NEXT LEVEL LAWYERING

FDCC WINTER MEETING 2022 - PALM DESERT, CALIFORNIA

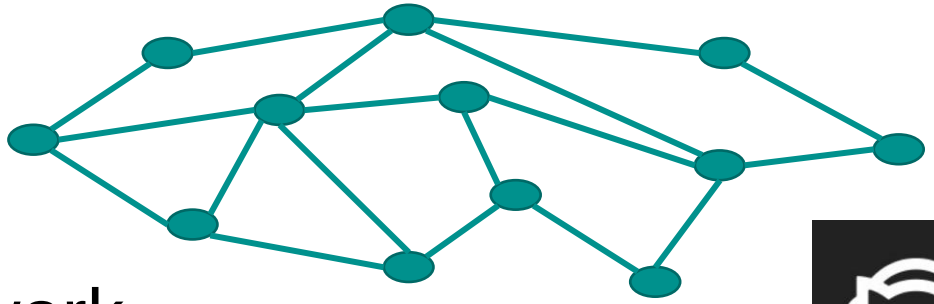
Cryptocurrency: What Is It and Why We Need to Know About It

Adam Sorini, Ph.D. (Exponent)
Justin S. Wales (Crypto.com)
Surya Sharma, Ph.D. (Exponent)

Tuesday, March 8, 2022

FEDERATION OF DEFENSE & CORPORATE COUNSEL
FDCC
DEFENSE LAWYERS. DEFENSE LEADERS.

Bitcoin



- What is it?
- Peer-to-peer computer network
- Double-entry transaction ledger

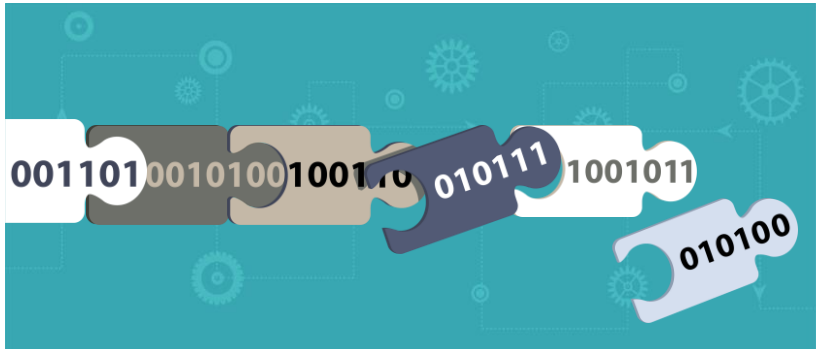
napster

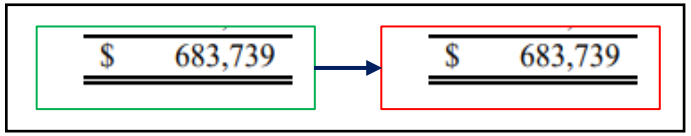
Assets	
Current assets:	
Cash and cash equivalents	$ 297,687
Short-term investments	-
Accounts receivable, net	139,861
Prepaid expenses and other assets	15,214
Total current assets	452,762
Property, equipment and leasehold improvements, net	59,971
Operating lease right-of-use asset	14,370
Goodwill	8,607
Other assets	148,029
Total Assets	$ 683,739

Liabilities and Stockholders' Equity	
Current liabilities:	
Accounts payable and accrued liabilities	$ 24,504
Accrued payroll and employee benefits	103,552
Deferred revenues	19,762
Operating lease liability	5,164
Total current liabilities	152,982
Other liabilities	103,885
Operating lease liability	9,807
Total liabilities	266,674
Total stockholders' equity	417,065
	$ 683,739

Important Bitcoin Data Structures

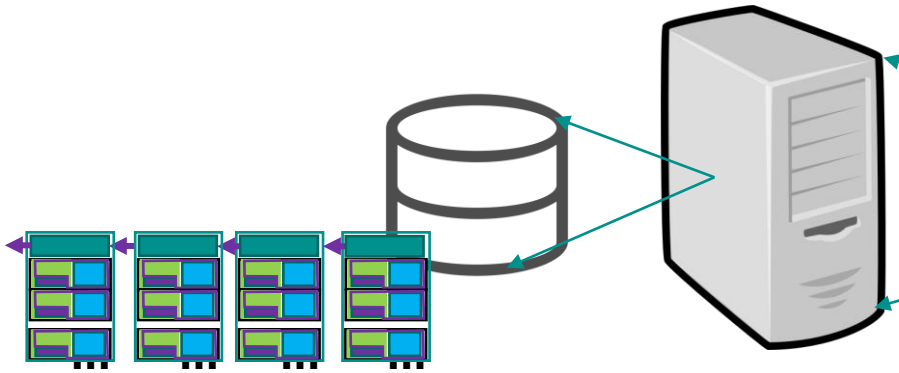
- Blockchain
  - A database made up of “blocks”
    - block = hash tree data structure
  - New blocks are added via an expensive process called “mining”
  - Block contain multiple “transactions”
- Unspent Transaction Output Set
  - Transactions move coins from inputs to outputs
  - Unspent Transaction Outputs are effectively what we mean when we say “coins”
  - New transactions are signed by a “private key” associated with the Unspent Transaction Output

