Blockchains & Cryptocurrencies

Anonymity in Cryptocurrencies III / Scaling I

Instructors: Matthew Green
Johns Hopkins University - Fall 2020

Housekeeping

- Project updates today
  - Gijs is sending a scheduling note for presentations
  - Please respond! We need this to schedule final classes
- Assignment 3 will be out tomorrow (11/12/2000)

News?

Vitalik Buterin at Ethereal Summit (Ali Powell/CoinDesk)

After going live last week, the deposit contract for Ethereum's 2.0 upgrade now holds over 50,000 ETH – 10% of the threshold needed to activate the watershed update.

This deposit contract is the cornerstone of the Ethereum 2.0 update and serves as a bridge for the migration Ethereum network away from proof-of-work (PoW) to a new technical infrastructure that supports proof-of-stake (PoS).

In order to become a transaction validator on the new network (those individuals who process transactions like miners in PoW), an Ethereum user must stake at least 32 ETH. There are currently 52,801 ETH locked up in the deposit contract worth $23.8 million, and it will need at least 524,288 ETH split between 16,384 stakers to trigger Eth 2.0's "genesis event" and activate the upgrade.

Subscribe to Blockchain Bites, our daily update with the latest stories.

News?

Review stuff (from last time)

Confidential Transactions

- Pedersen commitments are <u>additively homomorphic:</u>

  - Commit to "m1": $C_1 = g^{m_1} h^{r_1}$
    Commit to "m2": $C_2 = g^{m_2} h^{r_2}$

  - Now multiply the two commitments together:

$$\begin{aligned} C_3 &= C_1 \cdot C_2 \\ &= g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2} \\ &= g^{m_1+m_2} h^{r_1+r_2} \end{aligned}$$

Notice that C3 is a commitment to the <u>sum</u> m1+m2 (under randomness r1+r2)

Confidential Transactions

- Introduced by Maxwell
  - Does not provide privacy for the identity of the input transactions, can be combined with CoinJoin or ringsides
  - Does allow you to hide the value of input transactions
- Basic idea: use a Pedersen commitment to each transaction value, rather than revealing this in cleartext $C = g^v h^r$
- Do a CoinJoin, and use additive property of Pedersen commitments to sum the values and then subtract each output commitment (board)

MimbleWimble (Grin)

- Combines CoinJoin with Confidential Transactions, provides both services in a single network

Scaling

The problem

- Bitcoin transaction rate: 5-7 tx/sec
  - Bounded by block size (Segwit helps), TX size
  - All transactions must be globally verified, stored
- Ethereum: 15 transactions per second if they're small
- Visa: 24,000/sec peak (150M/day globally)
- WeChat 256,000/sec peak

Faster computers?

- Why not just build faster computers?

Faster computers?

- Why not just build faster computers?
  - Loss of decentralization
  - Eventually we saturate links, due to broadcast network
  - Replicated global state falls apart
  - Scaling is possible (see Visa, WePay etc.) but it requires dedicated, centralized servers

Can we do better?

Can we do better?

- Current ideas:
  - "Off chain"
  - "Sharding"
- New consensus algorithms

Off-chain transactions (channels)

- In current Bitcoin-style networks, every transaction appears on the blockchain
  - This allows the whole network to verify financial integrity
  - I.e., we can't go off an do transactions elsewhere, accidentally/deliberately inflate the money supply
- But why does the network need to see <u>every</u> transaction?

