Blockrate Binaries on Bitcoin Mainnet: Formalizing the Powswap Protocol

Thomas Hartman, Gleb Naumenko, Antoine Riard
thomashartman1@gmail.com, gleb@thelab31.xyz, antoine@thelab31.xyz

Abstract. One of the long-term factors of Bitcoin stability and security is the existence of a consistently high level of hashrate. This hashrate should reach a threshold rendering it computationally prohibitive for any entity to unilaterally re-order the chain of blocks. This prohibition might work under the condition that a sufficiently distributed market of mining nodes flourish and those mining nodes earn a steady Bitcoin income. Running a bitcoin mine has turned out to be a high-risk operation heavily dependent on the local energy market. Traditional financial tools to hedge against energy fluctuations have the downside of relying on trusted third parties.

We propose a formalization of the powswap protocol, blockrate binaries. Blockrate binaries enable hedging against energy fluctuations and changes in the hashrate production with pure Bitcoin native contracts. The contracts are trustless and rely solely on Script primitives. They can be deployed today with no consensus changes. At scale, they enable to build a censorship-resistant futures market for bitcoin ultimately grounded in kWh.

1 Introduction

The Bitcoin internal economy is in its early days, and there is no consistent anchoring of Bitcoin mining income to real-world energy flows. Better hedging against purchasing power volatility would make Bitcoin a more saleable asset.

Blockrate Binary Options are financial instruments enabling Bitcoin users to make peer-to-peer binary contracts on the future rate of block discovery. Bitcoin owners can use them to hedge purchasing power, both by transacting in them and by deriving (hard to censor) market expectations of future Bitcoin purchasing power from ongoing contracts. This is especially useful in situations where financial repression impacts the availability of more convenient liquidity management tools.

That being said, even under non-repressive market conditions, Blockrate Binaries may minimize surveillance and eliminate custody.

Technically, a Blockrate Binary is a contract on the Bitcoin blockchain enforced by pre-signed time-sensitive transactions. It does not require trusting custodians or oracles but may require careful counterparty selection for the best results. It relies on primitives already available on the Bitcoin mainnet.

Our work formalizes the PowSwap [1, 2] idea. We attempt to narrow and shape the discussion by outlining the exact construction and covering the most important implementation aspects.

2 Motivation: Censorship-resistant future markets

When Bitcoin owners have to convert into fiat (for spending or other reasons), they are prone to many exchange-related risks:

- custody failures;
- exposure of sensitive user data;
- failures of pegged side currencies;
- regulatory difficulties for on/off-ramps;

We propose Blockrate Binary Contracts as a trading instrument that mitigates these risks.

Blockrate binaries allow direct trading against a derivative of the energy cost (in hashes) of issuing new Bitcoins. Granting that energy cost and overall purchasing power must correlate to some degree, this enables censorship-resistant future bitcoin purchasing power discovery through historic and available offers.

The operations of a functional Blockrate Binary market require only the following:

- finding a counterparty to trade with;
- accessing information about blockrate fluctuations.

The former requires a bulletin board with anonymous identities, and the latter requires inspecting historical Bitcoin blocks (at minimum, while in practice the more information the better, as in any trading).

We believe that Blockrate Binary Contracts could become a useful financial tool for the success of Bitcoin as a hard currency.

3 Background

Bitcoin is a peer-to-peer electronic cash system [3] relying on a large network of independent nodes exchanging blocks of transactions.

3.1 Proof of Work

To maintain the consistency of the network, Bitcoin nodes rely on the Heaviest Chain Rule to choose which chain of blocks to follow, ultimately bringing them

to the shared state. The *heaviness* of the chain is determined by Proof-of-Work [4] associated with every block.

Bitcoin miners are rewarded with new bitcoins produced and transaction fees. Over time, Bitcoin mining went from an enthusiastic activity to a profitable one and finally became a niche business relying on access to a competitive source of energy, special-purpose hardware (lately, ASICs), a dedicated software stack [5], and non-trivial operational expertise.

3.2 Bitcoin Script

Bitcoin transactions rely on spending existing coins by satisfying the rule they are locked under [1] (e.g. requiring a corresponding digital signature authorizing a spend).

At the spending time, the spender asks the recipient to construct the address (transaction destination) encoding the spending rules. All nodes in the network are able to verify these rules, as the transactions are recorded in the blockchain. Balances are then derived from the latest blockchain state and are usually stored in the form of a UTXO (Unspent transaction output) set.

The spending rules are expressed in the language called Bitcoin Script. It utilizes opcodes: e.g., *OP_CHECKSIG*.

3.2.1 Timelocks

If a signature covers a timelock, the transaction becomes spendable either at reaching the blockchain height (a block index), or a timestamp recorded by miners in block headers.

