Bitcoin's Moving Parts

- Coins
- Digital signatures
- Chain of blocks
- Proof of work
- Censorship resistance
- Anatomy of a Transaction

The Problem

Alice emails Bob: "I give you 2 BTC"

The Problem

Alice emails Bob: "I give you 2 BTC"

1. Why is it worth anything?

The Problem

Alice emails Bob: "I give you 2 BTC"

1. Why is it worth anything?
2. What did Bob get that Alice gave up?

Coins

- Alice has two 1.5 BTC coins[1]
- Alice emails Bob:

  I spend my coins, and create:
  - 2 BTC coin for Bob; and
  - 1 BTC coin for Alice

[1] UTXO: Unspent Transaction (tx) Output

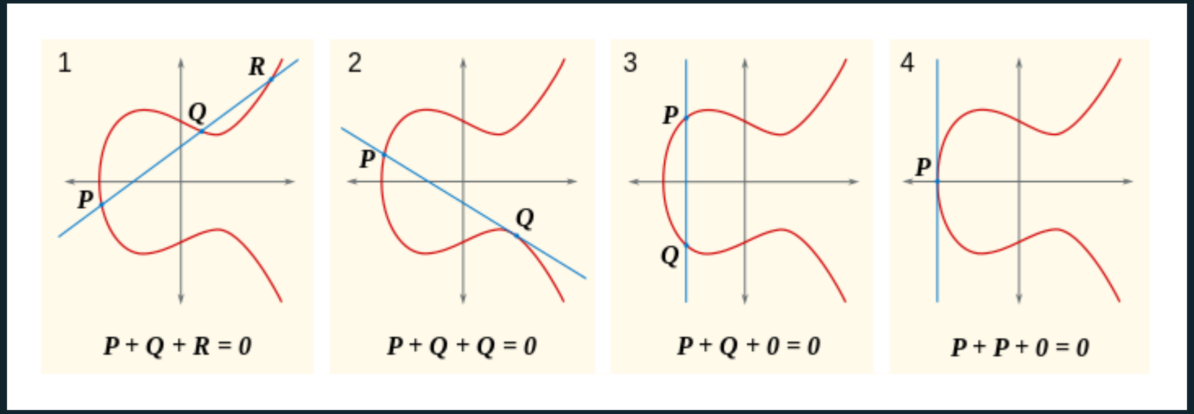
The Problem

Alice emails Bob: "I give you 2 BTC"

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?

Digital signatures

- Private key: big random number
- Public key: point on curve[2]



- private key can encrypt or sign message

[2] secp256k1

Digital signatures

- Coin is signed message: "Only Alice can spend me" [3]
- Who's Alice?
- Bitcoin address derived from public key [4]
- Alice is whoever can reveal public key and sign a message with private key

[3] The new coins say "Only Bob can spend me" and the change says "Only Alice can spend me".

[4] Q&A tip: why not just use the public key as a Bitcoin address?

The Problem

Alice emails Bob: "I give you 2 BTC"

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?
4. What if Alice sends Carol the same coins?