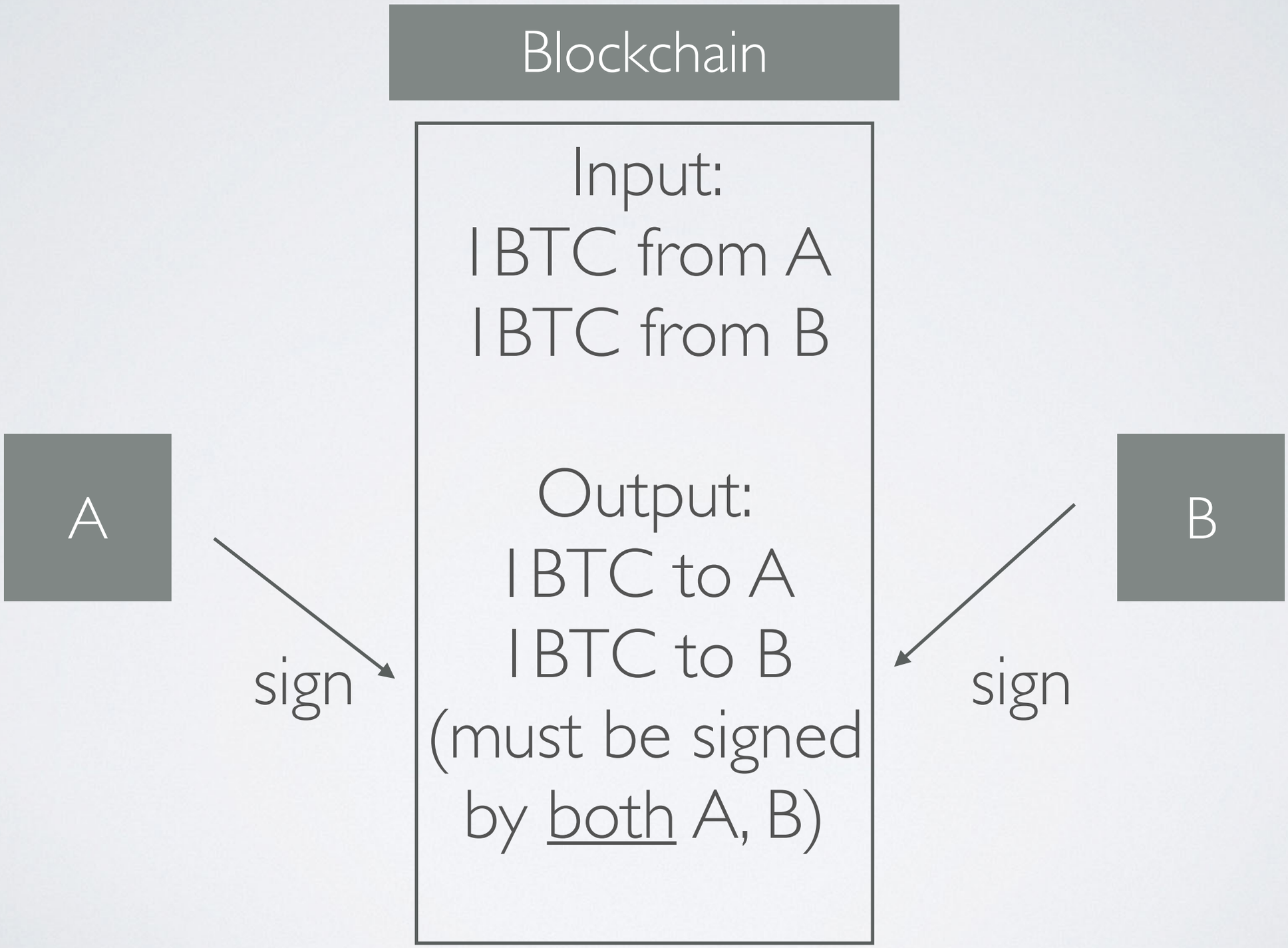
Off-chain transactions (channels)

- Overarching idea:
  - If a transaction doesn't affect anyone else (except for the parties willing to risk money), chain doesn't need to see it
  - Simplest example (but centralized):
    - Multiple parties deposit money into an exchange
    - Exchange is just a centralized bank, so everyone can quickly transmit money by adjusting balances
    - Only withdrawals need on-chain transactions

Off-chain transactions (channels)