Timelocks could be absolute (a fixed index/time) or relative (time to pass since a transaction was mined).

Timelocks are realized either through Bitcoin Script (e.g., *OP_CSV 700,000* making spending available only after the given block), or by signing the transaction's *nLockTime/nSequence* field in advance (making the signature invalid if this field is malleated).

3.3 Off-chain protocols and state revocation

Moving payments and smart contracts *off the chain* allows more transactions within the limited block space, an alternative approach to privacy and payment finality [6].

This is achieved by locking funds in a shared UTXO, and exchanging pre-signed transactions enforcing a virtual state (e.g., a balance distribution between two accounts). The state could be enforced via an on-chain transaction.

To make sure a malicious counterparty can't submit an outdated state on-chain, every update is accompanied by invalidating the previous state. Invalidating is usually done via Poon-Dryja mechanism[6] based on penalties:

[1] Apart from transactions minting new coins

initiator of *N+1* update discloses a revocation secret for the state *N*, which could be then used to take all the funds if they submit state *N*.

4 Blockrate Binaries

A Binary option is a financial tool to make predictions (e.g., *whether the price will go up or down*), with a yes/no outcome. Blockrate Binaries refer to the predictions of the Bitcoin block issuance rate, and therefore on the underlying hashrate fluctuations.

4.1 Hashrate

Hashrate charts are common artifacts of the bitcoin ecosystem. The hashrate is a measure on the probabilistic proof-of-work as a numerator and the time period in epoch as a denominator, expressed as:

$$\frac{ChainWork(EndBlock) - ChainWork(StartBlock)}{Timestamp(EndBlock) - Timestamp(StartBlock)}$$

Median time past is used for timestamp, to guarantee that consecutive timestamps never decrease.

Hashrate fluctuation is a valuable statistical tool for mining operations, as increases or decreases translate entry or exit of competitors or changes in the mining process of productions.

Viewed as a binary option, this hashrate measure can be considered a bet on whether the chain tip state reaches first the *Chainwork(EndBlock)* or *Timestamp(EndBlock)* value. This will be reached at an unknown future block, starting from a known past block. Such a binary option, however, is not expressible with mainnet Bitcoin Script primitives.

4.2 Blockrate

The blockrate is a prediction on the Bitcoin issuance rate with the height period as a numerator and time period in epoch as a denominator, expressed as:

$$\frac{Height(EndBlock) - Height(StartBlock)}{Timestamp(EndBlock) - Timestamp(StartBlock)}$$

The binary option can be considered as a bet on whether the chain tip state reaches first the *Height(EndBlock)*, or *Timestamp(Endblock)* value. Again, this value is reached at a yet unknown future block starting from a known past block.

This binary option *is* expressible in mainnet script, as follows.

4.3 Blockrate Binaries Contract

A blockrate binary option is a Bitcoin contracting protocol where two participants agree to lock their funds (in a UTXO locked by a multi-signature)

towards a bet on blockrate changes over a certain time frame. Once the outcome is known, the full amount is released to the winner (**Fig. 1**).

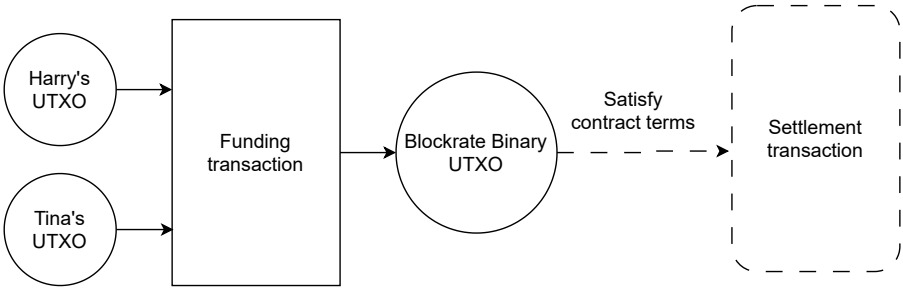

Fig. 1 Blockrate Binary is initiated when Harry and Tina lock their funds in a single UTXO. This UTXO could be spent based on the contract terms.

The rules are enforced by producing pre-signed transactions and releasing the amount to a relevant participant when a chosen chain height or block timestamp is reached.

4.4 Basic Protocol Flow

We will now describe the basic version of the protocol in an agnostic way towards transaction construction. The implementation details are discussed in Section 6.

Our protocol description refers to two protocol participants:

- Harry is betting a certain block height (*HeightStrike*) is reached first;
- Tina is betting a certain timestamp (*TimeStrike*) is reached first;
- Harry and Tina deposit certain amounts of bitcoin;
- If Harry wins, he takes everything[2];
- If Tina wins, she takes everything (**Fig. 2**).

