

18734: Foundations of Privacy

Bitcoin

Kyle Soska
Anupam Datta
CMU

Fall 2016

Part I: Reconstructing Bitcoin

Goal

- Cash has some nice properties
 - Easy
 - Anonymous
 - Permanent / Non-reversible
- E-commerce has some nice properties
 - Convenient, instant, etc
- Question: Can we support things like E-commerce using something like cash?

Goal

- Want to create the digital equivalent of cash
- Problem: The environment is much different, copy and paste (counterfeiting) is easy on a computer
- Solution: We need to be clever about how digital cash is constructed and used

Goal

- A rational reconstruction of Bitcoin



1. Start with straw man design
2. Identify weaknesses
3. Augment design and iterate

Step 1: A signed letter of intent

- Alice: “I, Alice, am giving Bob one coin”
- Alice digitally signs message and announces bits to everyone.
- Properties
 - Establishment of Alice’s intent
 - Limited protection from forgery
- Weakness
 - Coins are not unique; can be duplicated

Step 2: Unique serial nos. on coins

- Alice: "I, Alice, am giving Bob one coin, with serial number 8740348"
- Alice: "I, Alice, am giving Bob one coin, with serial number 8770431"
- Bank issues coins with unique serial numbers, keeps track of who owns coins, verifies transactions
- Properties
 - Establishment of Alice's intent
 - Better protection from forgery
- Weaknesses
 - Need trusted bank to issue coins, keep track of who owns coins, verify transactions
 - Bank can link transactions to identity

Possible design

- First generation crypto-currencies [Chaum et al.]
 - Retain bank
 - Ensure that bank cannot link transactions to identity
 - Agents cannot double spend their electronic coins
- Key novelty in Bitcoin design
 - No centralized bank

Step 3: Making *everyone* the bank

- Everyone maintains a copy of the public ledger (block chain) of transactions (keeps track of who owns coins)
- Alice: "I, Alice, am giving Bob one coin, with serial number 8740348"
- Bob uses his copy of the block chain to check that the coin is Alice's; he broadcasts both Alice's message and his acceptance of the transaction to the entire network, and everyone updates their copy of the block chain.
- Weaknesses
 - How to get serial numbers?
 - Double-spending: What if Alice gives the same coin to Bob and Charlie at the same time?

A Network Verification Design

- Bob does not verify Alice's coin by himself.
- Asks everyone on the network to verify
- When "enough" people confirm that the coin is indeed Alice's, Bob accepts and everyone updates their block chain
- Weakness:
 - Sybil attack: Alice creates many fake agents who lie for her; Alice spends the same coin many times

Step 4: Proof-of-work

- Computationally costly for network users to validate transactions
- Reward network users for validating transactions
- Properties
 - Sybil attack won't work unless dishonest agents put in significant computational resources
 - Verifiers rewarded with fixed number of bitcoins for a batch of transactions (details soon)
 - Additional ideas to ensure that ledger succinctly maintains history of all transactions (details soon)

Rigorous Analysis of Bitcoin

- Topic of ongoing research

Part II: Technical Overview

Bitcoin primer (1/2)

- A peer-to-peer digital payment system
- Completely decentralized digital currency
 - **No central mint** to produce currency
 - **No central bank** to verify transactions
 - Once confirmed, transactions are **irreversible**
 - Predictable, capped, currency supply
- Key innovation in Bitcoin: coin production and verification is done by **network consensus**



Bitcoin primer (2/2)

- There is actually no notion of a “coin”



- Bitcoins are exchanged from “wallet” to “wallet”
- **Transactions** are at the heart of the protocol
- Wallets are represented by **addresses** (e.g., `1VayNert...`)
 - (An address is the public key of the wallet)

Bitcoin transactions

- Alice wants to send 1 BTC to Bob
 - She picks a transaction (or a group of transactions) that she has previously been the recipient of and that cumulatively contain at least 1 BTC
 - She then appends Bob's wallet address to the transaction and digitally signs it
- When Bob subsequently wants to spend the 1 BTC, all he has to do is to repeat the operation

