```
  BIP: 197
  Layer: Applications
  Title: Hashed Time-Locked Collateral Contract
  Author: Matthew Black <matthew@atomicloans.io>
          Tony Cai <tony@atomicloans.io>
  Comments-Summary: No comments yet.
  Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0197
  Status: Draft
  Type: Standards Track
  Created: 2019-03-19
  License: BSD-3-Clause
           CC0-1.0
```

Abstract

This BIP describes a script for generalized debt agreement contract based on Hashed Time-Lock Contract (BIP 199) transactions according to the Atomic Loans specification (https://arxiv.org/pdf/1901.05117.pdf). For more details visit https://atomicloans.io.

Summary

A Hashed Time-Locked Collateral Contract (HTLCC) consists of two scripts that permit a designated party (the "borrower") to lock funds on the Bitcoin chain for a specified amount of time as collateral in a debt agreement where the loan principal is denominated in a currency on another blockchain. We denote the blockchain on which the loan principal is issued the principal blockchain.

The purpose of each script is to enable the creation of a debt agreement between two parties (the "borrower" and the "lender"), where the collateral is locked in a P2SH, and can only be spent once the borrower repays the principal and interest in the debt agreement on the principal blockchain. In the case that the borrower does not repay, the borrower or lender can opt for liquidation of the collateral, which will involve the atomic swapping of collateral for the loan currency. In the case that at least one of the two parties don't opt for liquidation, then each party will be entitled to a percentage of the collateral, decided when the funds are initially locked in the P2SH.

These funds are locked into two scripts. Refundable Collateral and Seizable Collateral scripts. The funds sent to these scripts represent the percentage of collateral that each party is entitled to in the case that repayment fails, and the parties don't opt for liquidation.

The Refundable Collateral script takes the following form:

```
   OP_IF
OP_SIZE  OP_EQUALVERIFY [HASHOP]  OP_EQUALVERIFY OP_DUP OP_HASH160  OP_EQUALVERIFY OP_CHECKS
OP_ELSE
```

```
OP_IF
[TIMEOUTOP] OP_DROP OP_SIZE OP_PUSHDATA(1)  OP_EQUALVERIFY [HASHOP]  OP_EQUALVERIFY OP_SIZE
OP_ELSE
[TIMEOUTOP] OP_DROP OP_DUP OP_HASH160  OP_EQUALVERIFY OP_CHECKSIG
OP_ENDIF
OP_ENDIF
```

The Seizable Collateral script takes the following form:

```
   OP_IF
OP_SIZE  OP_EQUALVERIFY [HASHOP]  OP_EQUALVERIFY OP_DUP OP_HASH160  OP_EQUALVERIFY OP_CHECKS
OP_ELSE
OP_IF
[TIMEOUTOP] OP_DROP OP_SIZE  OP_EQUALVERIFY [HASHOP]  OP_EQUALVERIFY OP_SIZE  OP_EQUALVERIFY
OP_ELSE
OP_IF
[TIMEOUTOP] OP_DROP OP_SIZE  OP_EQUALVERIFY [HASHOP]  OP_EQUALVERIFY OP_DUP OP_HASH160  OP_
OP_ELSE
[TIMEOUTOP] OP_DROP OP_DUP OP_HASH160  OP_EQUALVERIFY OP_CHECKSIG
OP_ENDIF
OP_ENDIF
OP_ENDIF
```

[HASHOP] is either OP_SHA256 or OP_HASH160.

[TIMEOUTOP] is either OP_CHECKSEQUENCEVERIFY or OP_CHECKLOCKTIMEVERIFY.

Interaction

- Alice (the "borrower") and Bob (the "lender") exchange public keys as well as two secret hashes A1, A2 created by Alice and three hashes B1, B2, B3 created by Bob. They then mutually agree upon a timeout threshold for the Loan Period, Liquidation Period, and Seizure Period. Alice constructs the script and P2SH address for the Refundable Collateral Contract and Seizable Collateral Contract. Bob constructs the script for the blockchain on which the loan principal will be issued - the principal blockchain.
- Bob sends loan principal funds to the loan script on the principal blockchain
- Alice sends funds to the Refundable Collateral P2SH address and the Seizable Collateral P2SH address. The amount of funds she sends to the two addresses will be determined beforehand off-chain between Alice and Bob.
- Either
  - Bob accepts locking of collateral by Alice and reveals B1, allowing Alice to withdraw the loan amount on the principal blockchain.
  - Bob doesn't accept locking of collateral by Alice, and recovers the funds after the approve expiration while revealing B2, which allows

Alice to refund the Refundable and Seizable collateral.

- – If Bob accepts the locking of collateral by Alice
- – Either
    - Alice repays the loan by the end of the Loan Period and Bob reveals the secret to Alice by revealing it in the loan repayment acceptance transaction; OR
    - Alice defaults on the loan and Alice and Bob both opt for collateral liquidation, where any third-party is able to bid on the collateral. The winning bidder, Charlie, will subsequently receive the liquidated collateral by way of an Atomic Swap between the collateral funds (ie. BTC locked in both the Refundable Collateral P2SH and the Seizable Collateral P2SH) and the bid funds (ie. funds denominated in the loan currency, put forth by Charlie as part of his bid). This is done by both Alice and Bob signing a multisig and revealing A2 and B2; OR
    - Alice defaults on the loan and at least one of Alice or Bob opts out of collateral liquidation, then Alice recovers the Refundable Collateral funds and Bob spends the Seizable Collateral funds.
    - Alice defaults on the loan and at least one of Alice or Bob opts out of collateral liquidation. But Bob doesn't spend the Seizable Collateral funds, so Alice recovers both the Refundable Collateral funds and the Seizable Collateral funds.

