
# BITPOKER: A BLOCKCHAIN – POWERED P2P ONLINE POKER PLATFORM

by Lucas Cullen

# TABLE OF CONTENT

# 1. ABSTRACT

Online poker in the context of online gambling is a relatively new phenomenon nearly defunct before 2003. During the recent years it has grown exponentially in terms of businesses, users and the circulation of betting capital. However, factors such as minimal transparency and human interference in the outcomes of hands have also grown. The online poker industry prefers to work inside a black box, where the likelihood of insider cheating and collusions increases. This makes it difficult for players to trust the outcomes of their bets.

A blockchain-powered version of traditional online poker solves the problem of trust and opaqueness, while reducing operational costs and regulatory burdens and introducing incentive mechanisms for participants.

Taking inspirations from the decentralized e-commerce market OpenBazaar.com, we propose to design a similar peer-to-peer protocol of turn based games, such as online poker, in which no central actor can control the outcome and thus rig the game and is provably fair. The game uses Bitcoin (or other digital tokens) and lightning network to settle bets between actors, and a blockchain to persist the state of the game.

Most blockchains are too slow for turned based games, but not all turns need to persisted back to the blockchain. For example, in poker, turns can be stored in memory on clients as "mini chains". Only when the outcome of the game is required such as awarding the pot, is the data required to be persisted back to the blockchain. Furthermore, players could agree this could be a higher cadence, such as each orbit, to save on fees.

Its hoped, that different clients developed in different programming languages will be built.

This document discusses the design and implementation of such a hybrid system from developers' and players' perspective on the abstract level.

# 2. INTRODUCTION

## 2.1. LIMITATIONS OF EXISTING POKER PLATFORMS

The online poker market has grown to be one of the largest in the online gambling industry. But until now, there have been only rough estimates about its total market size and nobody have knowledge about the revenue of online poker businesses.

We studied various such discrepancies in the poker market before concluding that there is an absolute lack of transparency between the players and the online poker platforms they play at.

Such black-box game hosting models always have an advantage over the players. There is always a central authority lurking over the players' undisclosed cards, and manipulating the hands using special software and bots. The outcomes of the bets, therefore, are principally disastrous for players.

Furthermore, there are also issues related to raking and racketeering from the casino's end. In an ordinary poker room, players are charged commissions that seem less but in reality are very large. All classic poker systems recycle players' deposits for yielding commissions. On average, 70% of the players' deposits is moving to the rake, which is plainly unfair. Only a few years ago, if you were a successful player on PokerStars, you could get up to 70% rakeback (return of commission paid by a player in different bonuses). But now, it is moved down to 10-15%!

We treat such loopholes as major setbacks for the entire online poker industry. Poker is a brilliant game, but it should not turn into an excuse for tricking the players into losing their money. (see Black Friday — https://www.pokernews.com/news/2016/04/black-friday-five-years-later-24506.htm)

We support regulation as one of the defenses against the malicious practices. But regulation itself is tedious, unfair and heavily centralized process. It is impossible for government to detect discrepancies in the online casino platforms. This is one of the reasons why many countries have simply preferred to ban casino operations than regulate them.

Based on the aforementioned disadvantages, we can safely assume that the growth of online poker industry is jammed between greedy companies and insufficient government control. What poker players need is a trustless system governed by cryptographic proofs instead of closed-door policies managed and administered by trusted third parties.

## 2.2. WHAT IS BITPOKER?

BitPoker proposes to solve the black-box and regulatory hurdles by integrating the blockchain and P2P technology into the existing poker models. It is a mathematically-driven trustless system that replace trusted third parties with autonomous agents. These autonomous agents can be anything from Ethereum or RSK smart contracts to Lightning Network.

The economy of such an autonomous and transparent system will be driven by smart decentralized currencies like Bitcoin. Their inherent properties will help reducing the transaction costs and circumvent regionally-imposed economic restrictions on poker gambling. Such a system will also reduce the overall operational cost of the online poker platform, resulting in fair and lesser commission cuts and improved player potential.

It simple translates to one simple fact: BitPoker enables a consensus-based poker system in which is run and controlled by users themselves. The money is distributed only among the participants, including developers, players and affiliates, leaving no scope for trusted third parties to flourish, at all.