Preventing double-spending

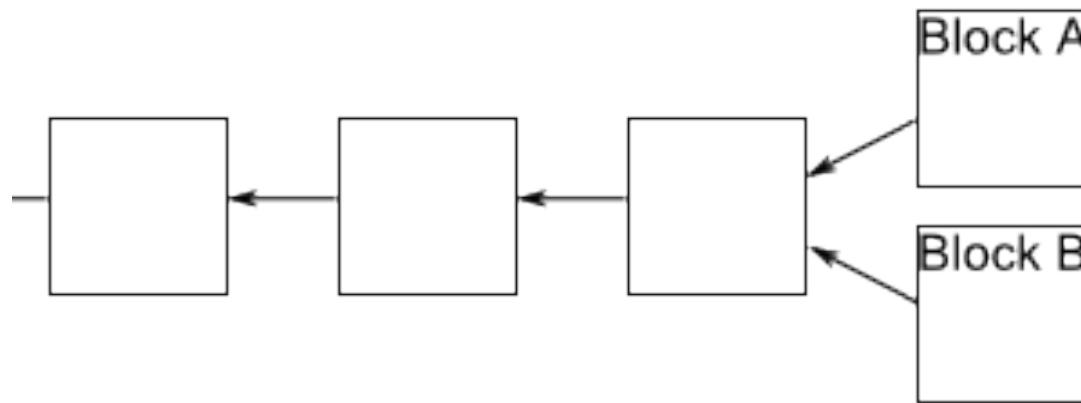
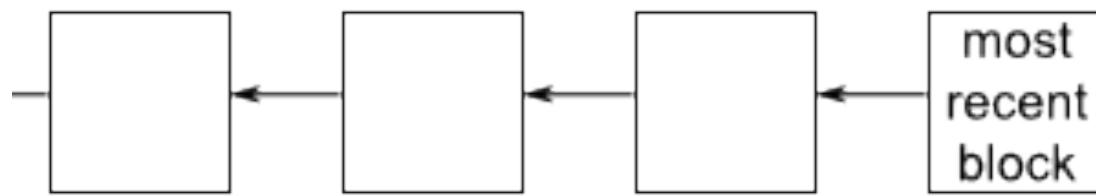
- Bob now has 1 BTC
 - He wants to send it to Charlie...
 - ... while keeping it for himself at the same time
- To prevent this Bob (and Alice before him) **has to broadcast** the transaction to everybody in the Bitcoin network
- Then other peers can verify that the transaction is not a double-spend
- Once this is done, the transaction is **embedded forever in a public ledger**