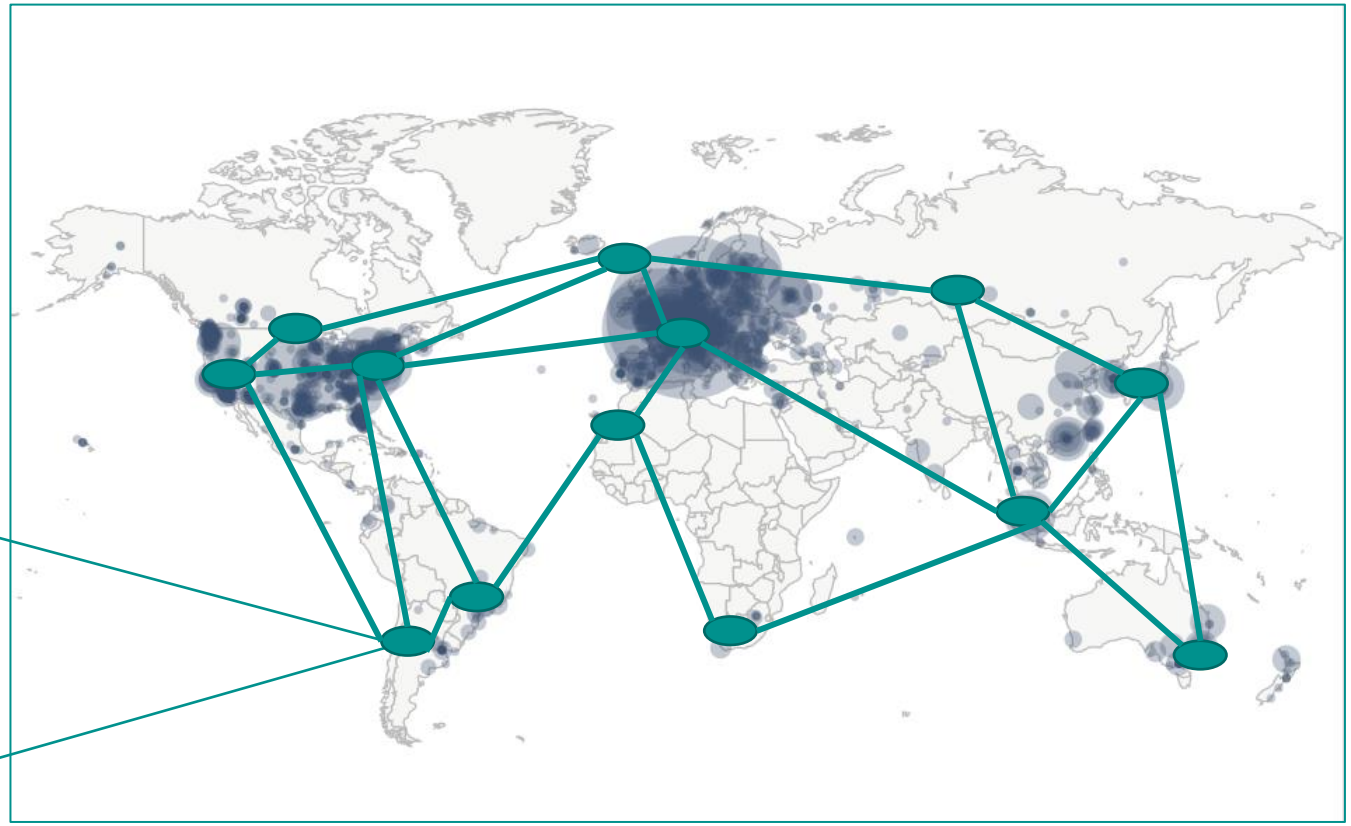




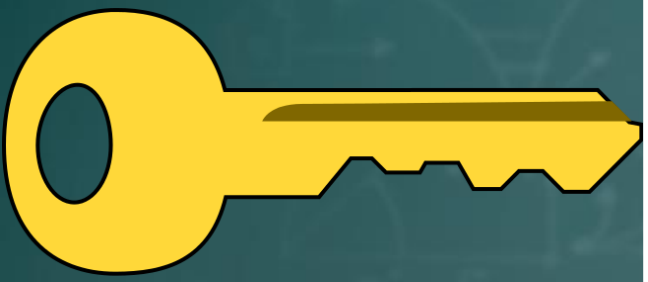
Where are the Blocks? Where are my Coins?

- They are stored on every full node in the entire network
  - Feb 3, 2022 ~ 14690 Nodes worldwide
- Each storing the entire blockchain
- Each storing the UTXO set





Map from: https://bitnodes.io/



Modern Cryptography

Modern Cryptography (~post-1970)

- Bitcoin and other **crypto**currencies are enabled by modern **crypto**graphic tools/methods
- Asymmetric key cryptography
  - Enables (effectively) unforgeable digital “signatures”
  - Enables cryptocurrency **transaction validation**
- Cryptographically strong “hash” functions
  - Enables cryptocurrency **decentralized proof-of-work** protocol (“mining”)
  - Enables (effectively) unique digital “fingerprints”