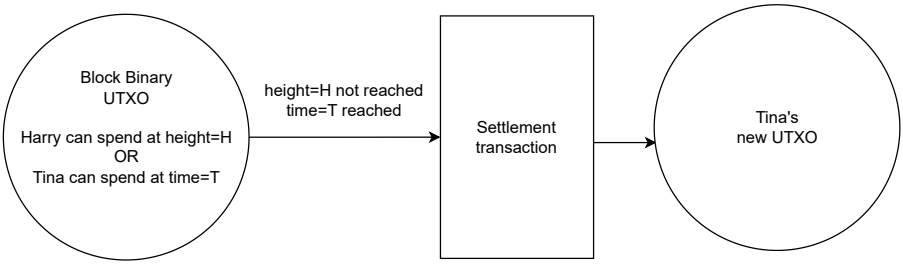

Fig. 2 Blockrate Binary Option terms enforce at which moment every participant could take the funds. Tina wins, because time=T was reached before height=H.

[2]Following a winner-takes-all pattern is not necessary. However, we presume it is the default.

Now, let's discuss the steps to enter the contract, once both parties agree to the terms and can communicate.

4.4.1 Funding

The contract is represented by a 2-of-2 multi-signature UTXO, with spending rules enforcing the contract conditions. To fund the contract, participants should exchange signatures which allow each to enforce the contract through their version of the Settlement transaction. This transaction may spend the contract UTXO to an address they control once they win.

Only once both participants have the signatures, they do authorize Funding. Once the Funding Transaction is mined, the contract is established.

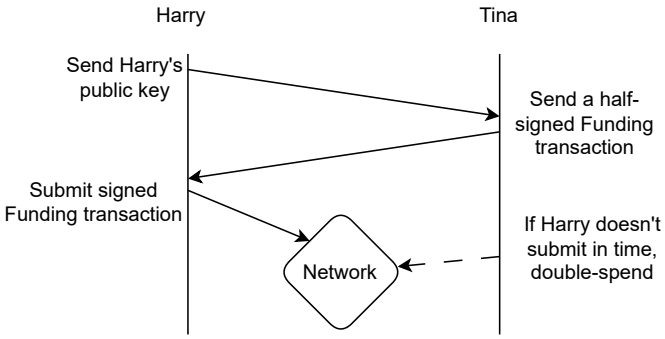

Fig. 3 Funding a Blockrate Binary Option involves one round-trip. Both parties should make sure the transaction confirms on-chain.

If Harry becomes unresponsive after receiving a signature from Tina, she should double-spend her coins into an independent UTXO (**Fig. 3**). Until Tina does so, Harry may start the contract arbitrarily later on, which is undesirable for Tina (and vice versa).

4.4.2 Settlement

Uncontested

Once *HeightStrike* or *TimeStrike* is reached, the winning participant can spend the UTXO through their Settlement transaction only by using their private key (without cooperation).

Contested

Both *HeightStrike* and *TimeStrike* may be reached roughly at the same time (**Fig. 4**). Since Settlement transactions are incompatible, this could result in a *scorched earth* race between the two transactions, where both parties fee-bump their transactions, resulting in the following:

- higher fees make the participation cost unnecessarily expensive;
- allocating capital for fee bumping in advance makes participation less capital-efficient.

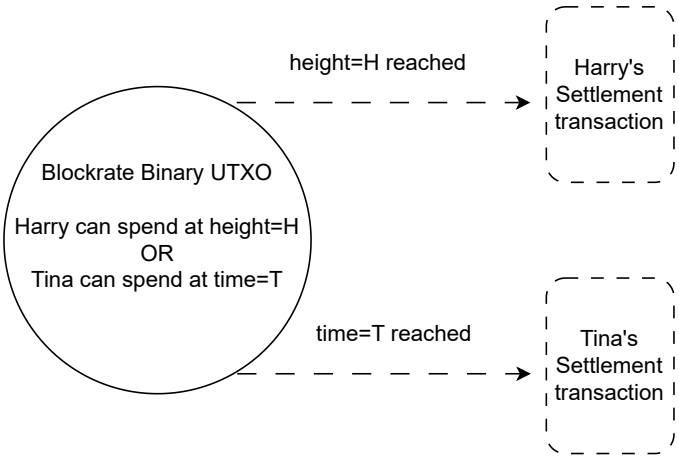

Fig. 4 Both conditions could be satisfied at the same time. The conflict of two incompatible transactions may result in a contestation fee-race.

In Section 8, we further discuss the game-theoretic model of contentions and risk minimization strategies.

4.4.3 Early Termination

Participants may agree to terminate the contract earlier: e.g. a losing side agrees to unlock the funds for a reward (**Fig. 5**). This allows better capital efficiency (the funds are released immediately) and hides contract details from external observers (see Section 5.3).