Key technical components

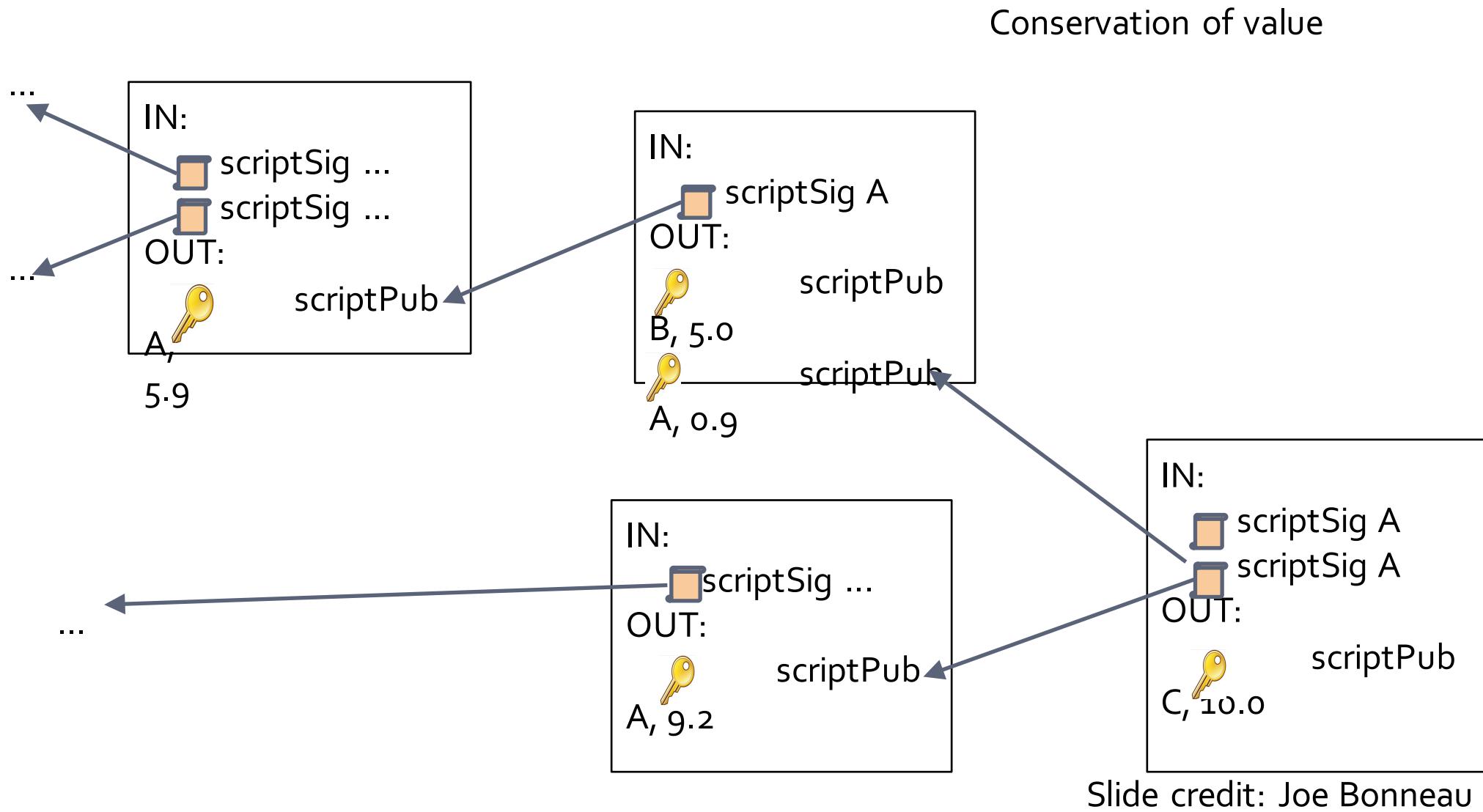
1. Transactions and Scripts
2. Consensus and Mining
3. Communication Network