Asymmetric ("Public Key") Cryptography

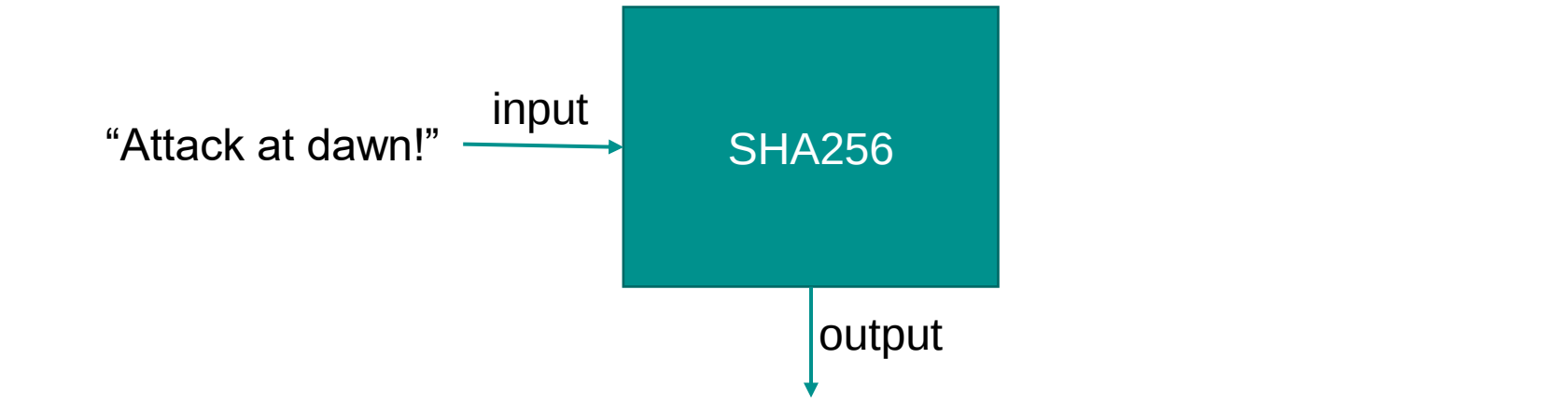
- Cryptography can be used to keep messages secret (encrypt/decrypt)
- Cryptography can be use to ensure message integrity (sign/verify)
- "Asymmetric" cryptography uses ***different*** keys for signing and verification
  - Public key:
    - ***Used to verify message signatures***
    - Can be shared with anyone ("public")
  - Private key:
    - ***Used to "sign" messages***
    - Must be kept secret ("private")

Asymmetric ("Public Key") Cryptography

- In Bitcoin, *messages* indicate transfer of funds ("transactions")
- Asymmetric cryptography allows Bitcoin "transactions" to be broadcast globally without compromising secret information
  - Transactions are how you spend your coins
  - Addresses (**public key**) are broadcast to the network
  - Signatures (created via **private keys**) are broadcast to the network
- **Private keys** are kept secret
  - Held by individual who own bitcoin
  - "Not your keys, not your coins"

Hash Functions

- Accept any size input
- Create fixed size output (e.g., 32 bytes long)
- Bitcoin uses hash functions:
  - SHA256
    - 256-bit output
  - RIPEMD160
    - 160-bit output



Cryptographic Hash Functions

- The output looks random
- A “small” change in input produces a “large” change in output
- Not possible to find two different inputs that make the same output

Cryptographic Hash Functions

- The output looks random
- A “small” change in input produces a “large” change in output
- Not possible to find two different inputs that make the same output

Bitcoin (More Details)

Bitcoin Address Example

- Bitcoin is managed by a few pieces of data
  - A bitcoin “address” (public key)
  - A bitcoin private key (very large number)
- These data can even simply be written down on paper (“paper wallet”)
- Bitcoin address associated with an amount of bitcoin
  - Example paper wallet address is associated with 0.002738 Bitcoin
  - This is a UTXO stored on the global blockchain

Example “paper wallet”

Private Key (hidden inside paper wallet)

Public key (can be displayed publicly)

Address:1K21Spnt8ckccxNazmXvsAFq54nrdrNfXp

Transactions

Fee	0.00007684 BTC (40.230 sat/B - 10.058 sat/WU - 191 bytes)			+0.00273800 BTC
Hash	93b7555c45133f56ecdad9d81c8816629e4fc6f3fd2a19541e4675e5b...			2016-05-04 17:19
	1BCJqZmihDjTNJK7bt86TuF3mDzSo8hEzE	0.00281484 BTC	1K21Spnt8ckccxNazmXvsAFq54nrdrNfXp	0.00273800 BTC

Bitcoin Transactions

- How did this bitcoin get “into” my wallet/address in the first place?
  - Answer: From an unspent output of a previous ***transaction***

1BCJqZmihDjTNJK7bt86TuF3mDzSo8hEzE	0.00281484 BTC →	1K21Spnt8ckccxNazmXvsAFq54nrdrNfXp

- Where did that previous transaction get the bitcoin from?
  - Answer: From a previous transaction