The Blockchain

A Blockchain is a chain of blocks

— Peter Todd

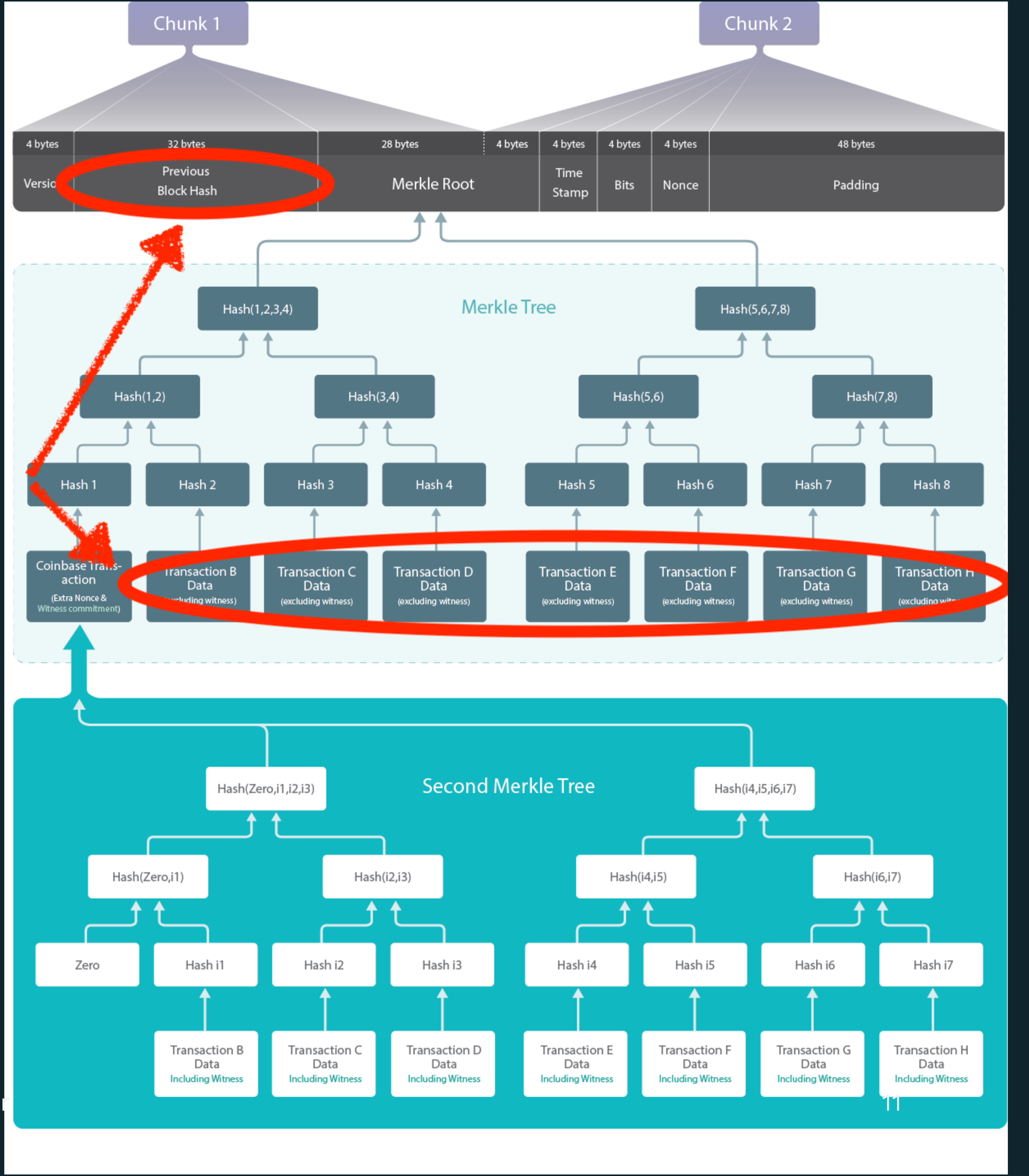
A Blockchain is a database with virtue

— Chris DeRose

The Blockchain

Things we care about to solve problem (4):

- Publish all transactions, and in which order
- Ensure everyone can see all transactions


Chunk 1
Chunk 2
4 bytes
32 bytes
28 bytes
4 bytes
4 bytes
4 bytes
4 bytes
48 bytes
Versio
Previous Block Hash
Merkle Root
Time Stamp
Bits
Nonce
Padding
Merkle Tree
Hash(1,2,3,4)
Hash(5,6,7,8)
Hash(1,2)
Hash(3,4)
Hash(5,6)
Hash(7,8)
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Hash 7
Hash 8
Coinbase Transaction (Extra Nonce & Witness commitment)
Transaction B Data
Transaction C Data (excluding witness)
Transaction D Data (excluding witness)
Transaction E Data (excluding witness)
Transaction F Data (excluding witness)
Transaction G Data (excluding witness)
Transaction H Data
Second Merkle Tree
Hash(Zero,i1,i2,i3)
Hash(i4,i5,i6,i7)
Hash(Zero,i1)
Hash(i2,i3)
Hash(i4,i5)
Hash(i6,i7)
Zero
Hash i1
Hash i2
Hash i3
Hash i4
Hash i5
Hash i6
Hash i7
Transaction B Data Including Witness
Transaction C Data Including Witness
Transaction D Data Including Witness
Transaction E Data Including Witness
Transaction F Data Including Witness
Transaction G Data Including Witness
Transaction H Data Including Witness

The Problem

Alice publishes: "I give Bob my 2 BTC"

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?
4. What if Alice sends Carol the same coins?
   — it's on the blockchain so everyone can see it
   — what if there's many different blockchains?

