A Practical Approach to Analyzing Bitcoin Payment Channels

Joël Gugger[1]

HES-SO Master, Lausanne, Switzerland
joel.gugger@master.hes-so.ch

Abstract. Payment channels or micropayment channels, as mentioned previously, are one part of the scalability solution. The idea of payment channels was suggested by Satoshi in an email to Mike Hearn. Since then, various schemes to construct such structures have been proposed. To have a better understanding of the differences between various channel schemes, and to be able to analyze a channel scheme objectively, a few formal definitions are needed. A list of formal definitions for payment channel construction is proposed. An analysis of different commonly exposed payment channel constructions is done following these definitions. The list does not contain all the payment channel schemes, some of them might be missing. However, the list contains a fairly good representation of the different existing constructions.

Keywords: Payment channels, State channels, Bitcoin

1 Introduction

2 Formal definitions

These formal definitions specify the necessary and sufficient conditions for a payment channel to be qualified as a member of a specific set. They set boundaries or limits that separate the term from any other term. The following formal definitions qualify properties that micropayment channels based on a blockchain such as Bitcoin can have with the view of a particular player. Transactions represent a set of information with a special meaning for the given blockchain that modify the channel state. A transaction can be broadcast to the network to effectively affect the on chain channel state or been kept by the player. Players are users of the given blockchain and they own funds. Funds are owned by one and only one player at a time. The meaning of owning an amount of funds in a channel for a given player is defined as holding a transaction not yet broadcast that allows this player to claim this amount of funds.

A channel is composed of n players $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_n\}$. We denote a channel going from player p_1 to p_2, where $p_1, p_2 \in \mathcal{P}$, with $p_1 \rightarrow p_2$ if p_1 can send money through the channel to p_2. Channels have states and it is possible to move from one state to another with $\mathcal{S}$ steps.

Definition 1 (Trustless). *A channel is trustless for a player $p_i \in \mathcal{P}$ if and only if the safety of his funds at each step $\mathcal{S}$ of the protocol does not depend on the behavior of players $\mathcal{P}' = \{\, p' \,|\, p' \neq p_i \,\}$.*

Definition 2 (Optimal). *A channel is optimal for a player $p_i \in \mathcal{P}$ if and only if the number of transactions $\mathcal{T}(\mathcal{C})$ needed to claim the funds for a given constraint $\mathcal{C}$ is equal to the number of moves $\mathcal{M}(\mathcal{C})$ needed to satisfy the constraint at any time.*

For example, for a refund constraint $\mathcal{C}$ in a channel $p_1 \rightarrow p_2$, refunding p_1 requires $\mathcal{M}(\mathcal{C}) = 1$, thus an optimal scheme requires $\mathcal{T}(\mathcal{C}) = \mathcal{M}(\mathcal{C}) = 1$.

Definition 3 (Open-ended). *A channel is open-ended for a player $p_i \in \mathcal{P}$ if and only if there is no predetermined channel lifetime at the setup.*

A channel that is not open-ended can have a refresh mechanism on-chain with a designated transaction before the end of the lifetime.

Definition 4 (Undelayed). *A channel is undelayed for a player $p_i \in \mathcal{P}$ if and only if this player can broadcast his set of transactions at any time.*

Definition 5 (Non-interactive). *A channel is non-interactive for a player $p_i \in \mathcal{P}$ if and only if this player does not have the responsibility to watch the targeted blockchain to react on arbitrary events $\mathcal{E}$ in order to guarantee his safety.*

With these five definitions, it is possible to infer a significant necessary corollary. If a channel is undelayed for a player this player can broadcast his latest state without constraint, and if this channel is also optimal for the same player only one transaction is needed to move the funds. If only one transaction is needed to move the funds, then the funds are directly available for this player. If the funds are available instantly, then the channel is instantaneous for the player.

Corollary 1 (Instantaneous). *A channel is instantaneous for a player $p_i \in \mathcal{P}$ if and only if the channel is undelayed and optimal for this player.*

2.1 Types of payment channels

We can distinguish two type of channels, unidirectional channels that allow one user to send money to another user and bidirectional channels that allow two users to send in either direction. Usually, a bidirectional channel is more optimal than two unidirectional channels but introduces other constraints.