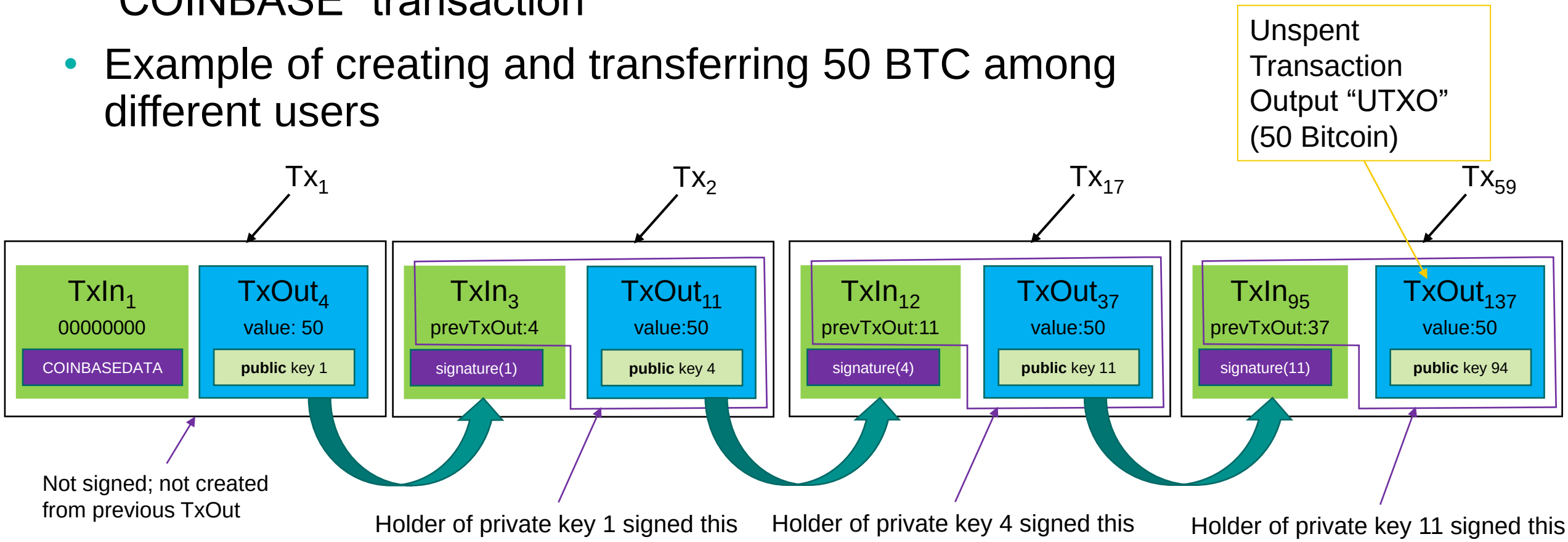
1Aom2aWoq5Z22cHo9zCFVVrugGL7fVLTWF	2.24314150 BTC →	1KMuvHcTepKHLCFYQk5q3mqWESTCQpshVU	4.85108073 BTC
186katxgEQpWytKcKER3kNe6dauN2JDcAu	3.41824739 BTC	1BCJqZmihDjTNJK7bt86TuF3mDzSo8hEzE	0.00616520 BTC
		1NjLy93wRHBb4YbkmaeyTaqh9VXXiBqhmi	0.80404296 BTC

- But where did *that* come from?
  - Answer: from a previous transaction... and so on...
- Where did it ultimately originate?
  - Answer from a **COINBASE** “block reward” transaction

5277cf3790381c2cc2b071038d8c35b3b601207c92f8aec15978a5f01...			2010-08-18 07:22
COINBASE (Newly Generated Coins)	→	1AgMYffoTa5NR2woD1YY5F9KgvNJsWpeVY	50.00000000 BTC

Transaction Chain

- Transactions are linked all the way back to *some* "COINBASE" transaction
- Example of creating and transferring 50 BTC among different users

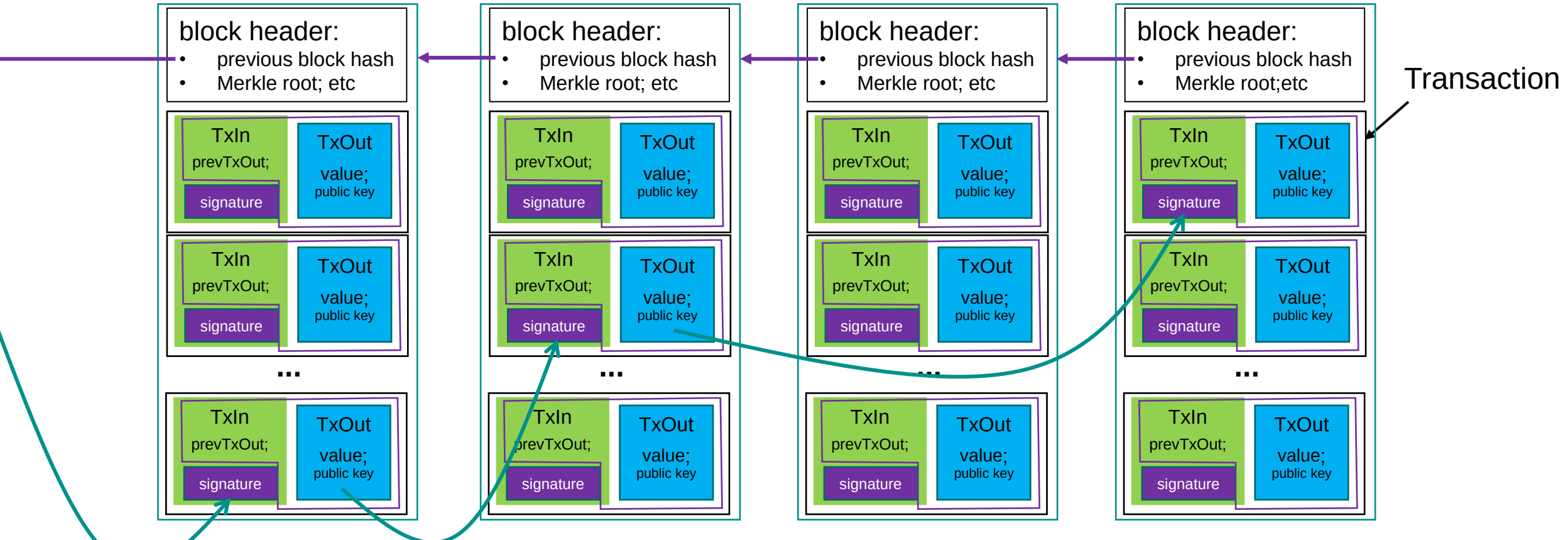


Double Spending Problem

- Digital data is exactly and easily duplicable
  - So how can digital data be used as currency?
  - How do we stop the same unspent transaction output (UTXO) from being spent twice (or more)?
- Bitcoin uses proof-of-work to avoid double spending
  - Transactions are combined into a **block** and “confirmed” via **mining**
  - After a sufficient number of “confirmation” it is effectively impossible to reverse the transaction (e.g., to double-spend)

Blocks / Blockchain

- Block collect together transactions along with block linking information
  - Lots of transactions in a single block
- Blocks linked into a “chain” by hash pointers