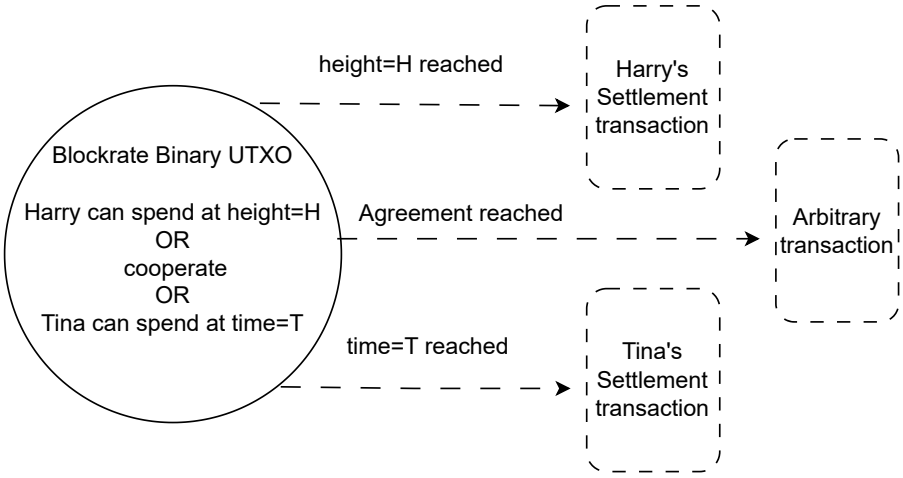

Fig. 5 Harry and Tina may cooperate to terminate the contract earlier via an arbitrary transaction they agree on. This allows avoiding a contestation fee-race.

This technique could be used to update contact details (e.g., make it longer, or change the odds) by spending it into a new contract. It comes at the cost of an on-chain transaction.

Off-chain update

Participants could cooperatively reduce (either or both) strike parameters by exchanging signatures corresponding to a new Settlement transaction.

Increasing one of the parameters won't be effective because the previous conditions are not invalidated. Instead, it would result in contention between the lower values for each strike parameter.

4.5 (Optional) Advanced Protocol Flow: Cooperative Update

Off-chain changes to a basic protocol can only shorten contract duration. Other changes require an on-chain transaction by early terminating the contract and spending it into a new contract.

Adding Poon-Dryja [6] revocation mechanism enables arbitrary off-chain contract changes (except replacing participants). The contract changes can affect the bet parameters or the payout distribution.

To apply it, an intermediate transaction should be added between Funding and Settlement (**Fig. 6**). This transaction commits to a certain state of the protocol. If the state is updated, a corresponding secret is disclosed (**Fig. 7**).

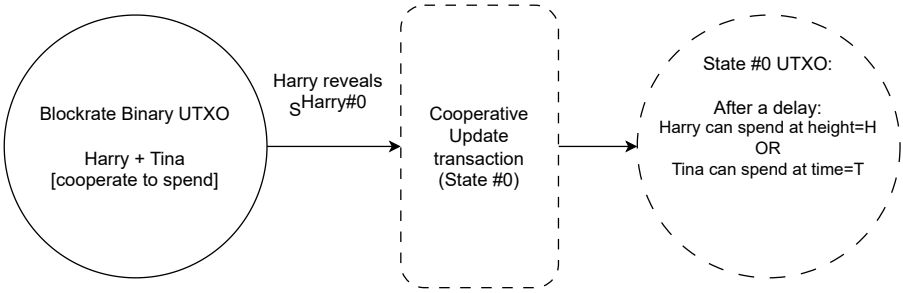

Fig. 6 The protocol could be upgraded to support arbitrary contract updates. This requires adding a new tx type: Cooperative Update Transaction; and modifying Settlement transactions.

If either party submits an outdated Cooperative Update transaction on-chain, the other party can use this secret to take all contract funds. To allow this reaction, the honest spending of the Cooperative Update transaction is guarded by a timelock (**Fig. 8**).

The features of this construction could be enhanced (e.g., participant replacement) by using PTLC [7]. We leave exploring this direction for future work.

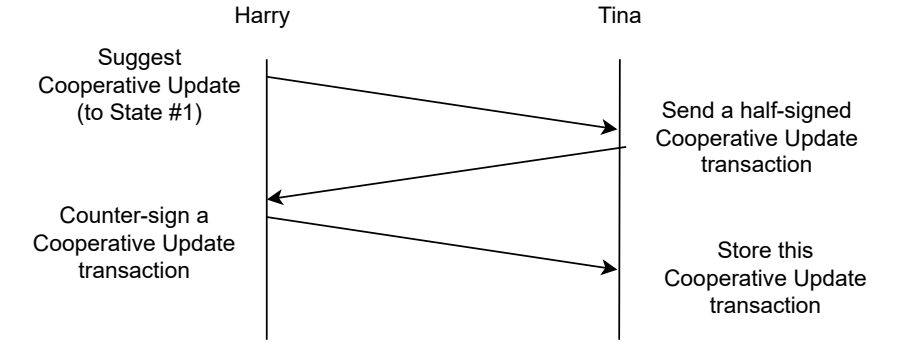

Fig. 7 Cooperative Updates require two round-trips.

