Dao.Casino

Blockchain Protocol for Gambling

Konstantin Lomashuk, Ilya Tarutov

Draft. Last edited by Ksenya June the 24th 2017

https://dao.casino

Abstract. Online gambling takes 10% of total legal turnover in the world gambling business [1], while the trusted third party - online casinos - remain a black box for players and the market remains hard to enter for the game developers.

We propose a decentralised public system for gambling industry - DAO.Casino. DAO.Casino consists of a) an automated value distribution protocol acting as an mechanism of incentives and expressed in a system of Ethereum contracts b) a system that is capable of providing equally unpredictable pseudorandom numbers for the games.

We show how applying cryptoeconomics and implementing such system on Ethereum can solve some common issues in traditional online gambling industry through removing the need of trust (i.e. automating crucial elements of online gambling industry that usually require trust reduce a risk of fraud and enable a new more sustainable business model to evolve), and also can minimise overheads for casino

operators which allows higher payouts for the players and game developers to monetise their work. Auditability and cryptographic verifiability of Ethereum powered software potentially makes certification process for gambling games easier.

This document discusses design and implementation of such system from social and technological perspective on the abstract level: participant’s roles and in their interactions, random number generation for deterministic virtual machine, economic incentive mechanisms for the participants, and the system components that express the logic of value transfer. We describe an MVP that will be released in Summer 2017 and further development plans. Technical specifications for reward distribution system and bankroll backing system will be published separately during spring and summer 2017.

1. Introduction

DAO.casino is a protocol defining interactions between untrusted participants in the context of online gambling industry. This includes two levels: game level - player’s trust in a particular casino operator in the context of the game (provably fair gambling) and a business model level: i.e. game developer doesn’t have to trust casino operator to get rewarded. In short: all participants that are needed for the system to function don’t have to trust each other in order to cooperate.

Trusted third party that is needed in a traditional online gambling industry to function is replaced with smart contracts that act as autonomous agents that automatically reward all the key contributors:

game developers, referrers and operators of independent platforms necessary for the game discovery, and those that take part in PRNG. The fact that reward system is fully automated and transparent allows to introduce a crowdfunding element into a bankroll of every game (see section 2.1.5) and incentivise community driven security audit.

Obtaining randomness necessary for gambling games in a deterministic virtual machine is not a trivial task, so that in addition to technically obtaining pseudorandom values an economic incentive layer should be introduced. Equally unpredictable random numbers that determine outcomes in each game are provided by economically incentivised participants interacting with the PRNG contract. PRNG method implemented in DAO.Casino MVP is described in a section 3.

This system can provide a P2P marketplace for game developers and support a large number of independent front-end platforms where players can discover and play gambling games fair by design.

In the beginning we can expect existing licensed online casino operators can integrate with the protocol to reduce costs, while long term new forms of regulations more suitable for decentralised, transparent and automated systems can emerge. Long term goal of DAO.Casino project, apart from providing experimental software and implementing main protocol components, is to contribute to a development of new forms of online gambling certification. So far gambling games on Ethereum remain in the gray legal area, but it doesn’t have to be so, since technologies such as Ethereum open up possibilities for far better customer protection and enable fair by design gambling.

BET - DAO.Casino internal token - is used as point system for incentivisation as well as a sub-currency which is used in the games. See section 2.2. BET token is a standard Ethereum ERC20 token.

DAO.Casino protocol MVP will be released in Summer 2017, consequent releases (see section 2.3)

1.1 Gambling games

Legal turnover in the world gambling business reached 50bn in 2017 with projected turnover of 56bn [1]. 60% of online casinos belong to 22 leading networks. Another 30% are subsidiaries of well-known offline casinos, and the remaining 10% is owned by private individuals. Taking into account these monopoly phenomena, the developer has few chances to attract the required number of audience members to start its project in this market.

Online gambling games cause distrust on the part of the players in most cases, however up until the development of technology providing a platform for automation of trust, such as Ethereum, there wasn’t a possibility to provide any alternative.

1.2. Common Issues in Online Gambling Industry

We can summarise existing issues in the online gambling industry as originating from the concentration of trust, in all the aspects of gambling industry from RNG to user balance account management, to developers access to the market as described in the introduction.