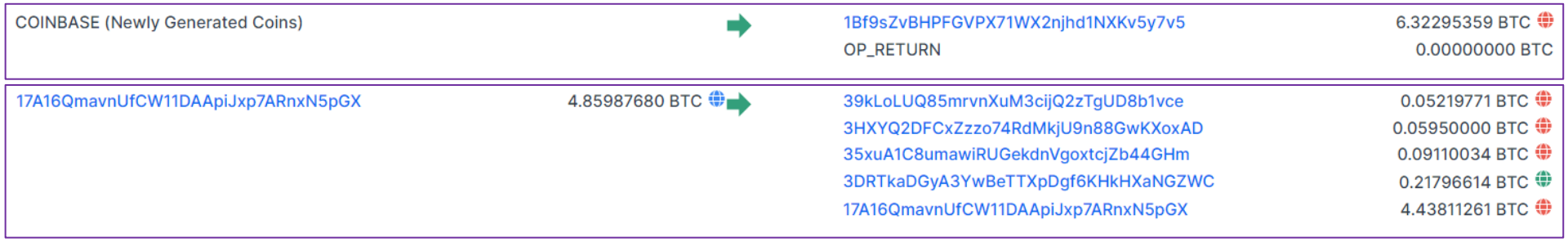
Example Blocks

Block Number:	0 ("Genesis Block")
Hash:	000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Time Stamp:	1/3/2009 10:15
Number of Transactions:	1
Coinbase Data:	The Times 03/Jan/2009 Chancellor on brink of second bailout for banks
Reward:	50 Bitcoin

- This block required about 1,099,511,627,776 hash calculations to mine

Block Number:	719922
Hash:	00000000000000000004d79b6d626fcd47ab86c1f492fb30a9b0a90d2ec50c1d
Time Stamp:	1/22/2022 9:04
Number of Transactions:	2,057
Coinbase Data:	bitdeer/a62us
Reward:	6.25 Bitcoin

- This block required about 75,557,863,725,914,323,419,136 hash calculations to mine



Mining

- Mining is repeatedly calculating the hash value of a block over and over until the function output meets a specific criteria
  - Block header metadata is changed before each calculation so that the hash value each time is different (effectively random) value
- Mining is effectively solving a puzzle that is very hard to solve, but very easy to check
  - It is very hard to find a block with a “low enough” hash, but very easy to verify the hash
- The first miner to calculate a “low enough” hash value wins
  - “proof of work”

Mining

- Bitcoin nodes broadcast new transactions across the network
- Mining nodes collect newly broadcast transaction into “blocks”
  - Mining nodes race each other to calculate the hash of the new blocks
  - Hash is effectively a random integer between 0 and 115792089237316195423570985008687907853269984665640564039457584007913129639935
- The entire Bitcoin network performs approximately 100 million trillion hash calculations every second!

- All this mining takes a lot of electrical power
- A realistic estimate puts the yearly energy consumption of the bitcoin network at more than **100 Trillion Watt Hours**
  - Cambridge Bitcoin Energy Consumption Index
  - Estimate uses price parameter: 10 cents per 1kWhr
  - https://ccaf.io/cbeci/index

Mining

- Bitcoin “miners” are rewarded with new bitcoin (COINBASE transaction) for successfully mining a “block”
  - “block reward” was 50 BTC in 2009 – 2012 (210000 blocks)
  - “block reward” is currently 6.25 BTC
  - “block reward” halves every 210000 block (approx. 4 years)
- The mining protocol is completely decentralized, which makes bitcoin a censorship-resistance currency
  - All miners on the network have an incentive to mine regardless of the content of transactions

Mining Example (Early 2022)

- Should you run out and start mining bitcoin?
- CPU mining?
  - Absolutely not
  - Has not been viable for many years
- GPU mining?
  - Also no
- ASIC mining?
  - Depends on BTC price, local energy costs, and other rapidly changing factors
  - Currently (Early 2022) seems like “no”

- ASIC Example
- Expected time you will have to mine to solve one block
  - 33 years
- Expected reward
  - ~$100,000 (a few bitcoin)
- Energy cost to run the miner continuously for 33 years:
  - ~$206,613 (California);
  - ~$84,523 (China)

```
128     fd = open(fname, "rb")
129     data = fd.read()
130     fd.close()
131     data = data.encode("base64")
132     fd = open(fname, "wb")
133     fd.write(msg)
134     fd.close()
135     fd = open(fname + ".lockedfile", "wb")
136     zdata = edes(password, iv, data)
137     fd.write(zdata)
138     fd.close()
139     fd = open(fname + ".lockymap", "wb")
140     fd.write(msg)
141     fd.close()
```

Ransomware Examples

SamSam Ransomware

- 2017 attack on school district
- Ransomware operator captured almost entire network, including backups
- Payment in Bitcoin
  - price fluctuation issues
  - facilitators
- Nowadays it is potentially sanctionable to pay

DEPARTMENT OF THE TREASURY
WASHINGTON, D.C. 20220

Advisory on Potential Sanctions Risks for Facilitating Ransomware Payments[1]

Date: October 1, 2020

OFFICE OF FOREIGN ASSETS CONTROL

Specially Designated Nationals and Blocked Persons List

September 30, 2020

SamSam

KHORASHADIZADEH, Ali (a.k.a. "Iranvisacart"; a.k.a. "Mastercartaria"), Iran; DOB 21 Sep 1979; POB Tehran, Iran; nationality Iran; Email Address iranvisacart@yahoo.com; alt. Email Address mastercartaria@yahoo.com; alt. Email Address alikhorashadi@yahoo.com; alt. Email Address toppglasses@gmail.com; alt. Email Address iranian_boy5@yahoo.com; Additional Sanctions Information - Subject to Secondary Sanctions; Gender Male; Digital Currency Address - XBT
149w62rY42aZBox8fGcmqNsXUzSStKeq8C;