## 2.3. BITPOKER BUSINESS MODEL

BitPoker Business Model is designed to benefit every participant, even if you are a player, a developer or an affiliate. A provably fair and transparent platform based on self-governed Ethereum blockchain protects players from raking and racketeering which are otherwise rampaged in poker platforms. The decentralized model allocates tasks of a central server among many participants while rewarding them attractive incentives.

**For players:**

- the smallest rake in history
- control of own funds
- influence on games and rules
- incredible transparency and fairness of the games
- the biggest poker community with no restrictions
- Privacy and data protection

**For developers/dealers:** You can set up your own poker room using bitpoker.io as a platform, and gain the profit from the running games under your host.

**For affiliates:** You will get rewarded for attracting players and system promotion on a fair basis with no change in conditions like classic poker rooms do.

**For investors:** Earn with the token's growth as a whole.

# 3. THE BITPOKER PROTOCOL

BitPoker does not intend to be just another poker platform, but a protocol in itself to drive further innovation in the industry. That being said, anybody interested in writing their own clients, in their own programming language, will be able to use the open-source BitPoker pr:otocol.

Here are some key points about the protocol to take away

**Transparency & Free:** The proposed BitPoker protocol is intended to stay open and available for pull requests and peer reviews.

**Provable:** The proposed BitPoker protocol will utilize mental poker algorithms to eliminate dependencies on trusted third parties. Users will be encouraged to inspect the shuffling algorithms, software and messages over the wire.

**Fast:** The proposed BitPoker protocol will make poker clients faster than any traditional poker platform by integrating Whisper and ZeroMQ for messaging, and Bitcoin for withdrawals and deposits.

**No Registration:** Players will not be required to submit any personal information other than an IP address and Bitcoin address. While the first clients will communicate using the Whisper protocol, we intend to create an additional I TOR client should there be a demand for it.

## 3.1. HOW IT WORKS: AN OVERVIEW

Each client connects to one another in the “lobby”. They can look for players who are either looking to start a game, or are requesting to join a running game. Signed messages are dispatched to all players, while referencing the existing message; thus, like a blockchain of messages. Here is a breakdown of events that follow:

- Table reaches consensus on whose turn to act based off the game contract
- Table reaches consensus on the legal moves / actions a player can make
- Table waits for a signed message from that player
- All other players validate that message
- Repeat

We propose two versions of how the game would develop: one using Ethereum / RSK smart contracts, and the other using a P2P network protocol. Let's look at this overview to understand it further.

**Using Ethereum / RSK Contracts:**

1. The game is defined as an Ethereum contract;
2. Players agree to the table contract;
3. Each player's actions are defined as inputs for the hand contract;
4. After the hand has ended, each player verifies the integrity of the hand contract. It is in everybody's best interest to verify correctly [Game Theory Citation]
5. The hand message chain is then executed on the Ethereum network for the pot to be awarded

**Using P2P Network Protocol**

1. Players connect to each other via a P2P network protocol.
2. A player either looks to join a table and reviews the contract.
3. A player can choose to start a table be defining a table contract.
4. Tables should also broadcast their game, status and number of current .players to other tables for better network propagation.
5. Leaving the table (closing the channel)
6. Lightning network will facilitate micro payments "off chain". The table can agree to bring them "on chain" after n hands are dealt.

## 3.2. THE CONTRACT SYSTEM EXPLAINED

The set of rules defined in this system is one of the major contributors of our project, and together they are referred as contract. The proposed system allows anyone to develop variations of the poker game, such as the "Seven Deuce" rule, Omaha, etc. In the contract below, we refer Texas Hold'em Poker game as an Enum.

### 3.2.1. MESSAGES

All actions are sent as JSON RPC. They must include a public key hash and should be signed. The payload must also reference their previous message hash.

1. Concatenate the payload the values
2. Hash the payload of step 1
3. Sign the output of step 2

**General message object as a JSON RPC param**

| Property | Eg |
|---|---|
| Version | Message Version |
| Id | GUID |
| Bitcoin Address (Public Key Hash) | msPJhg9GPzMN6twknwmSQvrUKZbZnk51Tv |
| Method | Enum (TABLE, ACTION, BUYIN, SHUFFLE, DECK) |
| Payload Hash | SHA256 |
| Message Signature | TODO |
| Previous Hash | TODO |

### 3.2.2. ADDING A TABLE CONTRACT

A client will define the table contract and store it locally. The client will become the table starter, and will thus define the conditions of that game. The parameters for a table are defined in the following schema. Developers are encouraged to create their own algorithms, such as voting or anti-collusion.