Compatibility

BIP 197 is compatible with [ERC 1850](https://github.com/ethereum/EIPs/pull/1850) for [atomic loans](https://arxiv.org/pdf/1901.05117.pdf) with Ethereum. Can be extended in the future to be compatible with other HTLC and smart contract compatible chains.

Motivation

In many different protocols, the revealing of secrets is used as a settlement mechanism. HTLCC transactions are a safe way of exchanging secrets to advance the state of a debt agreement, due to the ability to recover a percentage of collateral funds from an uncooperative counterparty, and ensure principal + interest + liquidation fee is paid with a cooperative party.

Definitions

borrower: entity that locks collateral on the Bitcoin chain and receives loan amount on principal blockchain from lender following the approval of the borrower’s borrow request

lender: entity that contributes funds to the Hashed Time-Locked Principal

Contract (HTLPC) on the principal blockchain, to be borrowed by the borrower upon the locking of collateral on the Bitcoin chain and the lender's approval

repay: when the borrower pays back the principal + interest before loanExpiration

default: when the borrower fails to pay back the principal + interest before the loanExpiration

secret: random number chosen by the borrower or lender, revealed to allow the parties to change the state of the debt agreement

secretHash: hash of the secret, used in the construction of HTLCC

SecretA1: secret generated by the borrower, used to prove that the borrower has withdrawn the loan

SecretA2: secret generated by the borrower, used to allow the bidder to withdraw the liquidated collateral funds

SecretB1: secret generated by the lender, used to accept the locking of collateral by borrower, enabling borrower to withdraw the loan amount

SecretB2: secret generated by the lender, used to refund themselves in the event they aren't satisfied with borrower'slocking of collateral. Also used to accept borrower's repayment of principal plus interest

SecretB3: secret generated by the lender, used to allow the bidder to withdraw the liquidated collateral funds

SecretC: secret generated by the bidder, used to accept the signatures of the borrower and lender for authorizing the liquidation of collateral

loan expiration num: timestamp before which the borrower must repay the loan; or otherwise risk the liquidation or seizure of their collateral

bidding expiration num: timestamp that determines the amount of time allocated to bidding before seizure period occurs

seizure expiration num: timestamp that determines the amount of time during which the lender can seize funds within the Seizable Collateral P2SH, after which the borrower can refund their corresponding amount of the collateral they are entitled to (ie. either just the funds within the Refundable Collateral P2SH, or both the Refundable Collateral and Seizable Collateral in the event where the lender failed to seize).

Approve Period

During this time, the lender deploys the HTLPC on the principal blockchain. Following this, the borrower locks their collateral on the Bitcoin blockchain in a HTLCC. The lender then either reveals secretB1 to signify that they are satisfied with the collateral, and the borrower can withdraw the loan by revealing secretA1.

If the lender is not satisfied with the collateral locked by the borrower, the lender can refunds their loan amount by revealing secretB2, which will subsequently allow the borrower to refund the collateral amount they deposited.

Loan Period

Once the borrower has withdrawn the loan amount, the Loan Period begins. Once the Loan Period is finished, the borrower is expected to repay the loan. If they do, the lender can then accept the repayment by revealing secretB2, enabling the borrower to refund their collateral amount. In the case that the borrower defaults or does not repay the full principal plus interest amount, the lender can choose to not accept the loan repayment, and the parties can opt for liquidation of the collateral in the Bidding Period.

Bidding Period

In the case of a default or the lender not accepting the borrower repayment, the lender and borrower can opt for liquidation of the collateral through the process of third party bidders bidding on the collateral. The Bidding Period can be initiated by either the lender or the borrower. Once the bidding timeout occurs, the lender and borrower must each provide a signature, followed by secretC revealed by the winning bidder once they have checked that the signature is proper. Finally, the lender and borrower must each reveal secretA2 and secretB3 to allow the collateral to be withdrawn by the winning bidder.

Seizure Period

In the case that either the lender or borrower don't accept the bid, the lender can seize a percentage of the collateral. The amount is dependent on the amount of collateral locked in the Seizable Collateral and Refundable Collateral script as described in this BIP. During this period, the borrower can also refund the funds locked in the Refundable Collateral script.

Refund Period

In the case that the lender does not seize the collateral locked in the Seizable Collateral script, then the borrower can refund the funds locked in the Seizable Collateral script.

Rationale

The rational for the following script checking the length of secrets pushed to the stack that are used with OP_SHA256 in the following script

```
 OP_SIZE  OP_EQUALVERIFY
```

is to ensure that the secret size is exactly a certain number of bytes long.

This is especially important when this script is used alongside the HTLPC on other chains like Ethereum where the sha256 opcode only takes up 32 bytes and disregards the rest, there is a need to ensure that the length on the Bitcoin side is 32 bytes.

Backwards Compatibility

As this is a new standard for collateralized debt, there is no need for backward compatibility. Once this is accepted as a standard there are certain aspects of the contract that can be modified while still retaining backwards compatibility, such as removing the need to verify the size of the hash if being used with two blockchains with the same maximum block size, which would be backward compatible with the current script.

Implementation

https://github.com/AtomicLoans/chainabstractionlayer/blob/bitcoin-collateral-provider/src/providers/bitcoin/BitcoinCollateralProvider.js

Copyright