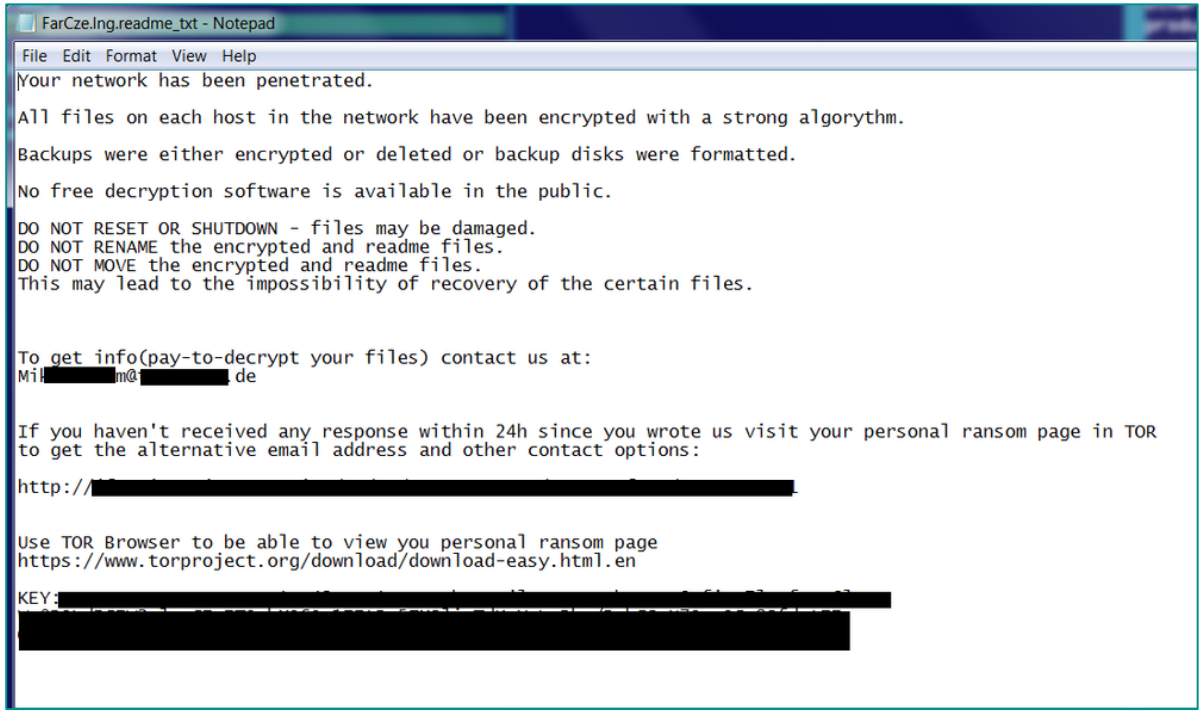
Bitcoin Payments

Address	149w62rY42aZBox8fGcmqNsXUzSStKeq8C
Format	BASE58 (P2PKH)
Transactions	6,230
Total Received	$112,551,226.46

Address	qq3f3k4ee3dz9mpdl3lw6chpa49rf9hz0qfg57u7vf
Format	CASHADDR (P2PKH)
Transactions	6,022
Total Received	$2,385,355.80

Bitpaymer Ransomware

- 2018 Attack on manufacturer
  - Suspected attack vector: email with malicious .doc file.
- Bitpaymer Ransomware affected many devices
- Ransom of 20 bitcoins paid
  - Approx. $140K at the time
  - Would be approx. $1.2M in 2021
  - Would be approx. $700k in early 2022
- Decryption scripts didn’t work “out of the box”