Common problems experienced by the players

- After transferring the money to the game account, it is not credited or it is stolen

- After withdrawing the money from the deposit, it has not been credited to the card

- The player has not received the promised bonuses

- The player is not able to enter his game account

- Hidden fees: casinos charge a fee for the gain withdrawal

- The player can withdraw funds only on a certain day

Some of the existing issues in the market of online gambling** that DAO.Casino protocol can solve**

- Risk of fraud on behalf of online casinos

- Inability to check the result of the draw

- High and hidden fees
- High entry level for the game developers
- High costs of running an online casino
- Operations overheads such as integrating payment systems & user account balance management

2. DAO.Casino Protocol

The objective of Dao.Casino protocol is to enable a sustainable model which benefits all the parties involved in the online gambling business process. (i.e it should be more profitable for the developer to use Dao.Casino protocol, rather than work on its own, and less costly for a casino operator to provide better service for the players. Players should have access to more diverse range of games from independent game developers while having a higher level of security than in traditional online casinos).

Single points of failure - processes of value transfer where trusted third party would be required in a conventional online gambling business is replaced by code consistently executed by Ethereum network - a system of smart contracts. These contracts are, simply speaking, just escrows that can be triggered by particular actions performed by the participants and nothing else. These actions correspond to the value that the participants add to the ecosystem.

When there is no human actors with administrator permissions that can change value distribution processes there is no risk of such actors would become corrupt and make changes in their favour. This consistency of code is a pretty useful feature in the context of gambling business.

Security of the system is achieved through transparency, consistency and cryptographic verifiability of software used to automate processes otherwise involving trust, and economic incentive mechanisms for it’s human participants.

DAO.Casino protocol still requires a fee system, but the distribution of fees is a hard coded reward system for all the participants described further. There is no hidden fees.

Suggested Dao.Casino reward distribution distributes tokens accumulated by game contracts as follows, rewarding all participants equally, however independent platform operators can choose their own reward distribution scheme (i.e. if the platform needs to incentivise more referrals but build their own games)

- Game Developer - 25%
- Casino Operator * - 25%
- Referrer - 25%

- Bankroll backers - 25%

*Casino operator / platform and a Referrer can be one entity

Despite of Gas cost that is required for Ethereum powered software to run, the fees should remain lower than in traditional server based online gambling. This is because the gas is paid per operation, and only for the operations and storage that is used.

Game’s front end can be stored either on the server or on the network using decentralised file storage systems such as IPFS or Ethereum’s native Swarm. Using Swarm will also allow to program the game to pay for it’s own front-end storage.

Dao.Casino Protocol Goals

Remove the need in a trusted party in all aspects of online gambling industry, therefore:

- Reduce operational costs of online gambling therefore provide higher payouts

- Reduce a risk of fraud

- Remove the need and responsibility from online casino operators to maintain player’s account balances

- Enable game developers to monetise their work while maintaining their IP
- Enable game developers to access bankroll for their game without extra responsibility for managing it
- Enable an open ecosystem of provably fair interoperable online casinos

Integrate a system of replicable templates and incentivised audit to allow game developers not familiar with Solidity to benefit from a new value transfer paradigm that Ethereum offers.

2.1. Roles

System Participants. During initial design we identified a number of roles that are needed for the system to develop and to function. Some are common roles from the traditional online gambling industry i.e. casino operators, referral, contracted game developers, other roles (bankroll backers, random providers) have been added to the decentralised architecture.

1. Game developers
2. Platform Operators
3. Referrers

4. Random Number Providers

5. Bankroll Backers

6. Players

7. Autonomous Agents (contracts without superusers)

2.1.1. Developers

Developers refers to both: game developers and contract developers. Whoever provides a functional piece of software automatically receives tokens to their EOA by the DAO.Casino system proportionally to the usage of this software. Independent game developers should be able to collaborate with platform operators easier, retain their IP rights in the game and a possibility to receive lifelong rewards for their work automatically.

A front-end developer who is capable of creating a good game doesn’t have to developer a gambling game smart contract, but use existing ones. In this case automatic developer’s rewards will be split between the creator of the contract and the creator of the front-end.

For the past 1.5 years we have seen growth of gaming and gambling on Ethereum. However we cannot expect that every good game creator will learn how to program for EVM, while there is a lot of talent amongst independent game devs. To leverage existing game design and development skills tested smart contract templates can be used by several game dev teams. Ethereum ecosystem allows teams to share backend without compromising security.

2.1.2. Platform Operators

DAO.Casino system can accommodate multiple front-end platforms customised for different user groups and regions. They remain independent while using the same governance and value exchange protocol. The platform layer of the system is therefore federated. Platforms receive tokens to their EOA accounts proportionate to the usage. If the platform attracts new players to the game it becomes a Referrer and receives Referrer’s reward (see 2.1.3)

Front-end platforms built on top of DAO.Casino system do not have to have additional account balance systems, since the balances and the balance history is stored in Ethereum blockchain.