Unidirectional In a two-player unidirectional channel, there is a payer, later referred to as player-one or client, and a payee, later referred to as player-two or provider. It is not possible to transfer money back in the reverse direction in the channel. These channels are asymmetric, each player benefits from different channel properties. The analysis must be done in the view of each player $p_i \in \mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_n\}$ at a time.

Bidirectional In a two-player bidirectional channel $\mathcal{C}$, the player A and the player B can send funds in direction $\mathcal{C}_{AB}$ and $\mathcal{C}_{BA}$. A bidirectional channel can be a specific scheme or a pairing of existing unidirectional channels. These channels are generaly symmetric, each player $p_i \in \mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_n\}$ benefits from the same channel properties.

3 Analysis of payment channels

3.1 Spilman-style payment channels

Spilman-style payment channels, proposed by Jeremy Spilman in 2013 [1], are the most simple construction of a unidirectional payment channel. They have a finite lifetime predefined at the setup phase and the client, i.e., the payer, cannot trigger their refund before the end of the channel lifetime (but he can receive his funds back if the payee settles the channel before the end of the lifetime.) The channel is one-time use. When the payer or the payee get their funds, the channel is closing. Neither the payer nor the payee need to watch the blockchain to react to events during the lifetime of the channel because only the payee can broadcast a transaction, so both do not need to watch the blockchain to be safe. It is worth noting that, without a proper fix to transaction malleability [2, 3, 4, 5], this scheme is not secure.

Player	Trustless	Optimal	Open-ended	Undelayed	Non-interactive
Payer	Yes	Yes	No	No	Yes
Payee	Yes	Yes	No	Yes	Yes

Table 1. Summary of Spilman-style payment channel properties

According to the previous definitions, Spilman-style ephemeral payment channels are instantaneous non-interactive channels for the payee, and optimal non-interactive for the payer.

3.2 `CLTV`-style payment channels

Introduced in 2015, `CLTV`-style payment channels are a solution to the malleability problem in Spilman-style payment channels. With the new `OP_CODE` check locktime verify (`OP_CHECKLOCKTIMEVERIFY`), redefining the `OP_NOP2`, it is possible to enforce the non-spending of a transaction output until some time in the future. With `OP_CHECKLOCKTIMEVERIFY` a transaction output can enforce the spending transaction to have a `nLockTime` later or equal to the specified value in the script [6].

Instead of creating a funding transaction and a refund transaction vulnerable to transaction malleability attacks, the client creates the funding transaction

```
IF
    <provider pubkey> CHECKSIGVERIFY
ELSE
    <expiry time> CHECKLOCKTIMEVERIFY DROP
ENDIF
<client pubkey> CHECKSIG
```

Listing 1: Locking script (scriptPubKey) with `CHECKLOCKTIMEVERIFY`

output with a script (Listing 1) that allows the provider and the client to spend the funds with co-operation or after a lock time the client can spend the funds without the co-operation of the provider.

`CLTV`-style payment channels have the same properties as Spilman-style payment channels following the previous definitions but are not subject to transaction malleability attacks.

3.3 Decker-Wattenhofer duplex payment channels

Decker-Wattenhofer duplex payment channels [7], also called Duplex Micropayment Channels (DMC), proposed in 2015, are bidirectional channels based on pairs of Spilman-style unidirectional channels. The construction has a finite lifetime predefined at the setup phase but can be refreshed on-chain to keep the channel open with an updated state. During the refresh process, it is possible to refill the channel, and the scheme allows payment routing with Hashed Timelock Contracts (HTLC).

DMC payment channels are not optimal. Uncooperative closing of the channel requires $d + 2$ transactions (where d is equal to the revocation tree depth). They are not undelayed, without other players cooperation the funds are recovered after `nLockTime` values. DMC are not open-ended, a dedicated transaction needs to be broadcast before the end of the `nLockTime`.

Trustless	Optimal	Open-ended	Undelayed	Non-interactive
Yes	No	(Yes)	No	No

Table 2. Summary of Decker-Wattenhofer duplex payment channel properties

3.4 Poon-Dryja payment channels

Poon-Dryja payment channels, also called Lightning Network, is a proposed implementation of HTLC with bidirectional payment channels which allow payments to be securely routed across multiple peer-to-peer payment channels [8].