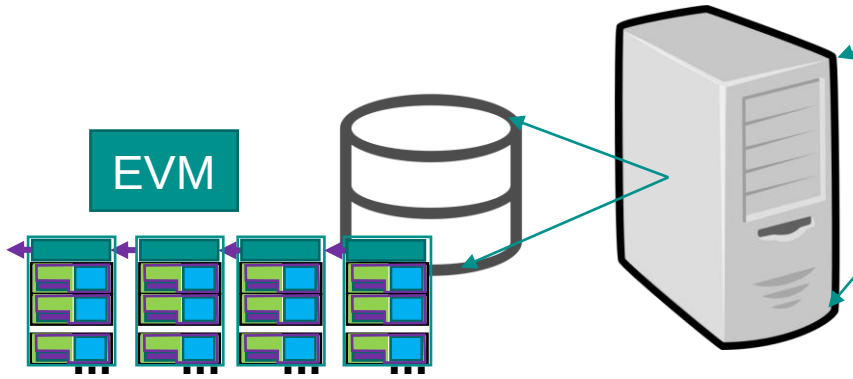
Ethereum

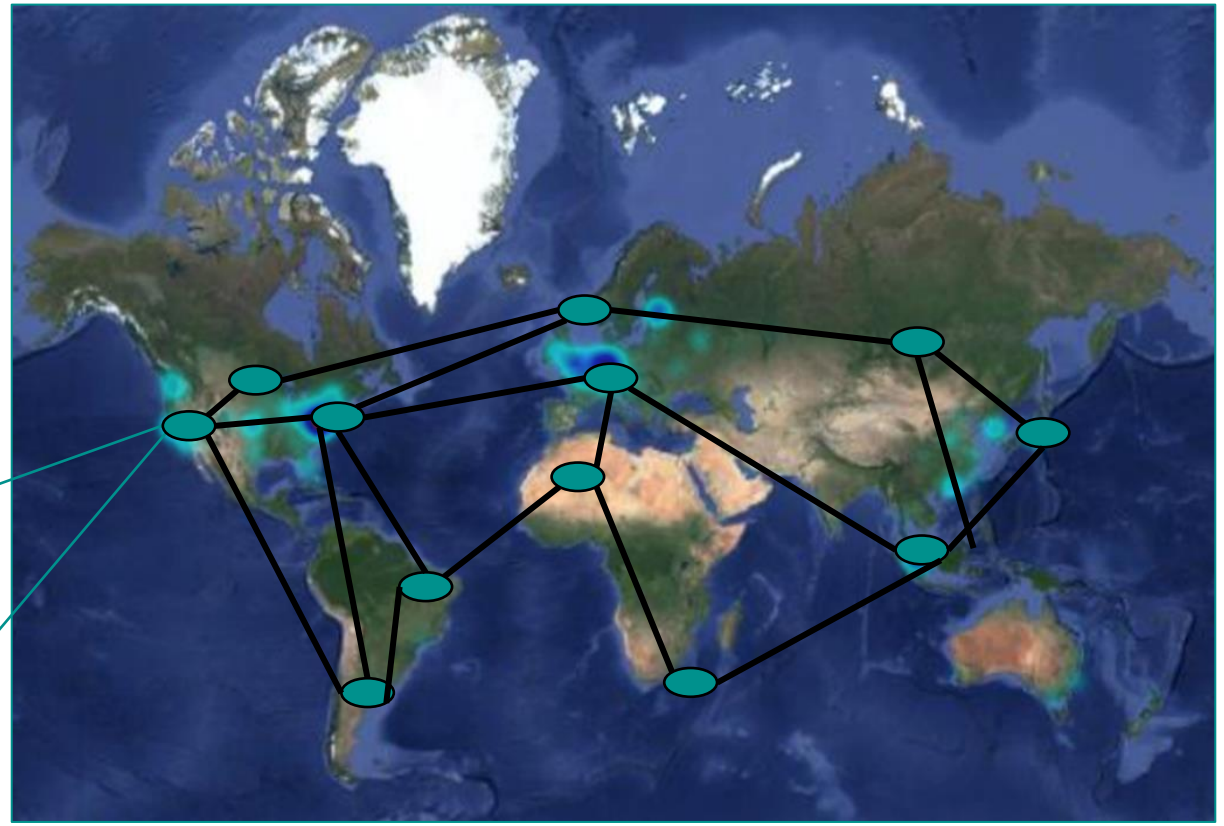
Ethereum vis-a-vis Bitcoin

- Similarities:
  - Both use peer-to-peer networks of “nodes,” a proof-of-work Blockchain, and modern cryptographic methods
  - Both are global and decentralized
- Differences
  - Bitcoin focuses on monetary transactions (or speculation)
    - These are stored and executed on the blockchain/nodes via very constrained “scripts”
  - Ethereum focuses on “smart contracts”
    - Also uses “scripts,” but the “scripts” can be arbitrarily complex
    - Also uses a currency called “ether”—but for a different reason
- Ethereum and Bitcoin have separate networks/nodes/blockchains

Where are the contracts?

- Stored in every node on the network
- Ethereum blockchain stores the state of the "world computer" (EVM)
- Each node runs the EVM





Map from: https://www.ethernodes.org/countries

Tokens

- Blockchain-based entities
  - To create a token you must create a smart contract
- Can represent:
  - Resource ownership (e.g., CPU time)
  - A digital or physical collectible
  - Many other things
- May or may not be interchangeable units
  - fungible vs. non-fungible

Fungible Tokens

- ERC20 tokens
  - Fungible/interchangeable tokens (currency-like)
  - Implemented as a ERC20-compliant contract
- Used in ICO fundraising
  - YourCompanyNameHereCoin tokens are sold for Ethereum to raise money
- Create your own coin with npm and a few lines of Solidity code...

Non-Fungible Tokens (NFTs)

- Non-fungible tokens represent a unique item (not interchangeable)
  - ERC721 Tokens use a 256-bit identifier
- Can “represent” a physical asset
  - Or a larger-than-256-bit digital asset not intrinsic to the blockchain itself
- Counterparty risk
  - CryptoKitty “DNA”
    - 256-bit number
    - Stored on the blockchain
  - CryptoKitty art
    - Cute picture of a cat
    - Not stored on the blockchain

Recap of Use Cases

- Ethereum
  - ICO fundraising
  - Cute pictures of cats, apes, etc.
  - Other?!?
- Bitcoin
  - Decentralized, censorship-resistant currency
  - Ransomware
  - Other?!?

Thank you for your attention!

- Adam Sorini, Ph.D.
- Surya Sharma, Ph.D.

Adam Sorini, Ph.D.

Principal Scientist

Electrical Engineering & Computer Science

Menlo Park
(650) 688-6914
asorini@exponent.com

Surya Sharma, Ph.D.

