Trustless, Interoperable Cryptocurrency-Backed Assets

Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, William J. Knottenbelt
Trustless, Interoperable Cryptocurrency-Backed Assets

Research Paper
(IEEE S&P 2019)
eprint.iacr.org/2018/643

PoC Code
(GPL-3.0)
github.com/crossclaim
This research was co-funded by Blockchain.com, Outlier Ventures, Bridge 1 858561 SESC, Bridge 1 864738 PR4DLT (all FFG), the Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), and the competence center SBA-K1 funded by COMET.
Motivation

Today:
Over 2000 heterogeneous cryptocurrencies

Different Properties

- Privacy
- Scalability
- Security
- Expressiveness
- Transparency
- Consensus
- Finality

Challenge:
Trustless and scalable cross-chain communication
A History of Theft and Loss

Technology

Bitcoin Price Plunges as Mt. Gox Exchange Halts Activity
Carter Dougherty
February 7, 2014, 8:25 PM GMT

Bitcoin plunged more than 8 percent today after a Tokyo-based exchanges halted withdrawals of the digital currency, citing technical difficulties.

Poloniex Users Suffering From Frozen Accounts, Suspended Withdrawals, and Disabled Markets

Bitcoin exchange BitFloor shuttered after virtual heist

Nearly a quarter million dollars worth of the peer-to-peer currency was stolen by accessing unencrypted backup wallet keys.

The DAO Attacked: Code Issue Leads to $60 Million Ether Theft

Bitcoin Worth $72M Was Stolen in Bitfinex Exchange Hack in Hong Kong

Bitstamp exchange hacked, $55M worth of bitcoin stolen

CoinCheck Hack: Bitcoin Exchange Security Under Scrutiny After $534M Cryptocurrency Theft

Isolated Partnership Makes the Most of a Great British Summer

Related Stories

Studying the Sources of Bitcoin and Crypto on Twitter
A History of Theft and Loss

Decentralized Exchanges?
Cross-Chain Communication Today

Centralized exchanges (CeX)
- Predominant method to exchange assets cross-chain
- > 99% of volume

Decentralized Exchanges (DeX):
- < 1% of volume
- Mostly limited to ERC20 tokens on Ethereum
  → Not „Cross-chain“!
Atomic Cross-Chain Swaps* (2012)

• Ensure $A \rightarrow B$ and $A \leftarrow B$ occur atomically
• Hashed Time-Lock Contracts (HTLCs)

Challenges:

- All parties must be online
- Need out-of-band channel (censoring!)
- Require monitoring of all involved chains
- No standardized interface for locks
- Race conditions, mempool sniffing, …

*we refer to the HTLC-based form of ACCS. Other constructions possible
Sidenote: HTCL ACCS

s \leftarrow \text{random()}

\textbf{Tx1} \leftarrow \text{if} \left[ \left( H(\text{input}) = H(s) \land \exists \text{sig}(A) \right) \lor \exists \text{multiSig}(A|B) \right] : A_{\text{BTC}} \rightarrow B_{\text{BTC}}

\textbf{Tx2} \leftarrow \text{if} \left[ t_{\text{current}} > t_1 \land \exists \text{multiSig}(A|B) \right] : B_{\text{BTC}} \rightarrow A_{\text{BTC}} \text{ spending } \textbf{Tx1}(!)

\textbf{sign}(sk_B, \textbf{Tx2})

\textbf{Tx3} \leftarrow \text{if} \left[ \left( H(\text{input}) = H(s) \land \exists \text{sig}(A) \right) \lor \exists \text{multiSig}(A|B) \right] : B_{\text{LTC}} \rightarrow A_{\text{LTC}}

\textbf{Tx4} \leftarrow \text{if} \left[ t_{\text{current}} > t_2 \land \exists \text{multiSig}(A|B) \right] : B_{\text{BTC}} \rightarrow A_{\text{BTC}} \text{ spending } \textbf{Tx3}(!)