Existing online casino operators can create front-end platforms to interface with the token system. We can estimate that as the system evolves, independent game developers teams will be able to form groups and become casino operators with appropriate authorisation.

In a federated model casino operators still provide value to their customers by customising platforms for a particular user group, creating their own ranking and recommendation systems. It is up for the operator to conduct KYC procedures required in their jurisdiction and securely store sensitive user data in compliance with personal data protection laws. User accounts in the whole system are EOAs the only data which is recorded on the blockchain is their balances and tx history.

Front-end platforms are, as in the traditional online gambling industry, are used for game discovery, rating and as a human interface. However, instead of the server based backend, they run on Ethereum.

They might also integrate user wallets to store small amounts of BET. It is recommended not to store larger amounts of BET in browser based platforms.

**Initial Test Platform. **At least one platform is needed for the system to function. DAO.Casino core team is implementing a test frontend platform with a number of games that will be integrated with DAO.Casino system on Ethereum testnet. See section 4.

2.1.3. Referrers

The Referrer is a member of the affiliate program who brought a referral (a new player). The Referral is a participant of the affiliate program who signed up under the recommendation of another participant.

If you decentralise all the things how do you know where things are? In a decentralised system discovery can be facilitated by economically incentivized participants. Affiliate programs are commonly used for promotion of products and services. The only difference in DAO.Casino system is that referrers, same as all other active participants, are rewarded automatically, and can be sure that they will get rewarded. Another difference is that the rewards depend on whether the referral in question is an active player, to make sure that the affiliate system is sybil proof and the game creator doesn't share the funds generated by the game with the referrer for nothing.

Referrers receive a percentage of the tokens generated by the game they help promoting while their referral is actively using it. If the referrer is absent, and the player came to the platform on its own through the platform's own channels, then the platform itself is considered the referrer, which means that it receives the reward. A player willing to top up their BET balance can become a referrer. It applies to other participants.

2.1.4. Random Number Providers *

*potentially viable way to provide equally unpredictable pseudorandomnes, not included in the first pre-token launch release.

Random Number providers provide values to the contract that is then generates a value that determines and outcome of the game and receive tokens in exchange.

In the implemented of PRNG contract (see section 3) for two parties games Random Provider and Bankroll Backer are the same entity.

Future versions may include a hybrid system of economically incentivised oracles:

Participants in incentivised oracles system: in addition to the pseudorandom algorithm, an economic algorithm is needed to ensure that the Random Number Providers cannot exploit their partial access to the data which is used in PRNG so that the outcomes of the games remain equally unpredictable for all parties. Random number provider’s system is a hybrid between generating pseudorandomness on EVM and using authenticated data feeds (oracles). Both methods and their weaknesses are described in a section 3.4.

To participate as a Random Number Provider a participant is required to a) lock their token in the contract b) send data to the RNG contract from which the PRNG will be generated. Intentional or unintentional , malicious behaviour i.e. an attempt to predict the outcome of the game will result in the locked tokens to be distributed as rewards to other participants. Because of this staking system Random Number Providers are somehow similar to miners in a PoS system.

2.1.5. Bankroll Backers

Any token holder can support any particular game by taking on a role of a bankroll backer. Bankroll Backers locks their token in a game contract of their choice and automatically receive a reward.

Bankroll Backer behaviour can vary depending on the game contract and what PRNG system is used. In a first implementation Bankroll

Backer provides BET tokens to the bankroll of the game and a value from which the the outcome of the game is derived.

2.1.6. Players

Players discover games through the platforms and use their tokens as an ingame currency. Player can obtain tokens by becoming an active participant such as a Random Provider or a Referrer or support a crowdfunding campaign at launch. A casino operator can also choose to exchange the tokens to and from common cryptocurrencies.

For the past year we see more and more gambling games projects built for Ethereum. We can expect that early adopters will be Ethereum and decentralisation enthusiasts. Game contracts are accessible through Ethereum clients, but in the ideal situation a player doesn’t need to know anything about Ether or the technology itself to be able to play and discover games.

2.1.7. Autonomous Agents

Autonomous agents are smart contracts that run on Ethereum that perform tasks that usually would require trust.

