An Analysis of Atomic Swaps on and between Ethereum Blockchains
Research Project I

Master of System and Network Engineering
Informatics Institute, University of Amsterdam

Peter Bennink
Lennart van Gijtenbeek
Supervisors:
Oskar van Deventer
Maarten Everts

February 5, 2018
Centralized cryptocurrency exchanges

- Transactions via the exchange
- Exchanges hold funds
- SPOF (wallet database)
- Hacks and disappearances
- Untrustworthy trusted third party
Atomic swaps

- More decentralized process
- Less dependence on (centralized) third parties
- Less chance of loss of funds

“A transaction between two parties that does not depend on a third party, for instance a centralized exchange, and either happens in full, or not at all.”
Are there reliable methods for making atomic swaps on and between blockchains?
Ethereum

- TNO Techruption blockchain
- Ethereum Classic (ETC) & Quorum
- Mining process similar to Bitcoin (PoW)
- Smart contracts
Smart contracts

- Programs stored on the blockchain
- Solidity
- Static and open-source code
- Gas costs
- Executed by miners to verify
- Tokens are implemented using smart contracts
ERC-20 Token Standard

- Homogeneity
- Smoother integration with crypto software.
- Requires the implementation of the ERC-20 interface
  - Return total supply
  - Keep track of wallet balances
  - Transfer tokens
  - Allowances
Types of atomic swaps

1. Single-chain token swaps
2. Single-chain coin/token swaps
3. Cross-chain coin swaps
4. Cross-chain token swaps
5. Cross-chain coin/token swaps
1. Single-chain token swap
2. Single-chain token/coin swap

Wallet A

Client

100 EOS

Wallet B

Client

1 ETH
3. Cross-chain coin swap
4. Cross-chain token swap

Wallet A

1 VeChain

Wallet B

Client

Wallet A

30 InPay

Wallet B

Client
5. Cross-chain token/coin swap

Wallet A → 2.5 EOS → Wallet B

Wallet A ← 1 ETC ← Wallet B
Types of atomic swaps

1. Single-chain token swaps
2. Single-chain coin/token swaps
3. Cross-chain coin swaps
4. Cross-chain token swaps
5. Cross-chain coin/token swaps
6. *Single-chain coin swaps*
Design & implementation

- Single usage swap contracts
- Transaction via the contract
- Rinkeby & Ropsten test networks
- Hashed TimeLocked Contract (HTLC)
- Compatible with all ERC-20 tokens
Cross-chain coin swap
Why is this atomic?

- If client A does not claim their funds, client B cannot either.
- If client A claims, they reveal the secret to client B, who can then also claim their funds.

```solidity
function claim(string _secret) public returns (bool) {
    if (hashed_secret == sha256(_secret) &&
        now < timeOut){
        selfdestruct(clientB);
    } else {
        return false;
    }
}
```
Single-chain token swap
Why is this atomic?

- When claim() is called the contract is in full control of the funds

```solidity
function claim() onlyParticipant public returns (bool) {
    uint token1_balance = token1_instance.balanceOf(this);
    uint token2_balance = token2_instance.balanceOf(this);
    if (token2_balance >= amountOf_token2 &&
        token1_balance >= amountOf_token1 && now < timeOut) {
        token1_instance.transfer(clientB, token1_balance);
        token2_instance.transfer(clientA, token2_balance);
        selfdestruct(clientA);
    } else {
        return false;
    }
}
```
Reusable contracts

- Indefinitely and concurrently
- Scales better
- No deployment costs
Reusable single-chain token swap
Conclusion

- Reliable swaps are possible

Future research

- Reusability
- Other blockchains
- Off-chain
- Decentralized exchanges (use cases)
- Investigate attack vectors

Project git repository:
github.com/clvang000/SNE_TNO_RP1
Questions . . .