Their scheme is trustless (assuming that Segregated Witness (SegWit) has been implemented), open-ended, and undelayed but not optimal when the channel closes without co-operation nor non-interactive.

Trustless	Optimal	Open-ended	Undelayed	Non-interactive
Yes	No	Yes	Yes	No

Table 3. Summary of Poon-Dryja payment channel properties

3.5 Summary

Channel	Type	Optimal	Open-ended	Undelayed	Non-inter.
Spilman-style	Uni	Yes/Yes	No/No	No/Yes	Yes/Yes
CLTV-style	Uni	Yes/Yes	No/No	No/Yes	Yes/Yes
Decker-Wattenhofer DMC	Bi	No	(Yes)	No	No
Poon-Dryja	Bi	No	Yes	Yes	No
Shababi-Gugger-Lebrecht	Uni	No/Yes	Yes/Yes	No/Yes	Yes/No

Table 4. Summary of different payment channels

This table summarizes the different properties of the proposed definitions of common channel schemes. The last row refers to the next presented scheme.

4 One-way channel (Shababi-Gugger-Lebrecht)

Our one-way payment channel for Bitcoin is a modified version of other layer-two applications, such as “Yours Lightning Protocol” or Lightning Network [8, 9]. The scheme is specially designed for a client to provider scenario, where the provider has multiple clients through multiple channels. The core design aims to be as cheap as possible for the provider while being flexible for settlement. The white paper “Partially Non-Interactive and Instantaneous One-way Payment Channel for Bitcoin” inserted after the appendices, describes the core design and the incentives.

Player	Trustless	Optimal	Open-ended	Undelayed	Non-interactive
Payer	Yes	No	Yes	No	Yes
Payee	Yes	Yes	Yes	Yes	No

Table 5. Summary of Shababi-Gugger-Lebrecht payment channel properties

A part of this thesis was devoted to writing the white paper describing our channel scheme while working on the scheme itself. During this work we found a possible attack described in the white paper which we fixed.

The next step has been to analyze how it is possible to optimize the channel with threshold cryptography. As it is possible to see, every channel construction depends on a funding transaction that locks funds in a 2-out-of-2 multi-signature script. This funding transaction is always on-chain, so if it is possible to replace this Pay To Script Hash (P2SH) with a standard Pay To Public Key Hash (P2PKH) output the savings should be attractive.

References

[1] Jeremy Spilman. *[Bitcoin-development] Anti DoS for tx replacement.* 2013. URL: `https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-April/002433.html` (visited on 02/04/2018).

[2] Eric Lombrozo, Johnson Lau, and Pieter Wuille. *Segregated Witness.* URL: `https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki` (visited on 02/02/2018).

[3] Pieter Wuille. *Dealing with malleability.* 2014. URL: `https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki` (visited on 02/04/2018).

[4] Marcin Andrychowicz et al. "How to deal with malleability of BitCoin transactions". In: *CoRR* abs/1312.3230 (2013). arXiv: `1312.3230`. URL: `http://arxiv.org/abs/1312.3230`.

[5] Christian Decker and Roger Wattenhofer. "Bitcoin Transaction Malleability and MtGox". In: *CoRR* abs/1403.6676 (2014). arXiv: `1403.6676`. URL: `http://arxiv.org/abs/1403.6676`.

[6] Peter Todd. *CHECKLOCKTIMEVERIFY.* 2014. URL: `https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki` (visited on 02/05/2018).

[7] Christian Decker and Roger Wattenhofer. "A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels". In: *17th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), Edmonton, Canada.* Aug. 2015.

[8] Joseph Poon and Thaddeus Dryja. "The bitcoin lightning network: Scalable off-chain instant payments". In: *draft version 0.5* 9 (2016).

[9] Ryan X. Charles and Clemens Ley. *Yours Lightning Protocol.* 2016. URL: `https://github.com/yoursnetwork/yours-channels/blob/master/docs/yours-lightning.md`.