We refer to those entities as agents, because a) their role is as important as any human actors in the system (with the only difference is that they can’t cheat and don’t need to be incentivised). b) once designed and deployed they do not have a human owner capable of withdrawing the tokens from the balances of contract accounts or control the contract in other ways. Instead a contract will “decide” whether to reward an EOA in question, according to it’s rules. Rules cannot be changed by a single party. If the contract in question is proved faulty and the rules do not satisfy the needs of human participants, it will be simply abandoned by the majority of human participants or by all of them. In a similar way an entire blockchain can be abandoned by miners.

All contracts that are considered a part of the protocol should be autonomous agents - controlled by fair and authorised participants behaviour. Contracts that are deployed for crowdfunding and the issuance of BET are not considered a part of the protocol, but a step to design, deploy and improve it, and in this sense are not autonomous - managed by a legal entity.

For more information on how BET will be distributed see Crowdfunding & Token Distribution

2.2. Token System

DAO.Casino internal token called BET is a ERC20 token. It is used as ingame currency for all the game contracts integrated with the protocol and to power DAO.Casino reward system. Ingame currency and a reward system are complimentary. We can assume that a reward system that allows people to collect tokens that can be used in games provides a better incentive mechanism than a purely reputational points system.

Overtime BET accounts history can be used as a reputation system and for ranking the games and the players. For example most popular games would have more transaction history.

Keys to the BET can be stored in platform wallets by players and in any Ethereum client or by more advanced Ethereum users, or a paper wallet. It is not recommended to store a large amount of BET in a browser, even though as a subcurrency it might be less prone to theft than widespread and more well known crypto-token such as Ether.

Ingame currency

In a traditional gaming and online gambling industry large gaming companies prefer not to have an ingame token that works across the games because it disrupts their business model. However, when it comes down to smaller independent game producers, having interoperable token makes sense, since it eliminates the need to maintain user account balances. Interoperability between games also what makes the token useful for the players, without it having inherent value.

Using Ethereum based subcurrency instead of ether reduces the risk of attacks of rational kind on the system after it scales. If the system is compromised by a malicious actor, the token will become worthless, so attacking the system wouldn’t be a rational thing to do.

Reward system

As described above all the participants contributing to the ecosystem in one form or another are automatically rewarded with BET to their accounts.

2.3. Core components and releases

2.3.1. MVP: Components of the system (to be released by the time of the token launch)

Contracts:

1. Random Contract 1.0 implementation of Signidice algorithm, integrated with game contracts below

2. Game contract Dice integrated with ref system module

3. Game contract Blackjack integrated with ref system module

4. Game contract HackDAO integrated with ref system module

5. Reward Distribution contract 1.0 integrated with the games above to distribute BET to: referrals, devs, platform operators.

6. Registry of Referrals contract that calls Referral contract

7. ERC20 integrated with 3 games

Other

1. Test Frontend Platform (Ethereum testnet only)

2. Front end for 3 games.

3. Legal infrastructure conducting further R&D

Components listed above allow minimum functionality of the system. However, a fully decentralised system requires additional components and other implementations of PRNG suitable for multiplayer games.

To conduct the crowdfunding campaign a standard Crowdfunding contract will be used.

2.3.2. Components released in 2017 after the token launch

After the BET is launched we will be able to test a running system with more participants. A percentage of the funds raised during the crowdfunding campaign will be used to implement, test and release following components:

1. Reward distribution contract 2.0 - added Bankroll Backers reward

2. Games Certification Contract example - a registry for audited game contract addresses with a reference to IPFS hashes and it’s contract in Solidity

3. DAO.Casino Browser 1.0 - a standalone desktop client that addressed Games Certification Contract and assembles the games taking their Ethereum addresses and the frontend stored in IPFS. Detailed specs of the browser will be released during the summer 2017.

In this release a bankroll crowsourcing aspect is completed. DAO.Casino protocol is public, and anyone can audit game contracts, but we can’t expect that every player will read the source code. For that reason a system for whitelisting audited games is useful. Platform operators can use their own Game Certification Contract for the games that they have audited. In this period research and testing of possibly better PRNG solutions continues.

2.3.3. Further developments

First releases of DAO.Casino are focusing on Ethereum powered components and a client - a browser that allows to navigate games without centrally hosted UI. After those components are implemented and tested the focus of the project should shift towards onboarding new developers and making the system more appealing to traditional casino operators and more useable. Working plan for the end of 2017 beginning of 2018 and onwards:

- Contracts Opensource Licence based on DAOFactory. Smart contracts as any software needs a licence to define the rules of it’s usage, distribution and relations to patented work. A unique thing about contracts, that it they can be their own licence and the licence is also a piece of code. In the context of gambling games integrated with reward mechanism we are planning to implement a reward based licensing. It means that anyone can replicate anyone’s contract within Ethereum to a new address, while the original contract author is still entitled to automatic rewards defined in the licence.