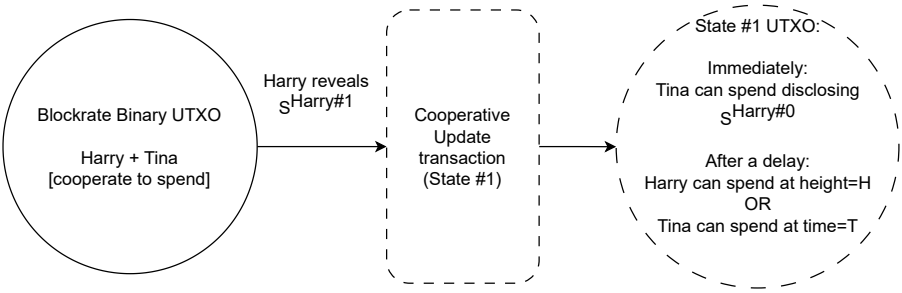

Fig. 8 Cooperative Updates require participants to reveal secrets making sure the previous state won't be transmitted. If it's transmitted anyway, the secret allows to take all the funds from it (*a punishment*).

5 Threat Model

5.1 Transaction Delay

A malicious participant may attempt to delay the mining of contract transactions through the following mechanisms:

- p2p-level attacks (e.g., Time-Dilation [8] prevents a target Bitcoin node from receiving blockchain progress, thus making it impossible to timely react);
- mempool-level attacks (e.g., Pinning [9] prevents a transaction from propagating through the network and thus reduces its chances of being mined);
- bribing miners.

A broader overview of these attacks is presented in Section 9. We will now discuss how they are applied to particular transactions.

Since the contract conditions are expressed in absolute locks, delaying **Funding** does not affect the contract flow directly. It is only required for the Funding transaction to be confirmed before contract conditions are satisfied.

However, it may allow either participant to execute a **free option**: an opportunity to cancel the contract after observing undesirable events. This

could result in a *fee-bumping scorched earth* (e.g., Harry fee-bumping the Funding transaction, and Tina is fee-bumping a cancellation).

There is also a risk of withholding **Settlement**, which effectively changes the bet conditions. For example, Harry was originally able to claim the funds at block *N*, but due to censorship would not be able to claim until *N+1*.

The p2p-level and mempool-level risk mitigations are in progress at the Bitcoin Core software level. Mitigating miner bribe risks depends on the exact bribe implementation, which currently does not exist in the Bitcoin ecosystem. Since these mechanisms introduce trade-offs, we don't implement them and leave them for future consideration.

A special case of transaction delay risks is ineffective dynamic fee management (if the counterparty refuses to cooperate). Proper fees for the Settlement transaction can't be predicted and negotiated at Funding time.

Thus, participants either pre-sign versions of Settlement transactions with different fees (capital-inefficient if overestimated); or the protocol should allow a non-interactive fee increase at settlement time.

The security of these non-interactive techniques (e.g., Anchor Outputs [10]) is imperfect. The implementations should take these risks into account and follow the latest protocol developments.

5.2 Gaming block headers

Unlike the monotonic and predictable block index, timestamps in block headers could be gamed by the miners producing those blocks [11]. Thus, the outcome of the contract could be affected by these manipulations.

Not only the last block could be gamed, but any block over the contract period could affect the contract outcome.

A solution has been proposed in the past to mitigate this attack [12], although it is not implemented.

5.3 Privacy

Every Blockrate Binary Contract settlement reveals the following data in the Bitcoin blockchain:

- contract conditions;
- funding UTXOs and settlement UTXOs;
- transaction metadata (time, fee management strategy, contract offers parameters, etc.).

The participants may use *Early Termination* or *Cooperative Update* to settle the contract without revealing contract conditions. In this case, it would look like a one-input-two-output transaction, a pattern used for any simple payment.

6 Contract Implementation details

6.1 Timelock choice

6.1.1 absolute or relative

The examples we previously introduced refer to absolute timelocks: either a fixed height or a timestamp should be reached for either party to claim the pot.

Bitcoin also has relative locks, formulated as follows: *the UTXO could be spent if a pre-defined number blocks (or seconds) passed since the transaction creating this UTXO was mined.*

The security concerns of relative locks are as follows:

1. **Funding delay** changes the Contract subject period, allowing the attacker to chose when the contract starts, opening up manipulations if the difference period includes an event of high importance;
2. **Free option** and **Settlement delay** risks are the same as for the absolute locks;
3. **Gaming block headers** risks are the same as for the absolute locks.