Transactions and Scripts

The Block Chain of Transactions



Bitcoin is *transaction-based*

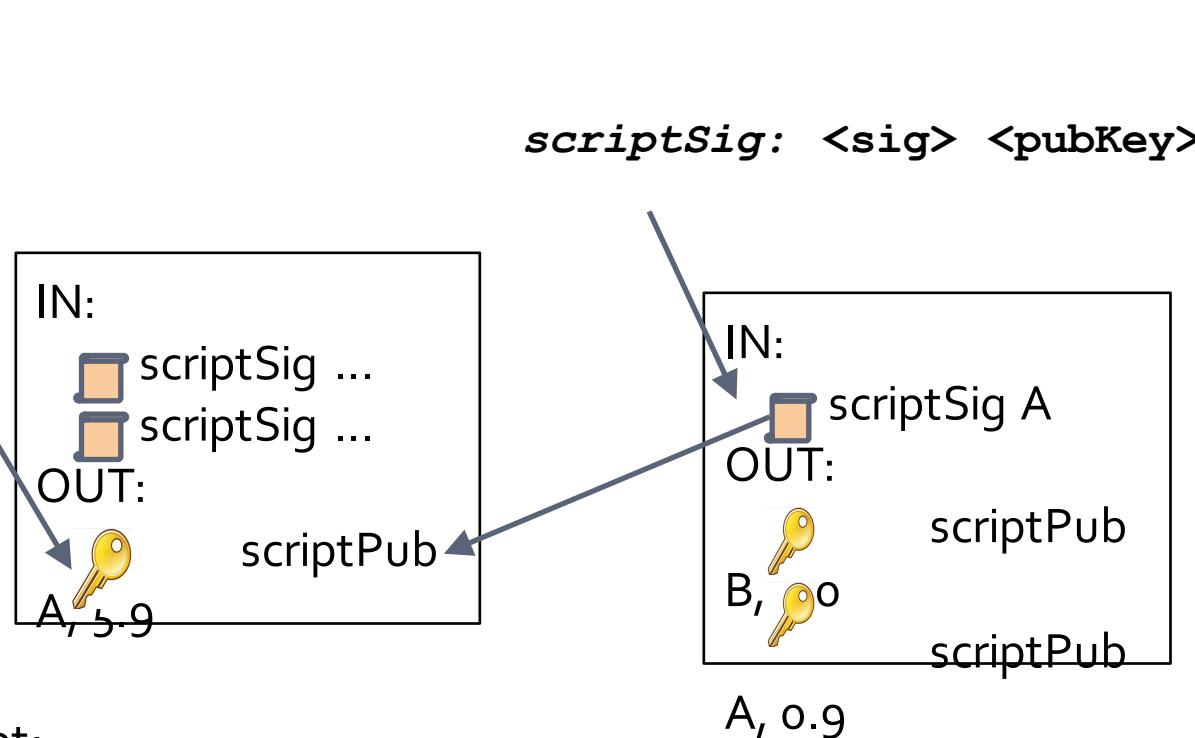


A Bitcoin Transaction

```
1. {"hash":"7c4025...",           //serial number: hash of transaction
2. "ver":1,                      // protocol version
3. "vin_sz":1,                   // no.of inputs
4. "vout_sz":1,                  // no.of outputs
5. "lock_time":0,                // transaction finalized after time
6. "size":224,                   // no. of bytes in transaction
7. "in": [                        // input of transaction 7-11
8.   {"prev_out":               // input is an output of a previous transact.
9.     {"hash":"2007ae...",      // serial number of previous transact.
10.    "n":0},                  // output number of previous transact.
11.    "scriptSig":"304502... 042b2d..."}], // signature and pub key of sender
12.   "out": [                     // output of transaction 12-14
13.     {"value":"0.31900000",    // outputs 0.319 BTC
14.      "scriptPubKey": "OP_DUP OP_HASH160 a7db6f OP_EQUALVERIFY
OP_CHECKSIG"}]}                 // script for verifying transaction
```

Bitcoin transactions specify scripts

scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG



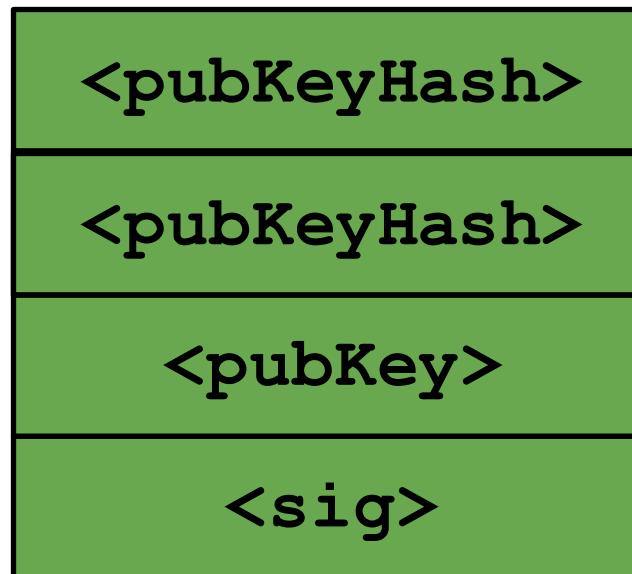
Redemption script:

<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

ScriptPubKey

- Conditions under which the transaction output can be redeemed, i.e. included as an input in another transaction