1. Encryption Algorithm (Enum AES-256)
2. Hash Algorithm (Enum SHA-256)
3. Id (GUID)
4. Currency (Enum)
5. Blinds
6. Rake*
7. Min players
8. Max players
9. Game type (Enum, No Limit Texas Holdem) *
10. Other (straddles, "run it twice") *
11. Channel Address / multisig

12. Consensus Algorithm
13. Anti Collusion Algorithm / Contract
14. Version
15. Voting Algorithm / Contract
16. Channel Address

### 3.2.3. JOINING A TABLE

Users send their intent to join a table by the JoinTable method. This is analogous to choosing a seat and sitting down at the table. Once the table reaches the maximum amount of players, or the players vote to start the table, a multi signature address is created. The required signatures are part of the agreed table contract.

### 3.2.4. BUYING IN

All players buying in open a lightning payment channel (see 3.3) with the multi signature address of the table. Players must add BTC within the range for the table contract (MinBuyIn, MaxBuyIn).

### 3.2.5. WITNESS NODES

Game witness can also be allowed or chosen to arbitrate a game. The witness could also help network propagation. A witness would be chosen by the table starter and a small rake will be paid to him.

There might become a market for reputable witnesses based off a HTTPS DNS endpoint and earn small revenues for witnessing hands.

1. Alice and Bob create a 2 of 2 address
2. Alice creates a deposit transaction
3. Bob creates a deposit transaction
4. Alice creates a refund transaction but does not broadcast
5. Bob creates a refund transaction but does not broadcast

### 3.2.6. GAME PLAY

The dealer's client is responsible for the orchestration of the game. As the dealer position rotates, there is never a risk of centralisation. The intent is to limit network traffic.

1. **Define the hand contract:** A hand contract is defined by the dealer at the start of each hand. The contract references the table contract, in which the following things are specified:

- The players and their seat positions
- The stack of each player
- An ID as a GUID

2. **Shuffle the deck:** The hand contract defines the role of dealer, small blind and big blind; whilst the deck is represented by an array[52] of bytes. Have a look at this lookup table:

- Card[0]=AH
- Card[1]=KH
- Card[2]=QH
- Card[3]=JH
- Card[51]=2C

3. **Encrypt the deck:** The deck is encrypted multiple times using a commutative algorithm such as RSA. The dealer shuffles the deck without disclosing the unencrypted result.

- Card[0]=AC
- Card[1]=3S
- Card[2]=AH
- Card[3]=2S

First, an array of 52 private key is created, 16 bytes represented at base 64. They do not leave the dealer’s computer. In the next step, each card is double encrypted. First round of encryption with the hand key. Then each card is encrypted again with the matching key and represented as base64. As the deck is encrypted, and assumed shuffled, the Small Blind has no way to know the contents of the deck.

It then encrypts the deck again and shuffles, and sends the result back to the dealer.

4. **Post blinds:** Using the lightning proposal, the Small Blind creates an unsigned TX of 0.001 to the Big Blind.

5. **Pre flop round:** We take Hold'em as the prime example to illustrate how the cards will be dealt, i.e. one at a time, starting from the left of the dealer (small blind).

- Card[0] → Bob
- Card[1] → Alice
- Card[2] → Bob
- Card[3] → Alice
- Alice → Action request message to Bob.
- Bob → Returns signed action message to Alice
- Alice → Checks signature, and adds action response to the block
- Alice → Broadcasts the concatenated block to all players
- All players → Verify the block and signature
- All players → Return verification message

6. **Flop, Turn and River:** The client software co-ordinates the game, based off agreed game rules. The algorithm is explained as follows

- Enforces action rules of its own player, such as check, bet or fold;
- If the action involves money, creates the tx;
- Creates a signed message and broadcasts to each player;
- Waits for next action message;
- Validates the message.

7. **Award the Hand:** Once the hand has been played, the table reaches consensus. The signed game history could then be persisted into the Ethereum blockchain while referencing previous hands. The table would also include a parameter, defining when to commit the hand (or hand history) to a chain. The more frequently it is done the more fees it will incur.

8. **Cash Out:** Closing the channel.