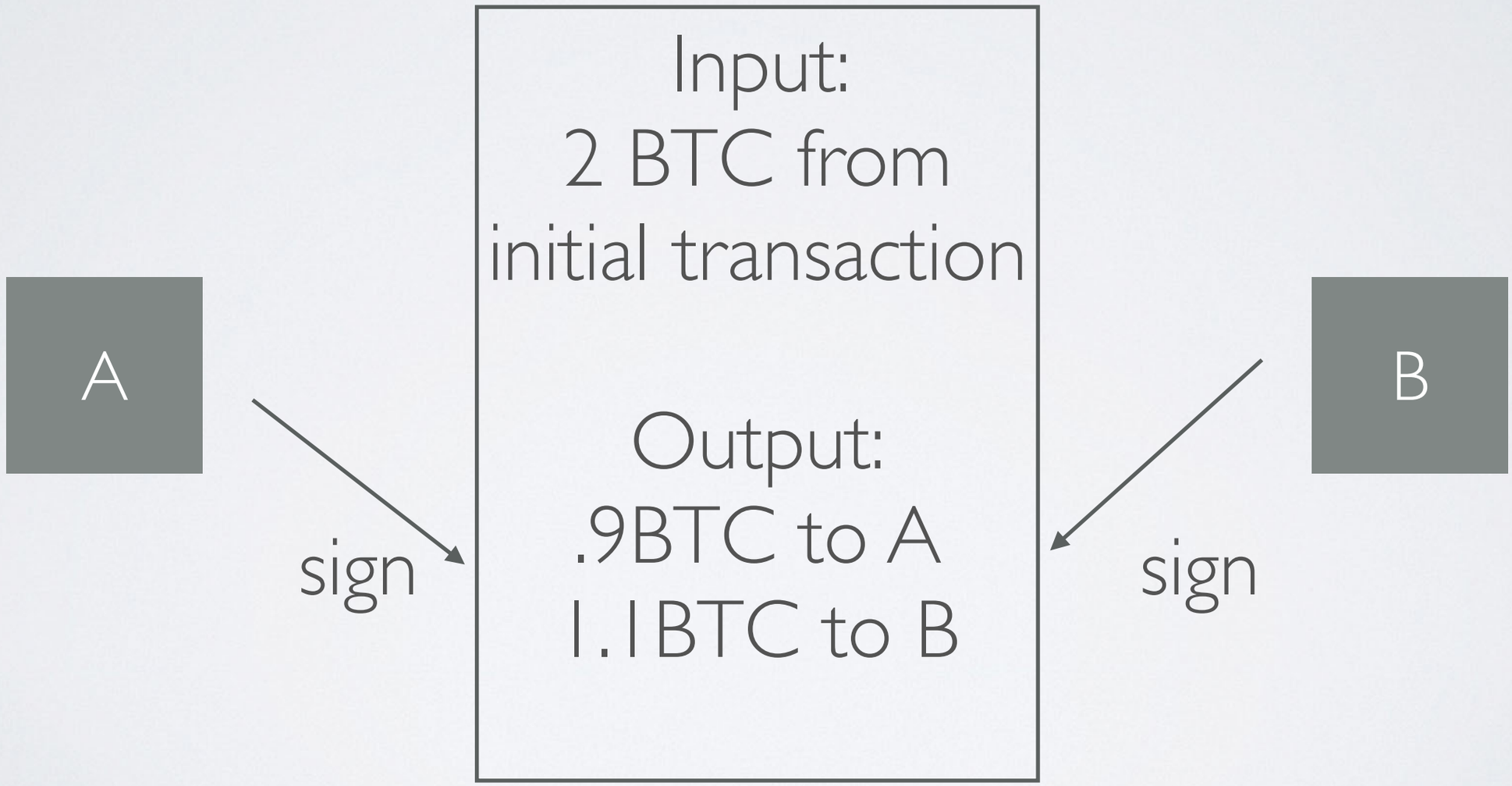
- Off-chain exchange example still risks loss of funds
  - If the exchange disappears, your money goes with it
  - See e.g., QuadrigaCX
- The only benefit here is that the rest of the network can't lose money, e.g., due to inflation