Bitcoin transactions specify *scripts* in a stack language



Slide credit: Joe Bonneau

Bitcoin script features

<https://en.bitcoin.it/wiki/Script>

Key technical components

1. Transactions and Scripts
2. Consensus and Mining
3. Communication Network

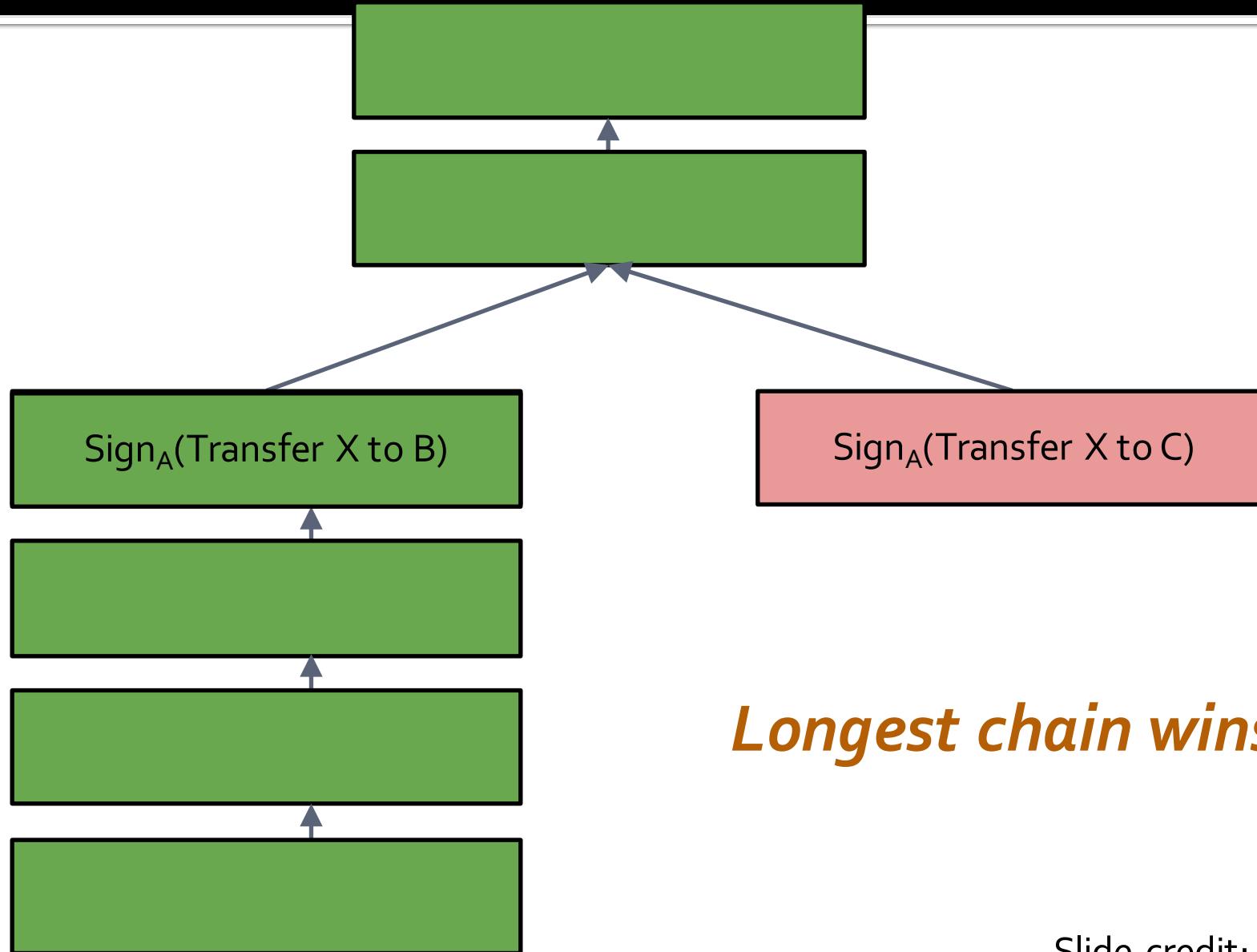
Consensus and Mining

Nakamoto consensus

- Any party can attempt to add a block to the block chain of transactions
 - 1. Collect set of valid pending transactions (i.e., with verified transaction scripts) into new block
 - 2. Solve associated cryptographic puzzle (more details soon)
 - 3. Append to longest block chain

