BTC RELAY

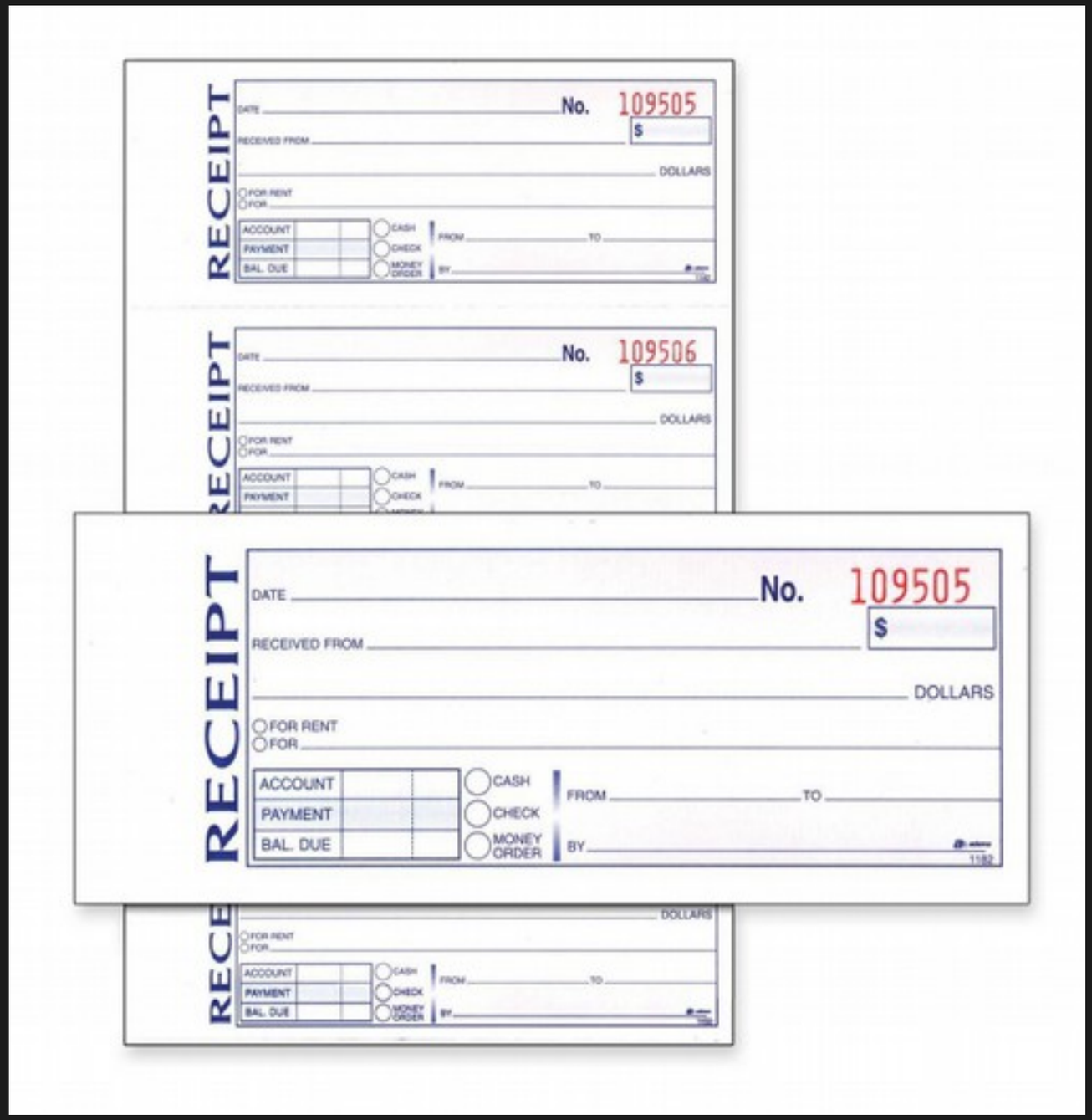
3 secrets for a bridge between the Bitcoin blockchain and Ethereum smart contracts