- Game Factory - a system for secure replication of game contracts based on DAO.Factory

- Host Global Game Jam locations end of January 2018. Global Game Jam is the biggest event for independent game developers. Our plan is to release a number of contract templates and documentation allowing game developers without prior knowledge of Ethereum to create games.

- Integrate a test VR application with DAO.Casino logic. Traditional online casinos are moving into VR, and if we if want to bring decentralised fair gambling to the masses, not just decentralisation enthusiasts, we will need to release VR/3D application integrated with DAO.Casino protocol to stay up to date.

We expect that further research for obtaining equally unpredictable pseudorandom numbers will be needed. Some components of the system, PRNG in particular, might also benefit from such systems as Dfinity while remaining Ethereum compatible.

3. Obtaining random numbers

In DAO.Casino context equal unpredictability of game outcomes that is determines by PRNG is what makes the game fair.

Obtaining randomness (RNG), or to be more precise pseudo-randomness (PRNG) in the context of gambling games is a complex issue. There are different approaches for obtaining such values. Mathematical proofs and high quality of the selected method is important. In simple words, pseudo-random number that determines an outcome of the game should be equally unpredictable for all parties involved. Only in this case we can say that the players and the casino are protected from fraud, and the scheme is tested and reliable.

This part describes existing methods of obtaining pseudo random numbers in deterministic virtual machine that have been implemented before, and the method that DAO.Casino is implementing for two parties games.

Previously implemented methods using Oraclize and internal blockchain data has been also tested by DAO.Casino team. These methods are viable, but can’t be considered sufficiently secure long term. Therefore, we DAO.Casino first release will use the Signidice algorithm, suitable for two parties games, that our team implemented in Solidity.

3.1. Definition

Pseudo random number generator (PRNG)[2] is an algorithm that generates a sequence of numbers, the elements of which are almost independent of each other and are subject to a given distribution (generally uniform).

Random Number Generator (RNG) is an algorithm which generates absolutely random numbers.

Such generators are mostly used for generating unique symmetric and asymmetric encryption keys. They are built mostly from a combination of cryptographically strong PRNG and external entropy sources (and, namely, this combination is commonly understood as RNG).

3.2. Methods of generating random numbers in a centralized casino

In a centralized casino (online and offline), different hardware and software compliant to certain standards is used to generate randomness.

Most poker rooms get special certificates to prove the viability of their RNG and software. Cigital, one of the largest companies in this field[3], is engaged, in certification of the poker software and RNG. The largest poker rooms Full Tilt Poker and PokerStars have the certificate of this company. The basis of any RNG testing is a set of NIST tests (National Institute of Standards and Technology), based on U.S. standard FIPS 140-2 (Federal Information Processing Standard). It includes various tests - from the test on ratio of 0 and 1 in the generated sequence, to the test on LZO algorithm compression (random sequence may not be significantly compressed, because it must not have many repetitive sequences).

The most common method of random number generation is called the linear congruential method. Alternatively, there is the additive congruential method. These methods generate a sequence of numbers satisfying the condition of randomness. The basis for the use of these and other methods of random number generation is the software, infinitely generating numbers, regardless of whether the participant is currently in the game or not. This eliminates the possibility of the player to independently determining the generation method used at this moment, and “guessing” the drawn numbers.

For example, U.S. law requires that the random number generators in slot machines should operate all the time. In addition, the software vendors deal with this issue directly.

FullTilt RNG is built on a similar principle with PokerStars, there are 3 independent generators: hardware RNG with a physical source of entropy and two independent PRNG (ISAAC and OpenSSL).

3.3. Existing methods of obtaining randomness in decentralised gambling games on Ethereum blockchain.

1. Random numbers should be equally unpredictable for all parties.

2. The mechanism of obtaining a random number should be maximally decentralized

3. The possibility of intervention to defraud game results should be very small

3.3.1. Internal method

Data derived from the blockchain data itself i.e block number, timestamps.