First party to complete wins

Prevents double spending



Slide credit: Joe Bonneau

Coin production

- Coin production is embedded in the verification process
- Verifiers (“miners”) verify batches of transactions at once
 - In exchange for which they are allowed to add a “creation” transaction to the batch and give themselves a fixed amount of money
 - 50 BTC originally, 25 BTC now, divided by two every so often
 - Verification is combined with a “proof-of-work” scheme to ensure
 - That transactions have proper timestamping
 - That currency production is rate-limited



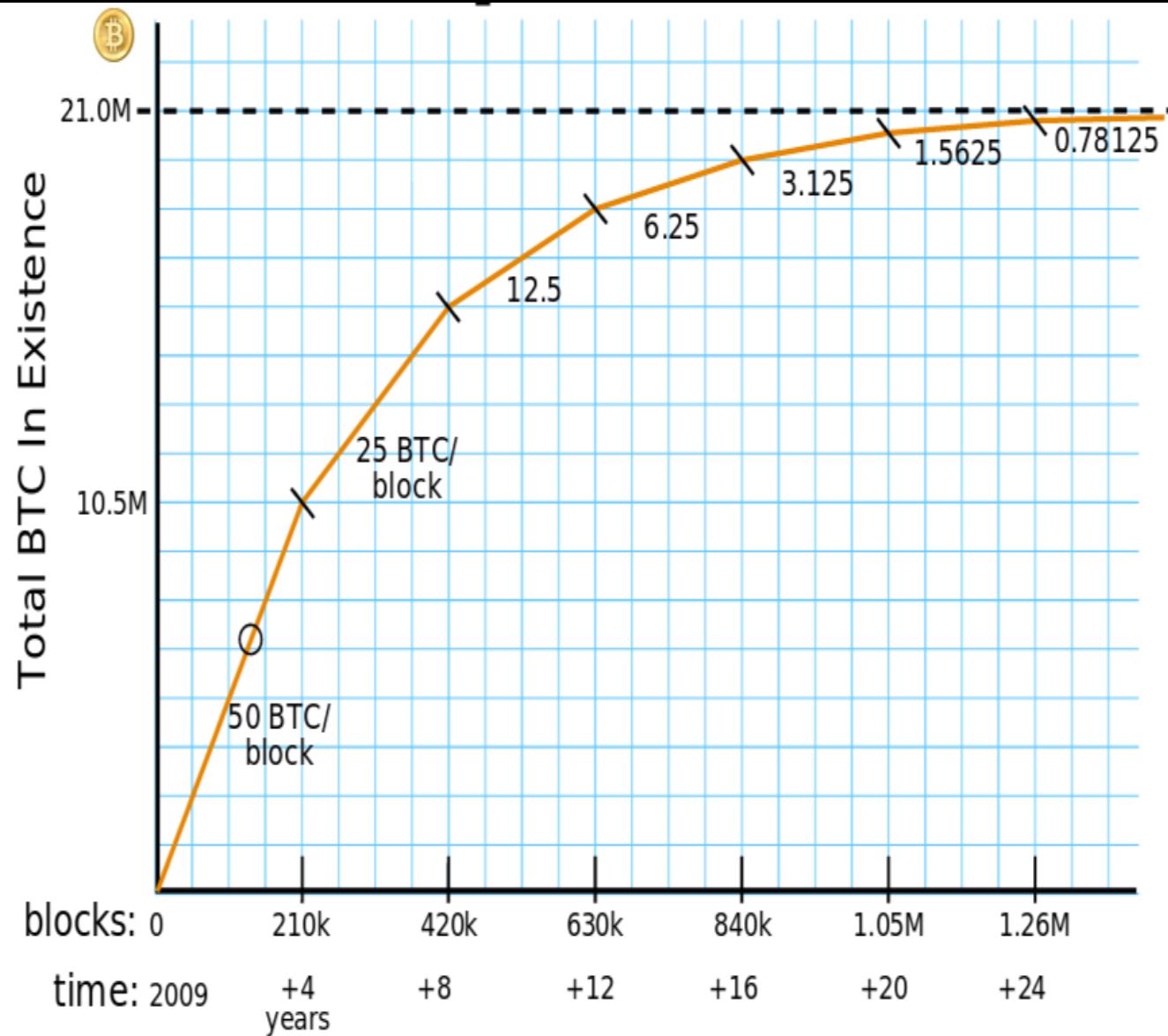
More on mining incentives

- Miners solve a cryptographic puzzle:
Find x s.t. $H(x||I) < y$ where I is the batch of transactions.
- There is no good algorithm to solve this (H is a cryptographically secure hash function)
 - **Brute-force:** try $x=0, x=1, x=2, x=\dots$
 - The lower y , the harder the puzzle
- Difficulty is tunable and is (by edict) designed to be inversely proportional to the total computational power of the network
- The goal is to have one block every ten minutes
 - Predictable supply of currency (independent of the difficulty)
 - **But this limits how quickly transactions can be verified**
 - At least 10 minutes, usually 60 minutes is recommended

Transaction fees

- In addition to the bonus they get for mining, miners get “transaction fees”
 - Leftover “change” voluntarily left in transactions
- Because the bonus is decreasing over time, the expectation is that transaction fees will increase over time to make up for lost mining revenue

Mining rewards



Courtesy:
Brian Warner

Stability Questions

- Will network reach consensus on which chain is the longest?
 - Yes, roughly if miners controlling majority of computational power follow the protocol faithfully + timely broadcast channel
- Can we be sure that a block that is on the current longest chain will end up in the eventual longest chain?
 - Yes, roughly with exponentially increasing probability in the depth of the fork under same conditions as above

Key technical components

1. Transactions and Scripts
2. Consensus and Mining
3. Communication Network

Communication Network

Overview

- Decentralized, ad hoc peer-to-peer broadcast network used to announce new transactions and proposed blocks
- Leverages prior work on this topic

Impact on consensus

- Latency between discovery of a block and its receipt by all other nodes implies higher chance of temporary fork