Additionally, relative locks are much less common among Bitcoin transactions, resulting in a privacy leak.

Although it is possible to mix two types of timelocks in a single contract, we don't recommend this. Slight differences in the security model are more likely to be exploited in an unexpected way.

We recommend the use of absolute timelocks.

6.1.2 transaction-level or script-level

Bitcoin Script allows enforcing locks either at the transaction signature level (the *nLockTime* field of a pre-signed transaction) or at the UTXO Script level (*OP_CLTV* and *OP_CSV*). Our protocol design (see Section 4) is agnostic w.r.t this mechanism.

The UTXO Script level construction is better because it allows choosing a Settlement destination at the last moment instead of the Funding time.

However, the UTXO Script level construction currently has one significant disadvantage: in case of contentions, participants may transmit an opt-out-RBF version of the transaction [13]. In that case, it becomes a relay race instead of a fee race, encouraging users to spam (or even exploit) the peer-to-peer network.

We recommend using the *nLockTime*-version of the protocol.

7 Peer-to-peer Market Infrastructure

Counterparty search

Although Blockrate Binary Options could rely on a centralized service, a fully peer-to-peer operation better withstands market manipulations, censorship, and attacks on privacy.

There are two ways of finding counterparties in a peer-to-peer way:

- over-the-counter mechanisms employing real-life social mechanisms;
- distributed bulletin boards of orders.

The former approach could use the experience of traditional trading, while also inheriting the downsides (e.g., legal risks). The latter approach is less prone to real-world issues by relying on the Bitcoin technology stack (e.g., using Lightning onions as a communication layer).

Market discovery

Market discovery is a crucial component of prediction markets. In traditional financial markets price is defined by the market authorities (e.g., established stock and commodities exchanges).

Peer-to-peer market discovery should accommodate the exchange and aggregation of the available contracts. Building bulletin boards in a trust-minimized, incentive-compatible, and privacy-preserving is an implementation challenge.

Counterparty choice

The game theory of Blockrate Binary contracts is not trivial. In some cases, optimal capital efficiency requires asserting a collaborative attitude of the counterparty, making it important to choose counteparties wisely.

A counterparty reputation system may consider historic behavior in similar contracts, in other Bitcoin protocols (e.g., a Lightning routing node), or even outside the Bitcoin ecosystem.

8 Discussion

8.1 Impact on Mining

Blockrate Binary participants may attempt to influence contract resolution through hashrate access (e.g., deploying/shutting miners, social engineering, or bribing).

If the hashrate availability is non-uniform w.r.t bet sides, and they coordinate the attack, unhealthy hashrate fluctuations may appear.

We believe that the current mining ecosystem is mature enough to withstand these risks. And, in the long-term, we believe that wider access to hedging through Blockrate Binaries only improves the security of the network by reducing the entry bar and thus making it more decentralized.

We are looking forward to more research in this direction.

8.2 Contention Game Theory

As we highlighted in Section 4, a basic Blockrate Binary construction may result in contentions and fee races, which makes these contracts less attractive. The game theory of these contentions (and solutions) requires additional research, especially w.r.t. the asymmetry of censorship opportunities and fee-bumping reserves.

8.3 Advanced Block Binary Contracts

More advanced Blockrate Binary constructions could improve the following aspects:

1. sophisticated contract conditions (e.g., non-binary outcomes, or even increasing the number of parties) to make contracts more flexible;
2. alternative spending conditions (e.g. non-binary outcomes) to reduce the odds of contestations, if they are added at Funding time;
3. replacing a participant with a third-party;
4. using contracts as building blocks for more advanced financial instruments;

8.3.1 Contracts over Payment Channels

The execution of Blockrate Binary Options could take place on top of the payment channels. This would lower the participation costs (transaction fees), and enable a high-frequency mode of operation, and confidentiality.

In the current LN payment channel design [6], the contracts would act as HTLCs. The contracts could be executed both directly and in a routed way. There are following fundamental design challenges with the routed version.

Routing compensation. Currently, the LN is not designed to facilitate operations with non-negligible lock times. If this is abused, routing nodes may simply disallow lengthy payments. Changing the LN to allow pay-per-time-locked is a non-trivial problem[14].

Forwarding synchrony. LN-routed payments operate over the chain of promises, formulated as *you will take the funds if you reveal a preimage for a given hash*, motivating the route participants to propagate the preimage back. Forwarding Blockrate Binary Contracts requires a more sophisticated design. This issue potentially may be solved via barrier escrow [15].

We leave solving these issues for future research.