This method involves the use of current block hash values or hash block.This method is not considered secure because it has a risk of manipulation on behalf of miners. See: http://martin.swende.se/blog/Breaking_the_house.html

3.3.2. External method

in this case, a value is obtained from external sources. This scheme cannot be considered fully decentralized. Simply speaking, using a single source of PRNG is a bottleneck. One example of this approach is project EtherDice[4]:

The only external dependency is a random number generator because the determination of blockchains prevents from reliably obtaining a random number in a simple manner. The idea is that the contract holds a certain amount of so-called “generations.” The generation starts when one of the two sources of random numbers produces hashes of proposed values. Two sources increase the complexity of compromise, nevertheless they do not cancel that possibility completely. The hashes are saved, and the contract is waiting for some time in order that the rates are obtained in the current generation. After the first bet, the generation takes the next bet for more blocks, and then closes. The contract is waiting for a few more blocks in order to put the generation into the “value disclosure” regime. The sources of random numbers reveal random values (the hashes are verified), after which the players can receive the gains. Several generations work simultaneously and in parallel, so the system is able to accept bets at any time.

**Oracles (authenticated data feed) **- Another external method. External generators of random numbers are translated into the blockchain network. This approach is used, for example, by etheroll.com[5]. The weak spot in this approach is that the oracles can also be compromised.

The source of random values is Random.org, which we can get through Oraclize.it[6]. The latter allows us to increase security with the help of “TLSNotary” technology, which can prove that the number was not changed after it was requested by Random.org. Unfortunately, there is no easy way to verify it directly from a smart contract. Such a verification can be made only after a certain time.

Commit / Reveal - quite an advanced distributed method for producing random values. This approach is actively used in RanDAO[7], Sleth[8], Maker-Darts[9].

The feature of this algorithm for finding random values is a scheme of work in two steps. In the first step, the participants send hashes of random values and deposit a pledge. In the second step, the disclosure of the values takes place, from which the resulting random number is drawn.

If one of the participants cheats and does not disclose a proposed number, then his pledge is lost. This motivates all participants in the generation to be honest. This method is subject to DDOS attack, resulting in the loss of pledges by honest participants.

Different gambling games have different requirements for the architecture and reliability of the PRNG. For a large jackpot drawings, they are higher, and at handing out cards in poker with small bets, they are significantly lower.

Each of the above approaches has its own value. One of the most sophisticated, reliable, long and costly is the Commit / Reveal algorithm (RanDAO).

On the other hand, the simplest one is the internal method. For example, the Rouleth project uses this method. The studies described below have shown that cheating is possible, but insignificant, if the bet does not reach high values.

To fight the miners' fraud, one must choose such parameters that it would be not economically interesting. The Rouleth conducted a simulation, the results of which are available on github[10]. The analysis showed that the attacker must possess at least 3% of the capacity of the network. In this case, the attacker should spend about 23 ETH per block. This value, however, decreases as the possession of the computation capacity by the attacker increases. If he possesses 10% of the network, then only 2 ETH per block is needed for the attack, and 25% of capacity decreases this amount to 1.2 ETH. The attacker will be forced to spend 0.5 ETH if he owns 51% of the network;the entire network is subject to far greater danger than a simple roulette hack.

Game developer & architect should intentionally keep the gain volume low so that it would be economically unprofitable for the attacker to practice deception. Please note that the cheating miner can make a lot of bets per block to increase the probability of winning. Therefore, we have set the maximum number of bets per block to 2 (but this can be changed).

In the world, as of 2017 there are 7 mining pools which have a capacity of more than 3% each.

EthereumLottery[11] project uses a hybrid method of random number generation: through BTCRelay. A contract receives a new block hash from Bitcoin network. The advantage of this approach is sufficiently high reliability, the disadvantage is the low performance of the algorithm, because the Bitcoin block is generated significantly longer than in Ethereum's. Here is what the author of the project says:

1. Casino generates a new pair of private/public keys (PrivKey and PubKey) for some deterministic signing algorithm (e.g.RSA).

2. Casino creates a smart contract, which contains the public key (PubKey), maximal number of participants, and the Ether bounty. (Optionally: casino changes PubKey of the existing smart contract).

3. Player chooses the number to bet on (B), and a random number ® in certain format (e.g. 20 bytes). The player might even specify the range of numbers B, if the rules of the game allow it (odd vs. even, etc.).

4. Player sends a transaction (TX) containing the Ether bet, along

   with the data: B and R.