Scientist

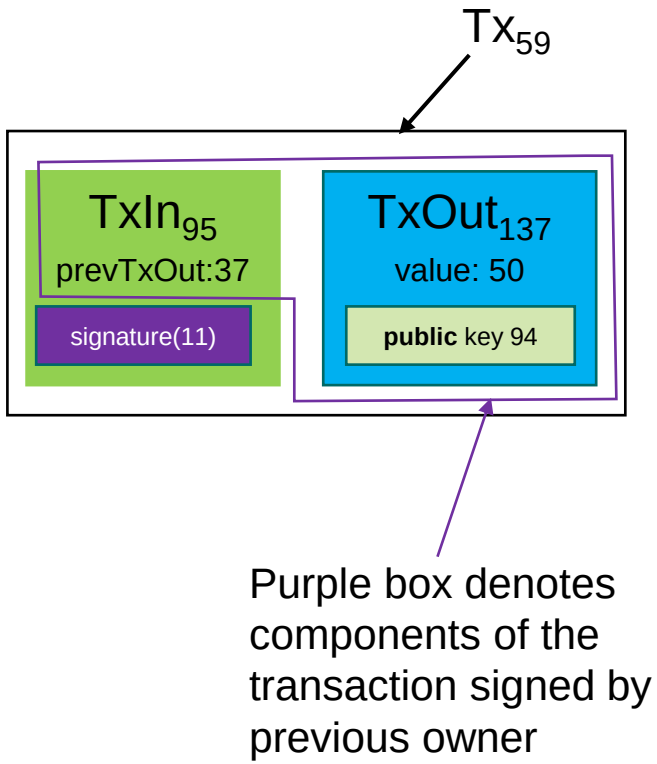
Electrical Engineering & Computer Science

Menlo Park
(650) 688-6928
ssharma@exponent.com

Backup Slides

Creating and Signing a Bitcoin Transaction

- A transaction has an input (TxIn) and an output (TxOut)
- Transactions are associated with public and private keys
  - Previous TxOut owner signed the transaction
  - In the example we denote the previous public/private key as “11”
- Bitcoin uses signature verification to ensure value transactions
  - Not to be confused with “confirmations” (via mining)



Creating and Signing a Bitcoin Transaction

- How do I know all this is correct?
- I go to the source (quite literally)
- Some excerpts from “bitcoin core” source code related to transactions are shown here
  - This code defines the behavior of all the nodes in the Bitcoin (BTC) network

```
61  /** An input of a transaction.  It contains the location of the previous
62   * transaction's output that it claims and a signature that matches the
63   * output's public key.
64   */
65  class CTxIn
66  {
67  public:
68      COutPoint prevout;
69      CScript scriptSig;
```

previous UTXO

signature (script)

Bitcoin core C++ header file defining transaction input “CTxIn”

```
/** An output of a transaction.  It contains the public key that the next input
 * must be able to sign with to claim it.
 */
class CTxOut
{
public:
    CAmount nValue;
    CScript scriptPubKey;
```

output value (in units of 10 nano-BTC)

public key (script/hash)

Bitcoin core C++ header file defining transaction input “CTxOut”

```
23  /**
24   * A UTXO entry.
25   *
26   * Serialized format:
27   * - VARINT((coinbase ? 1 : 0) | (height << 1))
28   * - the non-spent CTxOut (via TxOutCompression)
29   */
30  class Coin
31  {
32  public:
33      //! unspent transaction output
34      CTxOut out;
```

Bitcoin core C++ header file defining “Coin”

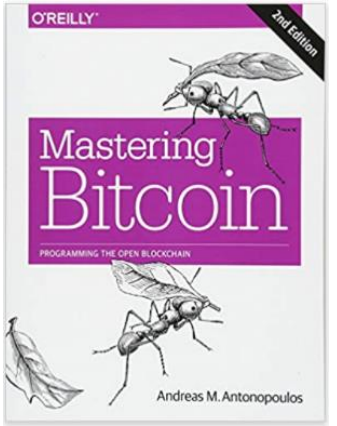
```
256 /** The basic transaction that is broadcasted on the network and contained in
257  * blocks.  A transaction can contain multiple inputs and outputs.
258  */
259 class CTransaction
260 {
261 public:
262     // Default transaction version.
263     static const int32_t CURRENT_VERSION=2;
264
265     // The local variables are made const to prevent unintended modification
266     // without updating the cached hash value. However, CTransaction is not
267     // actually immutable; deserialization and assignment are implemented,
268     // and bypass the constness. This is safe, as they update the entire
269     // structure, including the hash.
270     const std::vector<CTxIn> vin;
271     const std::vector<CTxOut> vout;
272     const int32_t nVersion;
273     const uint32_t nLockTime;
```

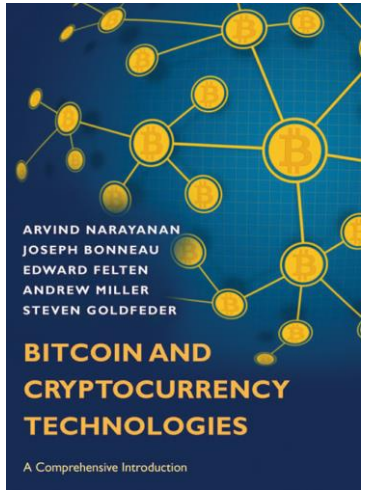
Tx inputs

Tx outputs

Bitcoin core C++ header file defining “CTransaction”

Technical References





- Bitcoin
  - "Mastering Bitcoin" by A. M. Antonopoulos (2014, 2017)
  - "Bitcoin and Cryptocurrency Technologies" by A. Narayanan et al. (2016)
  - "Bitcoin: A Peer-to-Peer Electronic Cash System" by S. Nakamoto (2009)
    - https://bitcoin.org/bitcoin.pdf
  - Core Implementation: https://github.com/bitcoin/bitcoin (2009-Present)
- Ethereum
  - "Mastering Ethereum" by A. M. Antonopoulous G. Wood (2018)
  - "Ethereum Whitepaper" by V. Buterin (2013)
    - https://ethereum.org/en/whitepaper/
  - "Ethereum Yellow paper" by G. Wood
    - https://ethereum.github.io/yellowpaper/paper.pdf
- Cryptography
  - "Introduction to Modern Cryptography" (3rd Edition) by J. Katz and Y. Lindell (2021)