8.3.2 Oracle-facilitated Contracts

Discreet Log Contracts [16] allow attesting real-world events with a federation of oracles selected by the participants at contract establishment. Oracles may attest blockrate changes (e.g., whether blockrate went up or down over a certain number of blocks).

This construction differs in trust assumptions: oracles may provide fake data, either on purpose or due to an implementation bug. In the future, this risk may be minimized by the wider deployment of fraud proofs and reputation systems for oracles, subject to further research.

9 Related Work

Off-chain protocols were proposed to enhance Bitcoin features [6, 17–19] (increase payment throughput, reduce fees, and offer an alternative approach to privacy). [20, 21] suggest sophisticated vaulting/cold-wallet strategies. [22]

was suggested to enhance Bitcoin fungibility. Blockrate Binaries build up on similar ideas.

The security research of off-chain protocols revealed many risks. Transaction jamming at the peer-to-peer level was studied, and countermeasures were implemented [14]. Transaction pinning also poses the risk to off-chain protocols, and being currently worked on [9]. Delaying transactions via Eclipse [23] and Time-Dilation [8] were studied, and measures were suggested [24]. Affecting mining difficulty via timewarp was demonstrated in this [11]. [25] concluded that transaction withholding by miners is impractical if security parameters are properly chosen. Block Binaries inherit all these risks.

The security properties of Bitcoin's consensus protocol (based on Proof-of-Work) were studied in [26, 27].

The first sketch of an Bitcoin Trustless Hashrate Derivate protocols were suggested in [1, 2], although no detailed specification or security model were proposed.

10 Conclusion

In this paper, we formalized the PowSwap [1, 2] idea as Blockrate Binary Options, which we believe are important as a hedging tool for the Bitcoin ecosystem. We suggested the exact construction and discussed its security and implementation considerations.

We recommend the implementors of Blockrate Binary Options to follow best practices on p2p and mempool safety, and fee management. We recomment implementing them with absolute nLockTime-based timelocks. We recommend considering a reputation system for counterparty selection.

We look forward to the deployment of Blockrate Binary software.

11 Acknowledgements

Thanks to Nadav Kohen for feedback.

Appendix A Blockrate Binary Transactions

A.1 Funding transaction

The contract is established by confirming the following transaction: Harry and Tina providing inputs and locking them in a SegWit v1 Taproot output.

- version: 2
- locktime: chain tip at signing
- txin: Harry inputs + Tina inputs
- txout count: 1 (omitting change)
  - amount: total amount (excluding tx fees)
  - script: aggregate (Musig2) pubkey

A.2 Settlement transaction (basic protocol)

Before exchanging the signatures for the Funding transaction, participants pre-sign two (one per participant) Settlement transactions spending the Funding transactions.

Harry's Settlement transaction

- version: 2
- locktime: HeightStrike
- txin count: 1
  - outpoint: funding_txid and output_index
  - sequence: 0xff_ff_ff_ff (signals replaceability)
  - script_bytes: 0
  - witness: aggregate signature
- txout count: 1
  - amount: balance to be paid (excluding tx fees)
  - script: defined by Harry at funding

Tina's Settlement transaction

- version: 2
- locktime: TimeStrike
- txin count: 1
  - outpoint: funding_txid and output_index
  - sequence: TimeStrike (signals replaceability)
  - script_bytes: 0
  - witness: aggregate signature
- txout count: 1
  - amount: balance to be paid (excluding tx fees)
  - script: defined by Tina at funding

A.3 Early Termination Transaction (basic protocol)

- version: 2
- locktime: 0
- txin count: 1
  - outpoint: funding_txid and output_index
  - sequence: 0xff_ff_ff_ff (signals replaceability)
  - script_bytes: 0
  - witness: aggregate signature
- txout count: 2
  - amount: As negotiated by Alice
  - script: defined by Alice before termination

  - amount: As negotiated by Bob
  - script: defined by Bob before termination

A.4 Cooperative Update Transaction (advanced protocol)

Advanced protocol flow assumes a new intermediate transaction (Cooperative Update), which must be invalidated if the contract moves to a new non-final state.

At any time, each participant carries their own version of the latest Cooperative Update Transaction, along with the secrets required to invalidate previous states.

Harry's Cooperative Update Transaction

- version: 2
- locktime: 0
- txin count: 1
  - outpoint: funding_txid and output_index
  - sequence: 0xff_ff_ff_ff (signals replaceability)
  - script_bytes: 0
  - witness: aggregate signature
- txout count: 2
  - amount: balance to be paid (excluding tx fees)
  - script: revocation_or_timelock

The *revocation_or_timelock* is constructed as follows:

- (Penalty) OP_IF Tina_revocation_key
- (Timelock) OP_ELSE HeightStrike OP_CLTV OP_DROP Harry_delayed_key
- OP_ENDIF OP_CHECKSIG

Harry's Settlement of this state requires using a transaction with *nLockTime=HeightStrike* and witness stack set to *Harry_delayed_signature 1*.