 - Bitcoin design choice: 10 min as block creation time
- Malicious miner who controls a large portion of the network may favor broadcasting their blocks implies their blocks more likely to be on winning fork
 - Bitcoin design choice: decentralized network, low latency, difficult to censor or delay messages

Network topology and discovery

- Any node can join network by connecting to random sample of existing nodes
 - Default: 8 outgoing connections, 125 incoming
 - Well-connected random network, with low degree yet low diameter
- Suitable for rapid broadcast of information through diffusion

Connection initiation

- Upon receiving an incoming connection, a node asks its peer for a sample from its list of known addresses

Communication protocol

- New blocks and pending transactions are broadcast to the entire network by flooding
- Performance optimizations
 - Forward new data once only
 - Only relay transactions and blocks that are valid
 - Only relay the first block they hear of when two blocks are found in a temporary fork
 - Don't broadcast pending transactions that conflict with (double spend) pending transactions they have sent

Relay policy

- Strict policies to defend against denial of service attacks
- Example:
 - Default nodes refuse to relay more than a few thousand transactions below BTC 0.001 per minute as a penny-flooding defense

Part IV: Using Bitcoin

Getting Bitcoin

- Become a miner
 - Nowadays only profitable if dedicated (ASIC) hardware
- Buy at an exchange
 - CampBX, Bitstamp, BTC-e, Coinbase...
 - (Mt.Gox before they went bankrupt)
 - **Very high concentration** on exchanges through which money is exchanged
 - Exchanges fail pretty often...
 - Increasingly scrutinized by regulators
- Buy from individuals
 - Satoshi Square in NYC



Main Bitcoin uses

- As a speculative instrument
 - People invest in BTC, betting on its rising value
 - Dominant use thus far



Main Bitcoin uses

- As a currency
 - Only currency accepted on underground marketplaces (Silk Road, Evolution,...)
 - (Except for LiteCoin, which is a clone of Bitcoin)
 - Because of its “anonymity properties”
 - Still relatively modest
 - Entire Silk Road revenue represented in 1st half of 2012 about \$15M/annum
 - Gambling, poker sites
 - Large number of transactions, volume not very high
 - Other uses still in their infancy
 - Campaign contributions, online stores (e.g., Overstock), etc



Part V: Anonymity?