\textbf{sign}(sk_A, \textbf{Tx4})

\textbf{Tx5} (spends } \textbf{Tx3} \text{ revealing } s

\textbf{Tx6} (spends } \textbf{Tx1} \text{ using } s

\textbf{time}

Legend:

- Published to LTC
- Published to BTC
- Unpublished / Sent directly to user
Cryptocurrency-Backed Assets

On-chain assets backed 1:1 by an existing cryptocurrency
e.g. **Bitcoin-backed tokens** on Ethereum

- Cross-chain DeX
- Cross-chain payment channels
- Improved atomic swaps
- Stablecoins
- …
Challenge: Conditional Locks in Bitcoin

Goal:
Unlock funds on Bitcoin only when tokens are burned

Challenge:
We cannot verify the state of e.g. Ethereum

Can we use hashlocks?
Publicly verifiable contracts cannot generate random secret

→ We need an intermediary
**System Model**

**Requester**: locks coins to issue tokens

**Redeemer**: burns tokens to receive coins

**Sender/Receiver**: Send/receive backed tokens

**Vault**: ensures correct redeeming on backing chain.  
*Non-trusted and collateralized*

**Smart Contract**: responsible for issuing, trading and redeeming on issuing chain.  
Enforces correctness of Vaults.

**Intermediaries**
Smart Contract

Base functionality:
• Issue
• Transfer / Swap
• Redeem

Chain Relay:
• Verify PoW
• Verify TX inclusion proof

Collateralization:
• Lock
• Conditional release / Liquidate
Chain Relay

Cross-chain SPV / light client
E.g. deployed on Ethereum to verify transactions in Bitcoin

Block Headers

Transaction +
Merkle Path

h7 = H(h5, h6)

h5 = H(h1, h2)

h6 = H(h3, h4)

LOCK TX

h2

h3

h4
# System Requirements

<table>
<thead>
<tr>
<th>Backing Chain</th>
<th>Issuing Chain (Smart Contracts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Chain relays</td>
</tr>
<tr>
<td>(Basic ledger functionality)</td>
<td>• Verify PoW of backing chain</td>
</tr>
<tr>
<td></td>
<td>• Verify transaction inclusion</td>
</tr>
<tr>
<td></td>
<td>On-chain assets / meta information</td>
</tr>
<tr>
<td></td>
<td>• Tokens, colored coins, …</td>
</tr>
<tr>
<td></td>
<td>Conditional payments</td>
</tr>
<tr>
<td></td>
<td>• Collateralization</td>
</tr>
<tr>
<td>e.g. Bitcoin, Ethereum, Ethereum Classic, Litecoin, …</td>
<td>e.g. Ethereum, Ethereum Classic, Zilliqa, Cardano?, …</td>
</tr>
</tbody>
</table>
## System Requirements

<table>
<thead>
<tr>
<th>Backing Chain</th>
<th>Issuing Chain (Smart Contracts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Basic ledger functionality)</td>
<td>Chain relays</td>
</tr>
<tr>
<td></td>
<td>• Verify PoW of backing chain</td>
</tr>
<tr>
<td></td>
<td>• Verify transaction inclusion</td>
</tr>
<tr>
<td>Smart contracts allow to automate/optimize the process</td>
<td>On-chain assets / meta information</td>
</tr>
<tr>
<td></td>
<td>• Tokens, colored coins, ...</td>
</tr>
<tr>
<td>e.g. <strong>Bitcoin</strong>, Ethereum, Ethereum Classic, Litecoin, ...</td>
<td>Conditional payments</td>
</tr>
<tr>
<td></td>
<td>• Collateralization</td>
</tr>
<tr>
<td>e.g. <strong>Ethereum</strong>, Ethereum Classic, Zilliqa, Cardano?, ...</td>
<td></td>
</tr>
</tbody>
</table>
Protocols
Issue

**Bitcoin**

Alice

Vault

**Ethereum**

BTC RELAY

Alice

- Ethereum transaction
- Bitcoin transaction
- Off-chain/other interaction
Issue: Precondition

→ Over-collateralization to mitigate exchange rate fluctuations
Issue
1) Lock

2a) Prove

Ethereum transaction

Bitcoin transaction

Off-chain/other interaction
**Issue**

The diagram illustrates the process of issuing BTC on the Ethereum blockchain through a vault service. The process involves:

1. **Lock**: Alice sends Bitcoin to the vault.
2. **Prove**: The vault verifies and confirms the transaction.
3. **Issue**: The vault issues Ether on the Ethereum blockchain (same TX).

The diagram uses arrows to indicate the flow of transactions and interactions:

- **Ethereum transaction**: Solid blue arrows.
- **Bitcoin transaction**: Orange arrows.
- **Off-chain/other interaction**: Dashed black arrows.

The diagram visualizes the interaction between Alice, the vault, and the Ethereum network, highlighting the key steps in the process.
Only issue if Issuer locked sufficient collateral!