If Harry submitted an outdated version of Cooperative Update, Tina Settles this state immediately with the witness stack of *Tina_revocation_signature*.

Similar rules applies to the Tina's version of Cooperative Update Transaction.

References

[1] Rubin, J.: POWSWAP | Trustless Bitcoin Mining Derivatives Exchange. https://powswap.com. [Online; accessed 2023-02-06]

[2] Rubin, J.: POWSWAP: Oracle Free Bitcoin Hashrate Derivatives. https://rubin.io/bitcoin/2021/12/21/advent-24 (2021)

[3] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com (2009)

[4] Back, A.: Hashcash - a denial of service counter-measure (2002)

[5] Stratum V2 The next-gen protocol for pooled mining. https://stratumprotocol.org/. [Online; accessed 2023-02-06]

[6] Poon, J., Dryja, T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. DRAFT (2016)

[7] Fournier, L.: witness asymmetric payment channels (2020). https://lists.linuxfoundation.org/pipermail/lightning-dev/2020-August/002785.html

[8] Riard, A., Naumenko, G.: Time-Dilation Attacks on the Lightning Network. arXiv (2020). https://doi.org/10.48550/ARXIV.2006.01418. https://arxiv.org/abs/2006.01418

[9] Newbery, J.: What is meant by transaction 'pinning'? https://bitcoin.stackexchange.com/questions/80803/what-is-meant-by-transaction-pinning. [Online; accessed 2023-02-06] (2018)

[10] Jager, J.: Anchor outputs by joostjager · Pull Request 688 · lightning/bolts. https://github.com/lightning/bolts/pull/688. [Online; accessed 2023-02-06] (2020)

[11] Harding, D.: What is time warp attack and how does it work in general? https://bitcoin.stackexchange.com/questions/75831/what-is-time-warp-attack-and-how-does-it-work-in-general. [Online; accessed 2023-02-06] (2018)

[12] Lau, J.: [bitcoin-dev] Getting around to fixing the timewarp attack. https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2018-August/016320.html. [Online; accessed 2023-02-06] (2018)

[13] Riard, A.: On Mempool Funny Games against Multi-Party Funded Transactions. https://lists.linuxfoundation.org/pipermail/lightning-dev/2021-May/003033.html (2021)

[14] Naumenko, G., Riard, A.: Lightning Channel Jamming: Solution Design space. https://jamming-dev.github.io/book/4-design_space.html. [Online; accessed 2023-02-07] (2022)

[15] ZmnSCPxj: A Payment Point Feature Family (MultiSig, DLC,

Escrow, ...). https://lists.linuxfoundation.org/pipermail/lightning-dev/2019-October/002214.html. [Online; accessed 2023-02-07] (2019)

[16] Dryja, T.: Discreet log contracts. URl: https://adiabat.github.io/dlc.pdf (2017)

[17] Decker, C., Russell, R., Osuntokun, O.: eltoo: A simple layer2 protocol for bitcoin. White paper: https://blockstream. com/eltoo. pdf (2018)

[18] Burchert, C., Decker, C., Wattenhofer, R.: Scalable funding of bitcoin micropayment channel networks. Royal Society open science **5**(8), 180089 (2018)

[19] Riard, A., Naumenko, G.: CoinPool Research. https://coinpool.dev. [Online; accessed 2023-02-07] (2022)

[20] Möser, M., Eyal, I., Gün Sirer, E.: Bitcoin covenants. In: Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20, pp. 126–141 (2016). Springer

[21] Swambo, J., Hommel, S., McElrath, B., Bishop, B.: Custody protocols using bitcoin vaults. arXiv preprint arXiv:2005.11776 (2020)

[22] Maxwell, G.: CoinJoin: Bitcoin privacy for the real world. https://bitcointalk.org/index.php?topic=279249.0. [Online; accessed 2023-02-07] (2013)

[23] Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: 24th {USENIX} Security Symposium ({USENIX} Security 15), pp. 129–144 (2015)

[24] Naumenko, G.: ASN-based bucketing of the network nodes · Issue 16599 · bitcoin/bitcoin. https://github.com/bitcoin/bitcoin/issues/16599. [Online; accessed 2023-02-07] (2019)

[25] Naumenko, G.: TxWithhold Smart Contracts | BitMEX Blog. https://blog.bitmex.com/txwithhold-smart-contracts/. [Online; accessed 2023-02-07] (2022)

[26] Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, pp. 281–310 (2015). Springer

[27] Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. Communications of the ACM **61**(7), 95–102 (2018)