Pseudonymity vs anonymity

- Wallets are public/private key pairs
 - Can create as many as you want
 - Think of them as zero-cost pseudonyms
- There is no central authority issuing Bitcoins or vetting transactions
- This means Bitcoin is anonymous, right?
NO!

Bitcoin tracing

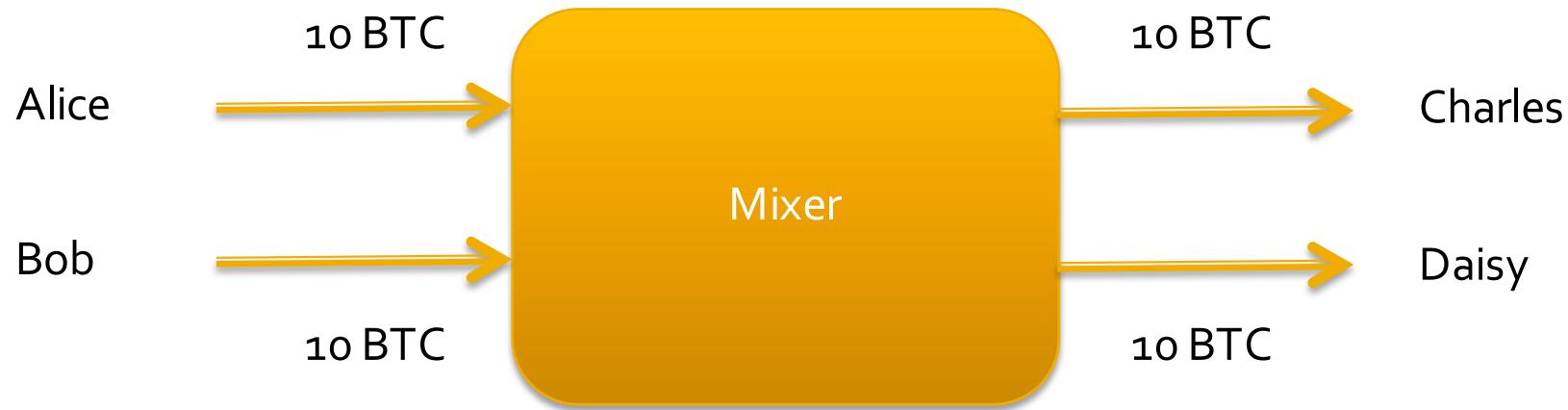
- Anonymity here implies **unlinkability** of transactions
- The **entire ledger of all transactions** is available, **forever**
 - Technically in a compressed form, but transaction chains can all be reconstructed
- Even if you add intermediary dummy steps wallets, linking the source and the destination of a transaction may be done by graph analysis...
 - Something that computer scientists know how to do!
 - Reid & Harrigan, 2011
 - Shamir & Ron, 2012
 - Meiklejohn et al., 2013

Bitcoin tracing

- Families of wallets can be pooled together as belonging to the same actual user
 - Example attack: Customer may need to assemble payment from multiple addresses she owns, thus linking her accounts
- Link address clusters to real-world identities
 - Example attack: Learn merchant identity by direct interaction with one of her addresses
- Network-level de-anonymization
 - Example attack: IP addresses leaked during broadcast

Anonymizing Bitcoin

- Mixers



- Did Alice give 10 BTC to Charles or Daisy?

Anonymizing Bitcoin

- Mixers in practice



- Need to also introduce arbitrary delays
- Introduction of change addresses, etc
- Mixer can be dishonest!

Anonymizing Bitcoin

- It's unclear how good existing Bitcoin mixers are
 - Key difference with message mixing (Tor, mixnets)
 - You can't implement arbitrary "padding"—money has to go somewhere eventually
 - Possible measure: taint
 - Amount of money that can be traced back to a given source

Anonymizing Bitcoin

- Zerocash uses zero-knowledge proofs to verify that transactions are well-formed



- These proofs are in general, very tedious to create, and so it leverages a general approach known as ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge)

Adding New Capabilities

- Ethereum: What if the scripts of Bitcoin were allowed to be Turing-complete?
- What if you could write programs that miners would execute based on some state of the network?
- Result: Smart Contracts



Acknowledgment

- Many slides from Nicolas Christin