Joseph Chow

Ð VCON1 - Ethereum Developer's Conference Nov 9th, 2015

btcrelay.org

Secret 1


RECEIPT
DATE
No. 109505
RECEIVED FROM
$
DOLLARS
FOR RENT
FOR
ACCOUNT
PAYMENT
BAL. DUE
CASH
CHECK
MONEY ORDER
FROM
TO
BY
No. 109506
1182

Bridge that empowers

- DApps to accept Bitcoin payments
- DApps to process arbitrary BTC transactions
- Innovation across Bitcoin & Ethereum

Decentralized exchange

Sidechains

Cross-chain payments

Offchain micropayments

BTC RELAY

Advantages

Trustless

Inputs, outputs, cryptography, open

Decentralized

Owned by community, no “admin”

Autonomous

Alive with BTC blockchain

Modular

Easy to integrate, building block

BTC RELAY

Secret 2

Interacting with Bitcoin

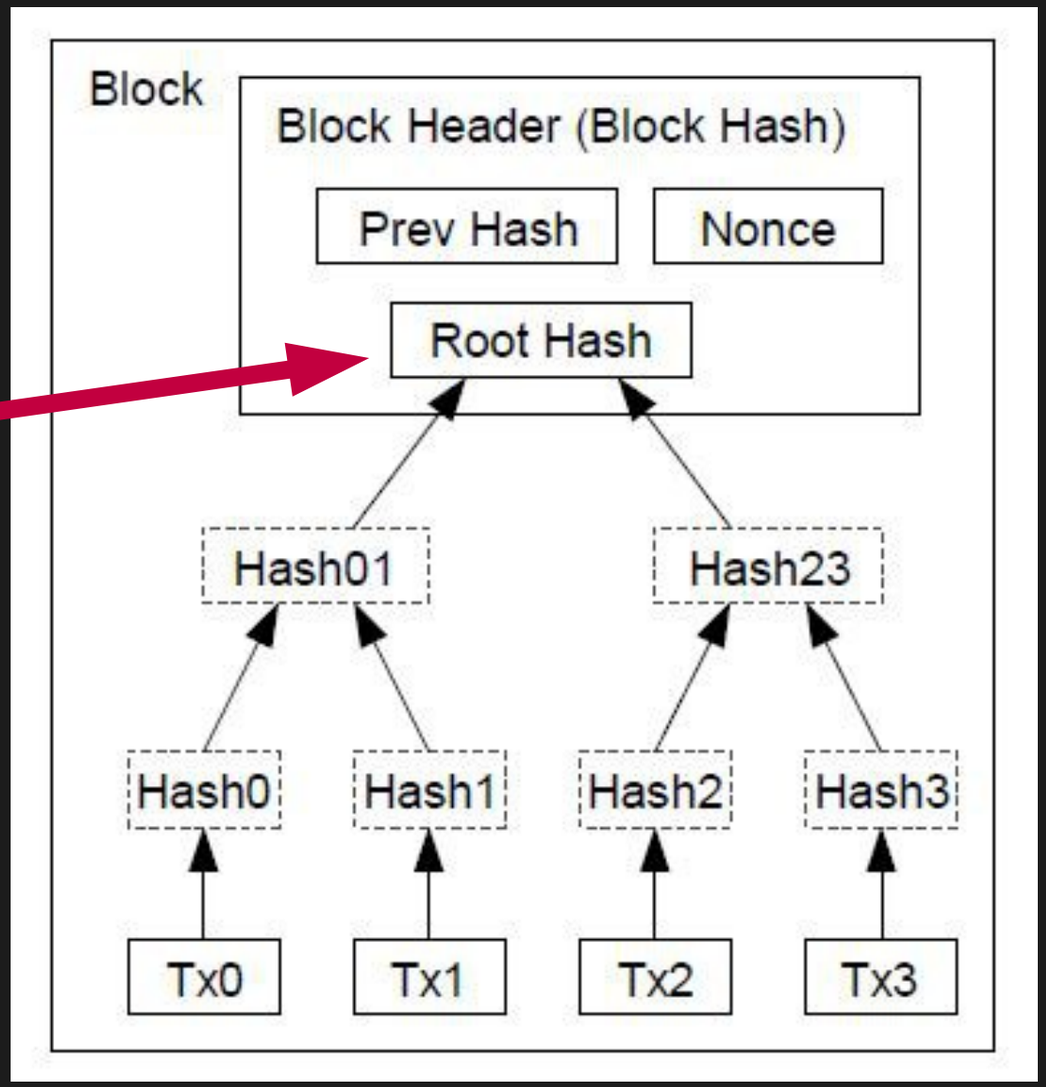
First secret: use proof of payment

Payment proofs need to be verified against something

“Secret”: BTC Relay builds a Bitcoin mini-blockchain a.k.a. SPV-chain

Whirlwind Technical Intro

- Block = header + Tx
- Block header
  - 80 bytes
  - Merkle root can verify Tx
- BTC Relay uses headers to build an SPV-chain



Tx = transaction

Verification APIs

verifyTx (transactionHash, transactionIndex, merkleSibling, blockHash)

relayTx (rawTransaction, transactionHash, transactionIndex, merkleSibling, blockHash, contractAddress)

Two Main Parties

- **Relayers**
  - Of blockheaders
  - Build the Bitcoin SPV-chain
- **Verifiers**
  - Process BTC payments and transactions
  - Contracts such as sidechains
- Can be both **Relayer** and **Verifier**

Secret 3: Incentives for autonomy

- **Relayer** sets fee for block header
- **Verifier** pays fee to **feeRecipient** (e.g. **Relayer** named Romeo)
- To prevent excessive fees, anyone can pay amount C to the **feeRecipient**

  Tango pays Romeo C and becomes **feeRecipient** but Tango must set a lower fee than Romeo.

  C is roughly double amount initially paid to relay the block header

How to be a Relayer and earn ETH

- https://github.com/ConsenSys/btcrelay-fetchd