Channels: Step 1



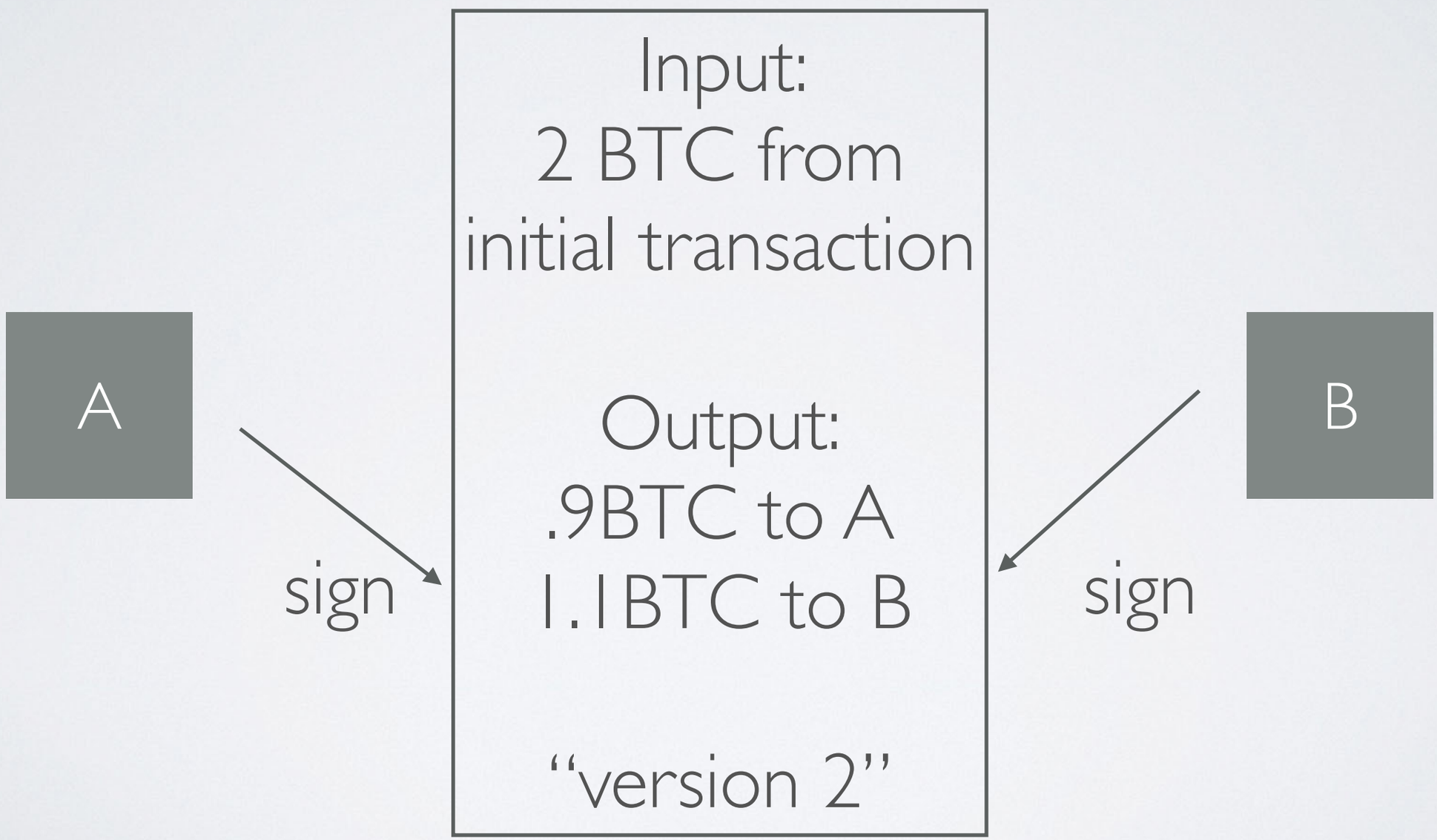
Channels: Step 2

* A pays B 0.2BTC



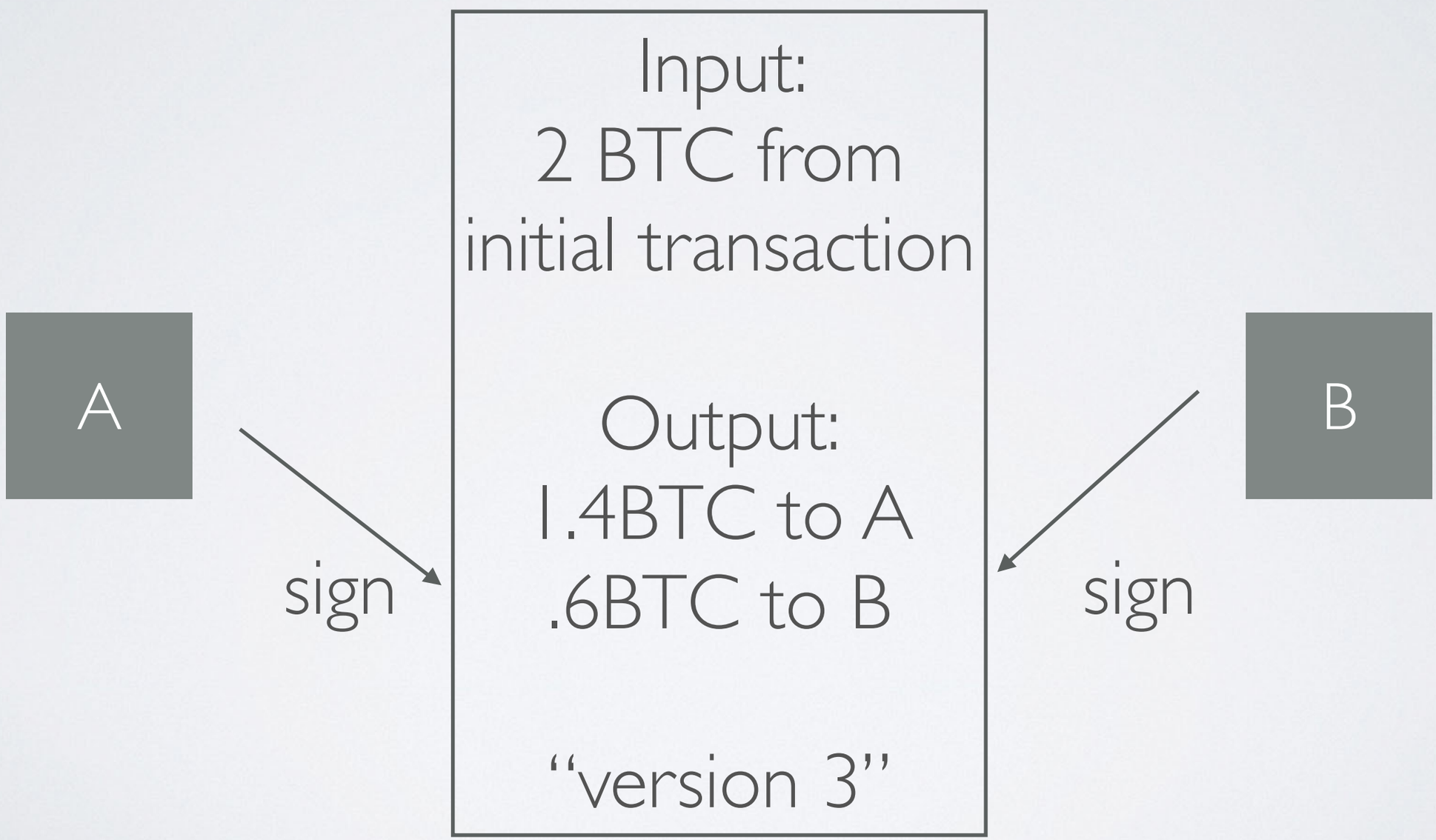
Channels: Step 2

* A pays B 0.2BTC



