligned}$$

Which is one of our initial conditions, Eq. (1). This means the irrational actor will always play the game, no matter how small the maximum wager is set to.

From Scenario 2:

$$\begin{aligned} EV &= mp \cdot P_{win} + (1-m)(1-p)(P_{loss}) + (1-m)p \cdot P_{win} \\ &= mp \cdot P_{win} + (1 - m - p + mp)(P_{loss}) + p \cdot P_{win} - mp \cdot P_{win} \\ &= (1 - m - p + mp)(P_{loss}) + p \cdot P_{win} \\ &= p \cdot P_{win} + (1 - m - p + mp)(P_{loss}) \\ &= p\left(\frac{wh}{p} - w\right) + (1 - m - p + mp)(-w) \\ &= wh - wp + w(m + p - 1 - mp) \\ &= w(h - p + m + p - 1 - mp) \\ &= w\big(h - 1 + m(1-p)\big) \end{aligned}$$

In this case, EV is positive when:

$$\begin{aligned} w\big(h - 1 + m(1-p)\big) &> 0 \\ h - 1 + m(1-p) &> 0 \end{aligned}$$

$$m(1-p) > 1-h$$
$$m > \frac{1-h}{1-p}$$

which is the same as our boundary condition from the Scenario, see Eq. (4).

From Scenario 3:

$$\begin{aligned} EV &= m^n \cdot p(P_{win}) \\ &= m^n \cdot p\left(\frac{wh}{p} - w\right) \\ &= m^n \cdot (wh - wp) \end{aligned}$$

In this case, EV is positive when:

$$\begin{aligned} m^n(wh - wp) &> 0 \\ wh - wp &> 0 \\ h - p &> 0 \\ h &> p \end{aligned}$$

Which is one of our initial conditions, Eq. (1), and the same result as when analysing Scenario 1. The only other boundary condition in this scenario can be the theoretical limit of how many blocks can be overwritten in a chain.[15]

10. Conclusion

Using a smart contract on a Proof-of-Work blockchain is sufficient to secure any dice-like game with a maximum wager of $w \leq 1/h$ (see Eq. (3)) against any attack from a rational attacker with any amount of mining power. This even holds if the wagers are resolved by the same block they are created. The maximum wager can be increased further if there is at least one more block in between the wager and the resolution and one assumes some maximum mining power any single entity or a mining cartel can amass in order to try to cheat.

For chains using Proof-of-Stake, naively using the block hash as a random seed opens one up completely for an easy attack from any entity creating blocks at virtually no cost to them.

In case of an attacker that is only motivated by the loss of money from a contract no matter the cost, there is no way of stopping that entity provided any length of chain of blocks can be overwritten. In real life, the attack would be limited by the length of the chain one can overwrite.

Acknowledgement

The author of this paper would like to acknowledge Dr. Christopher E. Wilmer of University of Pittsburgh for his feedback on formatting, charts and labelling of various parts of this paper.

Notes and References

[1] No Author. "Smart contract" *Wikipedia* (accessed 30 November 2015) `https://en.wikipedia.org/wiki/Smart_contract`

[2] *Ethereum Frontier* (accessed 30 November 2015) `https://www.ethereum.org/`

[3] No Author. "Provably Fair" *Wikipedia* (accessed 30 November 2015) `https://en.wikipedia.org/wiki/Provably_fair`

4 *Etherapps blockchain explorer* (accessed 30 November 2015) `https://explorer.etherapps.info/dice`

5 Wood, G. "Gavmble" *GitHub* (accessed 30 November 2015) `https://github.com/ethereum/dapp-bin/blob/master/gavmble/gavmble.sol`

6 Pseudonymous (/u/smartcontractor) "Rolldice.io - The first Ethereum powered dice site!" *Reddit* (accessed 30 November 2015) `https://www.reddit.com/r/ethereum/comments/3hzf47/rolldiceio_the_first_ethereum_powered_dice_site/`

7 *Satoshi Dice* (accessed 30 November 2015) `https://www.satoshidice.com/`

8 Nicolas, A. Comment in "Probability of an event occuring n times before its complement occurs m times" *Mathemarics StackExchange* (accessed 1 May 2016) `http://math.stackexchange.com/a/369992/20958`

9 Pseudonymous (/u/Rannasha) "High-roller loses $800K with all-in bet on Just-Dice" *Reddit* (accessed 30 November 2015) `https://www.reddit.com/r/Bitcoin/comments/1npmaq/highroller_loses_800k_with_allin_bet_on_justdice/`

10 King, S., Nadal S. "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake" No Publisher (19 August, 2012) `http://www.peercoin.net/assets/paper/peercoin-paper.pdf`

11 Harald, G. (Dr. Haribo) "What is proof-of-stake?" *Bitcoin StackExchange* (accessed 30 November 2015) `https://bitcoin.stackexchange.com/questions/9082/what-is-proof-of-stake`

12 Poelstra, A. "On Stake and Consensus" No Publisher (accessed 22 March) `https://download.wpsoftware.net/bitcoin/pos.pdf`

13 Piasecki, P. J. (ThePiachu) "In Proof-of-Stake, can a block creator fudge the block hash?" *Bitcoin StackExchange* (accessed 30 November 2015) `http://bitcoin.stackexchange.com/q/40848/323`

14 "Non-specialized hardware comparison" *Bitcoin Wiki* (accessed 30 November 2015) `https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison`

15 Rosenfeld, M. "Analysis of hashrate-based double-spending" No Publisher (13 December 2012) `https://bitcoil.co.il/Doublespend.pdf`