- Fetches new headers and submits to BTC Relay
- Public testnet, so you don't spend real ETH

```
python fetchd.py -s <YourUnlockedAccount> -r
<BTCRelayAddr> -n btc --rpcPort 8545 --fetch -d
--gasPrice 50000000000
```

- `--fee <weiAmount>` to set a fee
- Contract address of BTC Relay:
  http://rawgit.com/ethereum/btcrelay/master/examples/relayContractStatus.html

Extensibility Example

- Sidechain Flow

  >> lockBTC >> createToken >> destroyToken >> unlockBTC

- A contract could use BTC Relay to verify the lockBTC transaction, then parse it to create a BTC Token

- BTC Relay is modular and easy to integrate

Integrating into a DApp

https://github.com/ethereum/btcrelay/blob/master/examples/BitcoinProcessor.sol

```
contract BitcoinProcessor {
    address private _trustedBTCRelay;

    function BitcoinProcessor(address trustedBTCRelay)
        _trustedBTCRelay = trustedBTCRelay;

    function processTransaction(bytes txn, uint256 txHash) returns (int256)
        // only allow trustedBTCRelay, otherwise anyone can give fake txn
        if (msg.sender == _trustedBTCRelay) {
            // parse & do whatever with txn
            return 1;
        }
        return 0;
}
```

BTC RELAY

Conclusion

3 “secrets”:

1. Build SPV-chain
2. Verify transactions using Merkle proof
3. Incentives for headers needed for SPV-chain

Trustless • Decentralized • Autonomous • Modular

References

- btcrelay.org
- https://github.com/ethereum/btcrelay
- https://github.com/ethereum/btcrelay/wiki/References
- https://gitter.im/ethereum/btcrelay

Acknowledgments

- Martin Becze, Vitalik Buterin, Aaron Davis, Vincent Gariepy, Jesse Grushack, Connor Keenan, Martin Köppelmann, Alex Leverington, Joseph Lubin, Christian Lundkvist, Eva Shon, Vlad Todirut
- Many thanks to above supporters of BTC Relay for their priceless time, thought, and work!
- eg. insightful ideas, security gaps, logos, colors, website, and the design of these slides.