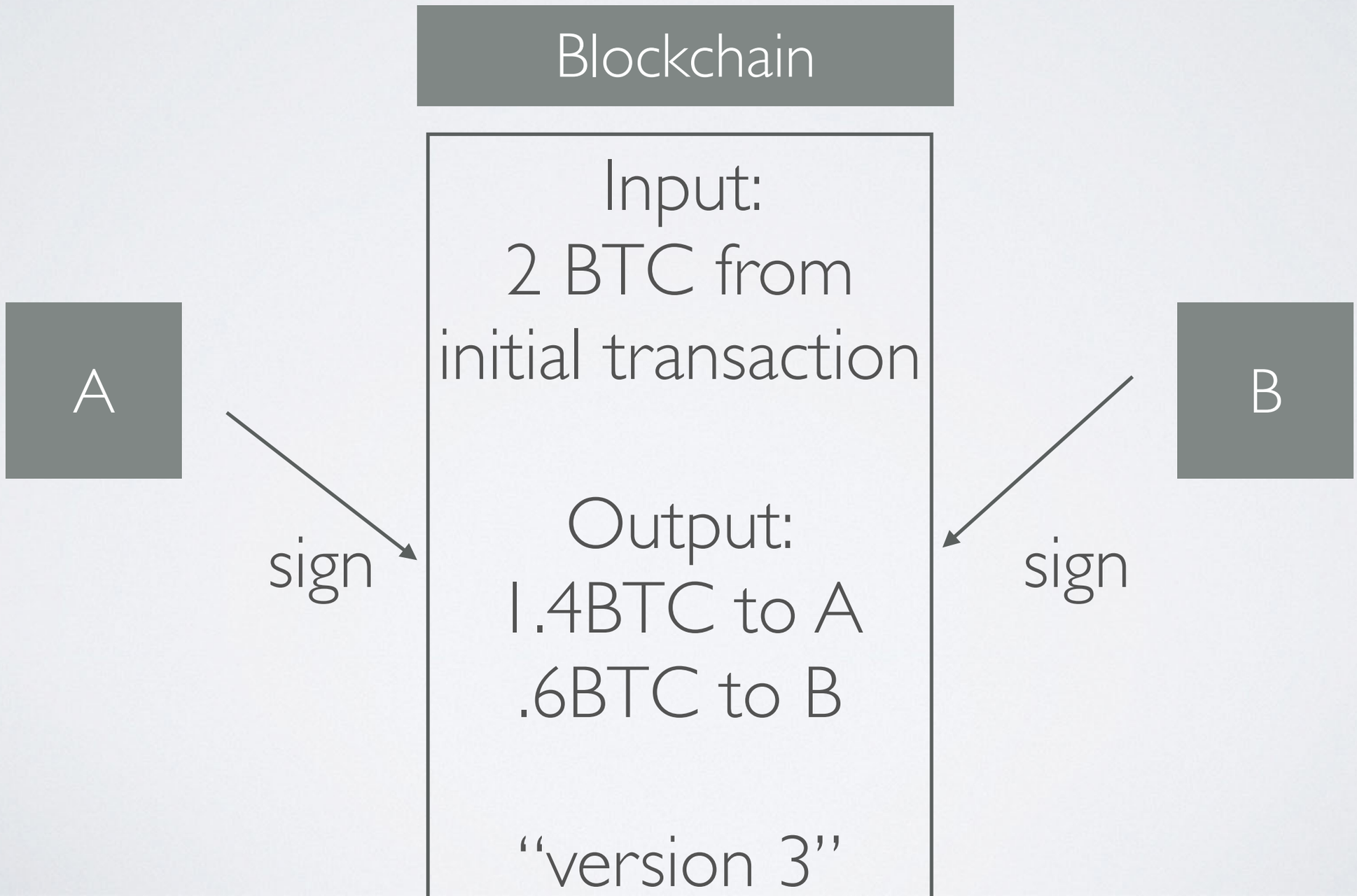
Channels: Step 3……

* B pays A .5 BTC



Channels: Closure

* Either party posts the most recent version of the transaction to the blockchain (all older versions get ignored)



Dispute resolution

* What if someone posts an older, out-of-date version?
* What if nobody signs the first "closure" transaction? How do you escape a payment channel in the worst case?