→ **Challenge: race conditions**
Issue – Race Conditions

Potential Problems:

• **Simultaneous issuing**
  - Alice and Carol try to lock same portion of the vault’s collateral
  - Loser of the race looses BTC

• **Vault withdraws collateral before Alice can finalize process**
  - Security waiting period for inclusion proof
  - Ethereum transaction inclusion time
  - Latency
  - DoS
Mitigation 1 – Delayed Collateral Withdraw

Issuer must announce withdrawal of unused collateral:

1) **Announce**

2) **Delay**
   - finalize pending requests
   - users know race conditions are now possible

3) **Withdraw**
Mitigation 2 – Collateralized Commitments

Alice registers **issue commitment** in smart contract
→ Temporarily locks vault’s *eth* collateral

Requirement: Alice must provide collateral to **prevent griefing**
Swap & Transfer…

Simple ERC20 transfer / atomic swap!
Alice → Bob
Redeem
Redeem

1. Lock / Burn
2. Signal to "unlock btc"
3. Observe / Verify

Bitcoin

Ethereum

Bob

BTC RELAY

Ethereum transaction
Bitcoin transaction
Off-chain/other interaction
Redeem

1) Lock / Burn

2) Signal to "unlock btc"

3) Observe / Verify

4) Release btc

Ethereum

BTC

RELAY

Ethereum transaction

Bitcoin transaction

Off-chain/other interaction
If the vault cannot provide proof of correct behavior:
- **Collateral slashed**
- **Bob reimbursed**
# Mitigating Exchange Rate Fluctuations

<table>
<thead>
<tr>
<th>Stage</th>
<th>Meaning</th>
<th>Action</th>
<th>Example threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Operation</td>
<td>Collateral surplus</td>
<td><strong>Vault</strong>: Withdrawal of unused collateral possible. <strong>Users</strong>: can issue new assets</td>
<td>&gt; 2.0</td>
</tr>
<tr>
<td>Buffered Collateral</td>
<td>Sufficient collateral buffer</td>
<td><strong>SC</strong>: no new Issue requests accepted <strong>Vault</strong>: Increase collateral.</td>
<td></td>
</tr>
<tr>
<td>Liquidation</td>
<td>Collateral buffer critically low</td>
<td><strong>Vault</strong>: increase collateral <strong>Users</strong>: redeem recommended <strong>SC</strong>: automatic liquidation (opt-in/out)*</td>
<td>&lt; 1.05</td>
</tr>
</tbody>
</table>

* Triggered by exchange rate oracle or user/watchtower
System Properties

1. **Auditability**: all actions on both chains logged

2. **Consistency**: backed-assets only issued if proof provided

3. **Redeemability**: receive Bitcoin or be reimbursed in Ether

4. **Liveness**: no third party required to use XCLAIM. *Any user can become a vault!!*

5. **Atomic Swaps**: swap Bitcoin vs Ether via smart contract

6. **Scale-out**: the more vaults / collateral locked, the more assets can be issued

7. **Compatibility**: minimal requirements for backing chain
Implementation

- XCLAIM smart contract: Solidity v0.5.x (~ 820 LOC)

- BTCRelay: Serpent ([https://github.com/ethereum/btcrelay](https://github.com/ethereum/btcrelay)) → new Solidity implementation is WIP

- Tested on Ropsten

[https://github.com/crossclaim](https://github.com/crossclaim)
Performance and Costs

Exchange rate: USD 220 / ETH (Gas cost: 5 gwei); USD 4,497 / BTC
“Recommended” security parameters: 14 sec x 12 ETH Tx confs; 10 min x 6 BTC Tx confs.
Comparison to HTLC Atomic Swaps

BTC-ETH swaps with XCLAIM are 95.7% faster and 64.5% cheaper for 1000 independent swaps.
Challenges and Ongoing Work

Feasibility of chain relays

- Off-chain verification games: TrueBit, Arbitrum, ...
- Compact proofs: NiPoPoWs, FlyClient
- Combination: Game + Fallback NIZK Proof
  $\rightarrow$ PoW verification (hash preimage $\rightarrow$ hash?)

Multi-signatures to prevent theft
(feasible via off-chain channels)

Incentives for Issuer F(r)ee Market

Decentralized Exchange Rate Oracles & Stabilization
Questions?

Trustless, Interoperable Cryptocurrency-Backed Assets

Research Paper
(IEEE S&P 2019)
eprint.iacr.org/2018/643

PoC Code
/GPL-3.0

github.com/crossclaim

Website: xclaim.io