5. The contract checks validity and format of the numbers B and R. Invalid TX is rejected.

6. Additionally, the contract checks if the number R has*Imagine that the jackpot is $5,000, and the attacker owns 5% of the capacity of the whole Bitcoin network. Imagine that the attacker buys tickets for $5,000, and now owns 50% of all tickets, and the jackpot becomes $10,000. *

7. _At this moment, the attacker has the expectation of $10,000 * 50% * 99.5% (the lottery takes a commission of 0.5%) = $4,975. In one of twenty cases (5% capacity belongs to the attacker), the attacker can replace the block, which will decide the fate of the draw._

8. _If he finds a block, and learns that he has not won in the lottery, he drops it (which gives him another attempt), and if he wins he sends it to the network. This increases the chances of success from 50% to 75% because only in 25% of cases 2 attempts lose._

9. _But when the attacker drops a block - he loses the reward for mining. The losses amount to $4218.75, taking into account the current size of the reward equal to 12.5 BTC._

10. _An increase in the chance to win gives an expectation equal to_

*$10,000 * 75% * 99.5% = $7,462.50. The attacker spent $5000 for tickets, and lost $4218.75 at the unit dropping. This means that such fraud is not economically profitable. It becomes profitable only when the jackpot is more than $10,000.*

3.5. Algorithm implemented in MVP of DAO.Casino protocol

For our first release we chose to implement and test Signidice algorithm proposed to solve PRNG in deterministic virtual machine and described by Gluk256 who suggested a number of other solutions.

Games released for testing by the time of the launch will be using this algorithm. We previously tested external method (using Oraclize) and internal method described above in the test games, but chose this particular method. Signidice has been proposed for Ethereum based games using Ether. Our implementation is adopted for the usage of BET. “Ether bounty” mentioned in the algorithm below is also replaced with BET. Bounty is essentially a staking system.

This algorithm is suitable for those Ethereum-based games, where outcome of every round for the player depends only on the RNG and (optionally) the number, chosen by the player, but not on the action of other players. E.g. it may be suitable for roulette, slots, etc., but not for those games, where outcome depends on the other players or just their number, as may be the case in lotteries. For example, a game of roulette could be modeled as a number of rounds where a single player *plays against the casino. In this case, a pseudorandom number could be generated with the following algorithm.*

1. *Casino generates a new pair of private/public keys (PrivKey and PubKey) for some deterministic signing algorithm (e.g.RSA).*

2. *Casino creates a smart contract, which contains the public key (PubKey), maximal number of participants, and the Ether bounty. (Optionally: casino changes PubKey of the existing smart contract).*

3. *Player chooses the number to bet on (B), and a random number ® in certain format (e.g. 20 bytes). The player might even specify the range of numbers B, if the rules of the game allow it (odd vs. even, etc.).*

4. *Player sends a transaction (TX) containing the Ether bet, along with the data: B and R.*

5. *The contract checks validity and format of the numbers B and R. Invalid TX is rejected.*

6. *Additionally, the contract checks if the number R has already been used by this player in previous rounds, in which case TX is rejected. (This step is necessary if the contract is reused for multiple rounds of the game).*

7. *The contract concatenates the random number R with the public address (A) of the player's Ether account, from which TX was sent: V = A + R. The resulting value V is stored in the contract. The size of V is always the same: size(V) = size(A) + size®. At this point the outcome of the round (win or lose) becomes deterministic.*

8. *Casino must sign the resulting value V with its PrivKey, thus producing the digital signature S = sign(PrivKey, V), and send the corresponding TX, containing S.*

9. *The contract recovers the actual public key (K) from the digital signature S, and verifies that it is equal to the previously published PubKey (K == PubKey). If APK does not match PubKey, or if casino fails to perform step 8 within a predefined time frame, it is tantamount to cheating.* ***In this case the contract sends the bounty to the player along with the original bet, and the contract is closed via suicide. (In case of multiplayer game, the bounty is shared between all players)****.*

10. *The contract uses S as a seed for the predefined PRNG algorithm (e.g. SHA-3 based), which produces the lucky number (L), e.g. between 0 and 36.*

11. *If B corresponds to L, the player wins, otherwise casino wins. The contract sends the bet to the winner.*

12. *Now casino may close the contract and recover the bounty, or initiate a new round of the game. Alternatively, the contract might be programmed to automatically proceed to the next round, unless the casino closes it.*

After casino has chosen the PrivKey, its actions become deterministic. The player can not predict the result of digital signature, and therefore his choice of the random number R can influence the outcome only in the same way as rolling the dice in the real life (hence the name of this algorithm). Thus, neither of the participants can manipulate the outcome in any meaningful way.

The bounty that casino commits to the contract must be high enough to compensate the players for any possible opportunity loss. The time slot must be long enough to accommodate for any kind of network disruption. Although it gives the casino opportunity to postpone the outcome, it has no incentive to do so, since the outcome is still deterministic. Postponing without a valid reason will only result in reputational loss, whereas no financial gain is possible. If the casino wants to maintain good reputation, it might even intercept the player's TX before the next Ether block is mined, immediately calculate the outcome of the game, and reward the player in case he wins, even before the step 7 occurs (before corresponding TX is incorporated into the next mined block). Instant gratification will also help to prevent the reputational loss in case of long delay (e.g. due to the network disruption). It might also allow the player to start a new round without waiting for the next block to be mined.

