Trustless, Interoperable Cryptocurrency-Backed Assets
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

Challenge:
Trustless and scalable cross-chain communication

Different Properties
- Privacy
- Scalability
- Security
- Expressiveness
- Transparency
- Consensus
- Finality
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 exchange halted withdrawals of the digital currency, citing technical problems.

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

Bitstamp exchange hacked, $5M worth of bitcoin stolen

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.

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

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

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

\[ h_7 = H(h_5, h_6) \]

\[ h_5 = H(h_1, h_2) \]

\[ h_6 = H(h_3, h_4) \]
## System Requirements

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

<table>
<thead>
<tr>
<th>Backing Chain</th>
<th>Issuing Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>None</strong></td>
<td><strong>Issuing Chain (Smart Contracts)</strong></td>
</tr>
<tr>
<td>(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></td>
<td><strong>On-chain assets / meta information</strong></td>
</tr>
<tr>
<td></td>
<td>- Tokens, colored coins, ....</td>
</tr>
<tr>
<td></td>
<td><strong>Conditional payments</strong></td>
</tr>
<tr>
<td></td>
<td>- Collateralization</td>
</tr>
<tr>
<td></td>
<td><strong>e.g. Bitcoin, Ethereum, Ethereum Classic, Litecoin, ...</strong></td>
</tr>
<tr>
<td></td>
<td><strong>e.g. Ethereum, Ethereum Classic, Zilliqa, Cardano?, ...</strong></td>
</tr>
</tbody>
</table>

Smart contracts allow to automate/optimize the process
Protocols
Issue

**Bitcoin**

- Alice
- vault

**Ethereum**

- Alice
- BTC RELAY

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

→ Over-collateralization to mitigate exchange rate fluctuations
Issue

1) Lock

Bitcoin

Ethereum

Alice

BTC RELAY

Ethereum transaction

Bitcoin transaction

Off-chain/other interaction
Issue
Issue
Only issue if Issuer locked sufficient collateral!
→ Challenge: 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**
Simple ERC20 transfer / atomic swap!
Alice → Bob
Redeem

Vault

Bitcoin

Bob

Ethereum

Bob

BTC RELAY

BTC RELAY

Ethereum transaction

Bitcoin transaction

Off-chain/other interaction

1) Lock / Burn

2) Signal to "unlock btc"
Redeem

3) Observe / Verify

1) Lock / Burn

2) Signal to "unlock btc"

Bob

Ethereum

BTC RELAY

Dashed lines:
- Ethereum transaction
- Bitcoin transaction
- Off-chain/other interaction
Redeem

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

1) Lock / Burn

2) Signal to unlock btc

3) Observe / Verify

4) Release btc

5a) Prove redeem (Issuer)

5b) Verify & Confirm (same TX)

BTC RELAY

Bob

Vault

Ethereum

Bitcoin

Ethereum transaction

Bitcoin transaction

Off-chain/other interaction
Redeem

4) Release btc

3) Observe / Verify

2) Signal to "unlock btc"

5c) Release collateral (same TX)

5a) Prove redeem (Issuer)

5b) Verify & Confirm (same TX)

1) Lock / Burn

Bob

BTC RELAY

Ethereum

Bitcoin

Vault
Redeem

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
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) → new Solidity implementation is WIP

• Tested on Ropsten
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.
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
  → PoW verification (hash preimage → hash?)

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

Incentives for Vault 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