The Problem

Alice publishes: "I give Bob my 2 BTC"

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?
4. What if Alice sends Carol the same coins?
   - it's on the blockchain so everyone can see it
   - what if there's many different blockchains?

Solution 1: use a regulator

- declare existence of conflicting blockchains fraud
- give regulator(s) access to your database
- add additional crypto magic
- brand it "blockchain inspired technology"

Solution 2: Proof-of-Work

Convince someone to:

— do useless work
— which uniquely commits to transaction data
— in exchange for coins.

Throw dice on a piece of paper with the transaction list.

Extreme Luck?

Proof-of-Work

- `sha("000001 | Alice sends Bob 2 BTC, etc") = 0fed9a90`
- `sha("000002 | Alice sends Bob 2 BTC, etc") = e7c54529`
- `sha("000003 | Alice sends Bob 2 BTC, etc") = 6c48ab21`
- `sha("855453 | Alice sends Bob 2 BTC, etc") = 000005e6`
- N leading zeros -> X kWh * £0.10 -> £... per block [5]
- Miner creates coin out of thin air which Alice & Bob consider valid

[5] Any scarce resource will do, but the simplest known combination with the right properties is electricity + specialized chips + hashing.

The Problem

Alice publishes: "I give Bob my 2 BTC", miner burns electricity to attest this.

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?
4. What if Alice sends Carol the same coins?
   - it's on the blockchain so everyone can see it
   - what if there's many different blockchains?

The Problem

Alice publishes: "I give Bob my 2 BTC", miner burns electricity to attest this.

1. Why is it worth anything?
2. What did Bob get that Alice gave up?
3. How does Alice prove she owned the coin?
4. What if Alice sends Carol the same coins?
5. What if someone doesn't like Bob?[6]

[6] i.e. wants to stop the transaction

Censorship resistance

- Miners compete, fees offer extra incentive
- P2P: transactions and blocks route around censorship
- Fungability: all transacions (should) look the same
- Lot's of problems left to solve

Anatomy of a Bitcoin Transaction

Alice publishes: "I give Bob my 2 BTC"

- Bob is whoever can reveal public key corresponding to Bob's address and sign a message with private key:

Alice publishes:
OP_DUP OP_HASH160 <Bob's address> OP_EQUALVERIFY OP_CHECKSIG

Bob spends:
<Bob's signature><Bob's pubkey>

Anatomy of a Bitcoin Transaction

Alice publishes: "I give Bob my 2 BTC"

— Bob is whoever can reveal public key corresponding to Bob's address and sign a message with private key:

Alice publishes:
OP_DUP OP_HASH160 <Bob's address> OP_EQUALVERIFY
OP_CHECKSIG

Bob spends:
<Bob's signature><Bob's pubkey>

Script stack

- <Bob's signature>
- <Bob's pubkey>
- OP_DUP
- OP_HASH160
- <Bob's address>
- OP_EQUALVERIFY
- OP_CHECKSIG

Script stack

- <Bob's signature>
- <Bob's pubkey>
- <Bob's pubkey>
- OP_HASH160
- <Bob's address>
- OP_EQUALVERIFY
- OP_CHECKSIG

Script stack

- <Bob's signature>
- <Bob's pubkey>
- <Bob's address>
- <Bob's address>
- OP_EQUALVERIFY
- OP_CHECKSIG