The same smart contract can be reused multiple times, provided that the player never chooses the same number R twice (otherwise he can predict the outcome). Hence, the necessity of the step 5 – the contract should reject any TX, where the same player uses the same number R in the same contract twice. Alternatively, the casino can regularly change the pair of PrivKey/PubKey, publishing the PubKey before every new round. But it requires an additional TX, resulting in unnecessary delays and increasing TX costs.

Nothing should prevent multiple players from using the same smart contract simultaneously. Different players even choose the numbers N, previously used by other players: V will be unique, since A is unique. The number of possible participants will only be limited by the bounty the casino is willing to commit.

The game of roulette in online casino was used purely as an example. The Signidice algorithm can be used in any case, where transactions occur between the two participants. In this case the Signidice is better than RANDAO: the players neither have to trust the oracles (in case of external oracle use), nor can disrupt the next round of the game by refusing to reveal the committed number (in case of players being the oracles).

As far as we're concerned this method haven't been yet implemented. See Gluk256 repository for this algorithm here, and our implementation in progress here.

4. Initial test platform

Dao.Casino platform is an example of an online platform where the developers can showcase their contract based games and the players can discover them. It is important to emphasise again that this particular platform is not a central point of the DAO.casino project.

By the time of system launch the main goal of the platform is to provide an interface. As we have mentioned before, at least one front-end platform is needed for the whole reward system to make sense.

In the alpha version that we released front end is stored on a central server. The platform is open source and will be released by completion under Apache 2.0 licence

Alpha platform tasks are:

- Provide an example of a frontend platform that plugs into business logic expressed on Ethereum.
- Show a basic tool for game discovery and ranking
- Show user interface familiar to the wide range of players, not just decentralisation enthusiasts
- Provide a simple way to deposit and withdraw BET tokens on testnet

- Re-direct developers to developers documentation, wikis and code.

Thematic priority of alpha platform are classic gambling games: card games, slot machines. Currently Blackjack from partner’s team and Dice from DAO.Casino core team are available for testing.

Alpha platform will also play a role of information hub that aggregates developer’s resources: documentation, wikis, tutorials and examples.

In the next release decentralised hosting solutions will be used i.e. IPFS, Swarm.

Conclusion

Smart-contract powered gambling games are gaining traction, we believe that gambling related use cases for Ethereum can be developed beyond a few experimental provably fair games, into a sustainable ecosystem - a backbone that caters for all aspects of gambling industry.

Three issues need to be addressed for this to happen:

- There should be a system of incentives for all parties that add value to the system. Apart from provable fairness of the game, that will drive adoption and further development.

- Reliable PRNG. Most likely different sources of randomness for different games and most likely secured by economics, not just technical aspects.

- There should be ways for the game developers to create games on Ethereum without deep knowledge of Solidity: replicable contract templates, examples and documentation.

- There should be ways for platform operators to run set up a platfrom that plugs into Ethereum and display games from independent creators without deep and detailed knowledge of Ethereum ecosystem.

Appendix

Crowdfunding, token and funds distribution

References

[1]Market volume of online gaming stats http://www.gbgc.com, https://slotegrator.com https://www.statista.com/statistics/270728/market-volume-of-online-gaming-worldwide/

[2] Random Number Generation - https://en.wikipedia.org/wiki/Random_number_generation

[3] Habrahabr - The Truth about RNG of Pokerooms.

https://habrahabr.ru/company/pokeroff/blog/95090/

[4] Etherdice - https://etherdice.io

[5] Etheroll - Provably fair Ether gambling on the Ethereum blockchain

http://etheroll.com/

[6] Oraclize - A reliable bridge between smart contracts and the

Internet http://www.oraclize.it/

[7] RanDAO - A DAO working as RNG of Ethereum.

https://github.com/randao/randao

[8] Sleth - Ethereum Slot Machine. https://github.com/jorisbontje/sleth

[9] Maker Darts - A Random Number Generating Game for Ethereum.

https://github.com/makerdao/maker-darts

[10] Rouleth - A provably fair roulette : note on Random Number

Generation.

https://github.com/Bunjin/Rouleth/blob/master/Provably_Fair_No_Cheatin

[11] EthereumLottery - The First Ethereum Lottery in the World.

https://ethereumlottery.net/