Script stack

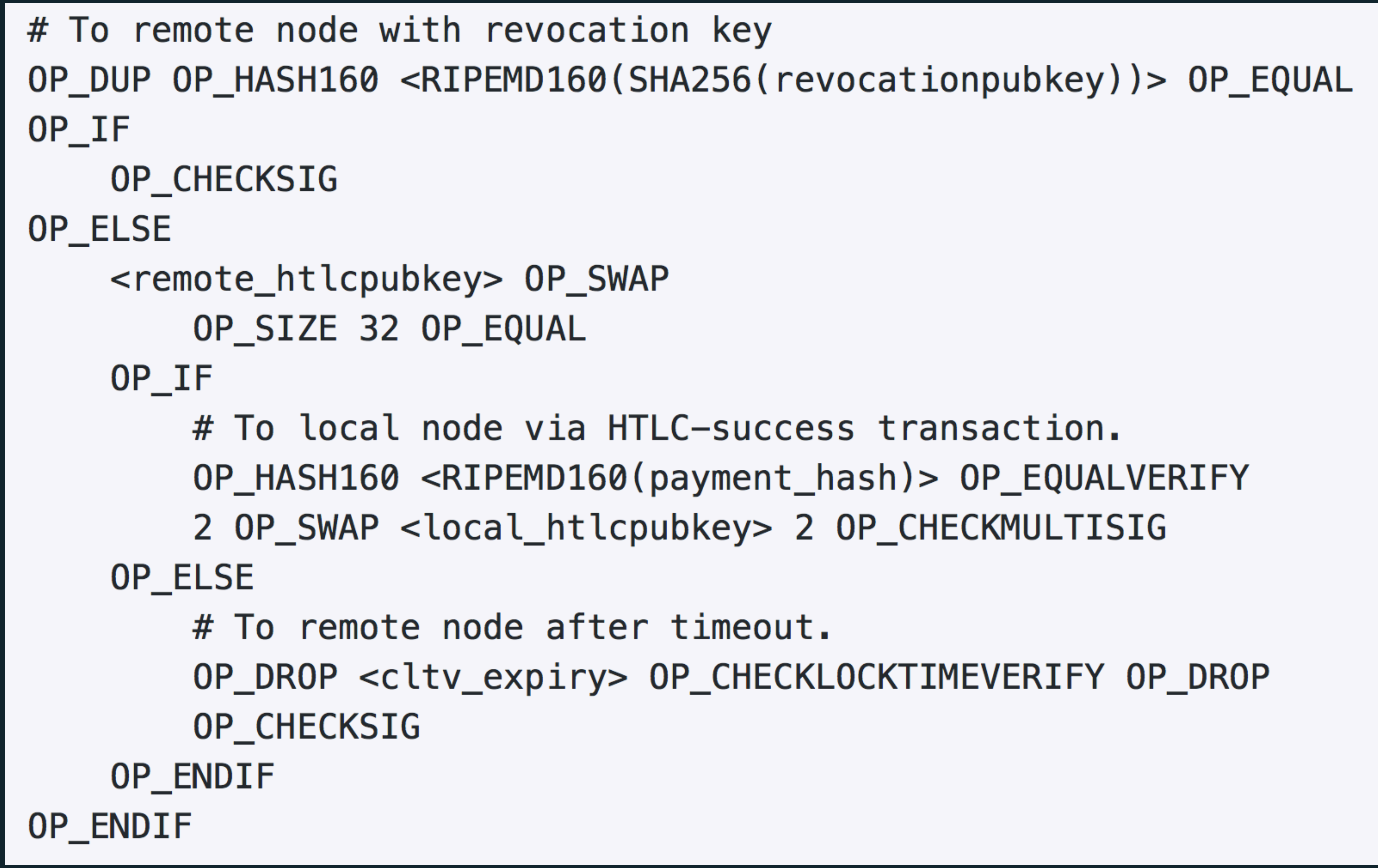
- <Bob's signature>
- <Bob's pubkey>
- OP_CHECKSIG

Script stack

— `true`

Other opcodes

- OP_CHECKMULTISIG : N of M sigs
- OP_CHECKLOCKTIMEVERIFY: HODL
- OP_IF / OP_ELSE
- OP_RETURN: 80 bytes spam [8]
- OP_NOP: does nothing (yet!)



```
# To remote node with revocation key
OP_DUP OP_HASH160 <RIPEMD160(SHA256(revocationpubkey))> OP_EQUAL
OP_IF
    OP_CHECKSIG
OP_ELSE
    <remote_htlcpubkey> OP_SWAP
        OP_SIZE 32 OP_EQUAL
    OP_IF
        # To local node via HTLC-success transaction.
        OP_HASH160 <RIPEMD160(payment_hash)> OP_EQUALVERIFY
        2 OP_SWAP <local_htlcpubkey> 2 OP_CHECKMULTISIG
    OP_ELSE
        # To remote node after timeout.
        OP_DROP <cltv_expiry> OP_CHECKLOCKTIMEVERIFY OP_DROP
        OP_CHECKSIG
    OP_ENDIF
OP_ENDIF
```

[8] e.g. Rare Pepe trades: CryptoKitties, but with frogs

Thanks

Slides: slideshare.net/provoost

Blog: medium.com/provoost-on-crypto

PGP:

```
ED9B DF7A D6A5 5E23 2E84  5242 57FF 9BDB CC30 1009
```