Blockchain Principles and Applications

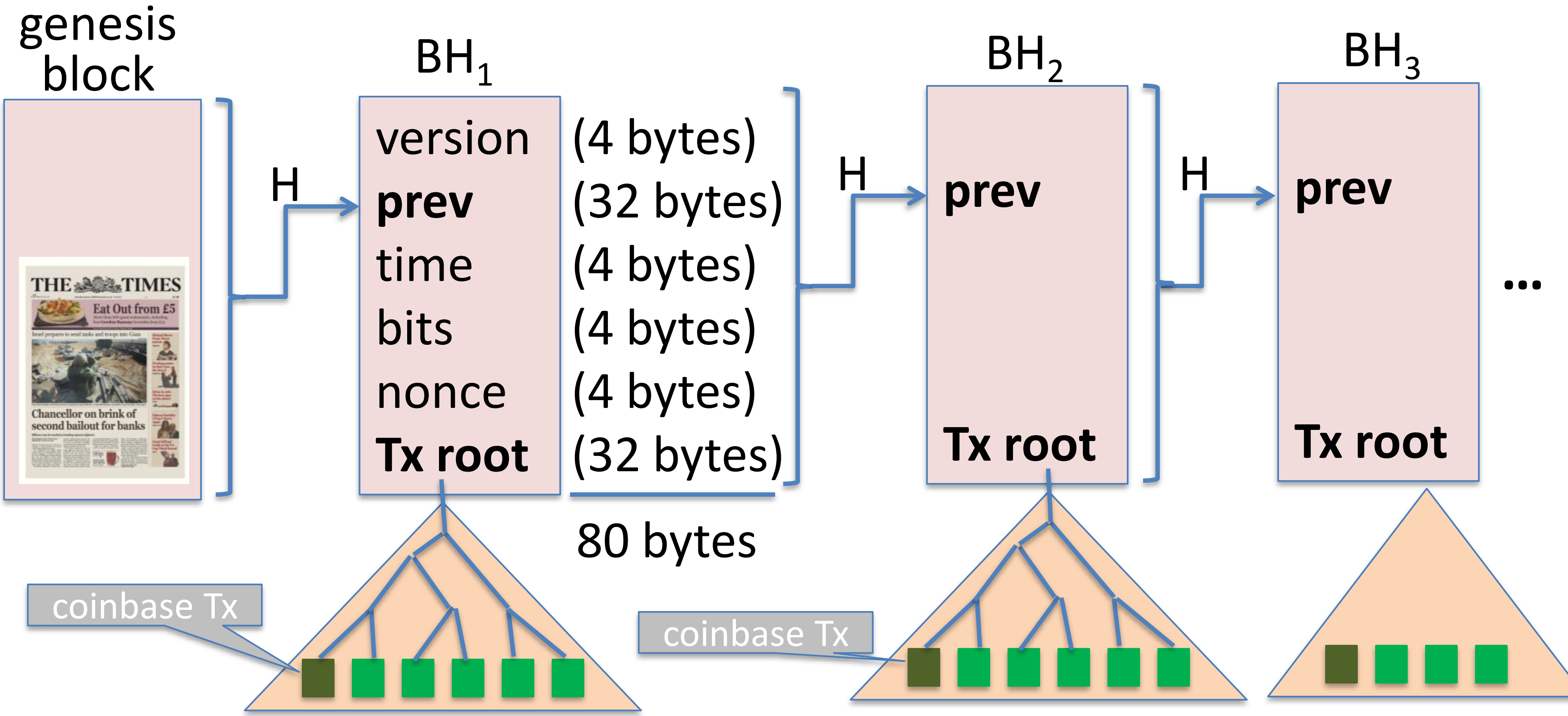
Amir Mahdi Sadeghzadeh, PhD

Data and Network Security Lab (DNSL)
Trustworthy and Secure AI Lab (TSAIL)

Most slides have been adapted from ECE/COS 470, Pramod Viswanath, Princeton 2024 and CS251, Dan Boneh, Stanford 2023.

Recap

Bitcoin blockchain: a sequence of block headers, 80 bytes each



Bitcoin blockchain: a sequence of block headers, 80 bytes each

time: time miner assembled the block. Self reported.
(block rejected if too far in past or future)

bits: proof of work difficulty
nonce: proof of work solution

for choosing a proposer

Merkle tree: payer can give a short proof that Tx is in the block

new block every ≈10 minutes.

An example

Tx data

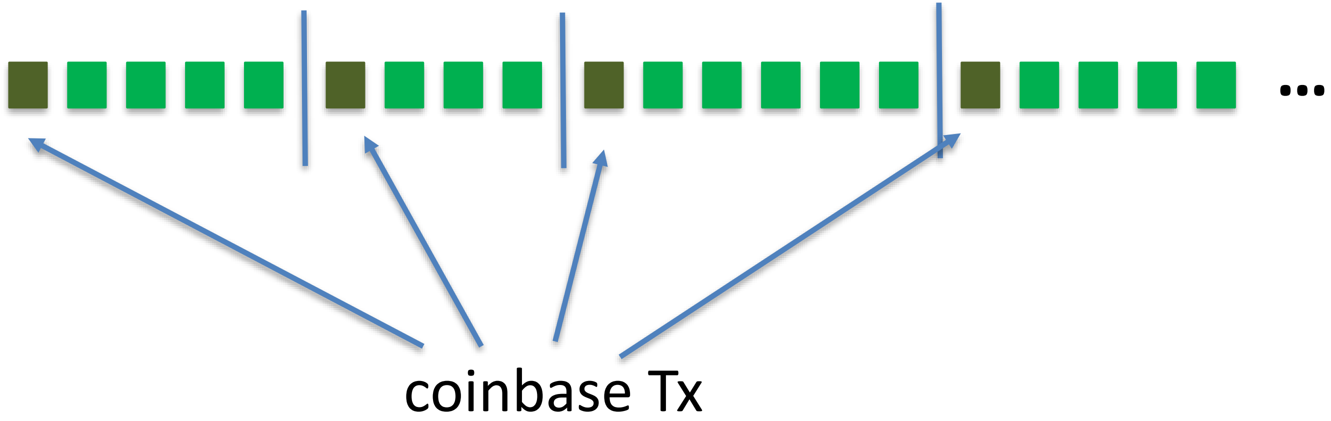
Height	Mined	Miner	Size	#Tx
648494	17 minutes	Unknown	1,308,663 bytes	1855
648493	20 minutes	SlushPool	1,317,436 bytes	2826
648492	59 minutes	Unknown	1,186,609 bytes	1128
648491	1 hour	Unknown	1,310,554 bytes	2774
648490	1 hour	Unknown	1,145,491 bytes	2075
648489	1 hour	Poolin	1,359,224 bytes	2622

Block 648493

Timestamp	2020-09-15 17:25	
Height	648493	
Miner	SlushPool	(from coinbase Tx)
Number of Transactions	2,826	
Difficulty (D)	17,345,997,805,929.09	(adjusts every two weeks)
Merkle root	350cbb917c918774c93e945b960a2b3ac1c8d448c2e67839223bbcf595baff89	
Transaction Volume	11256.14250596 BTC	
Block Reward	6.25000000 BTC	
Fee Reward	0.89047154 BTC	(Tx fees given to miner in coinbase Tx)

This lecture

View the blockchain as a sequence of Tx   (append-only)



Tx structure (non-coinbase)

inputs	input[0] input[1] input[2]
outputs	output[0] output[1]
(segwit)	witnesses (part of input)
(4 bytes)	locktime

earliest block # that can include Tx

TxID = H(Tx)
(excluding witnesses)

input:

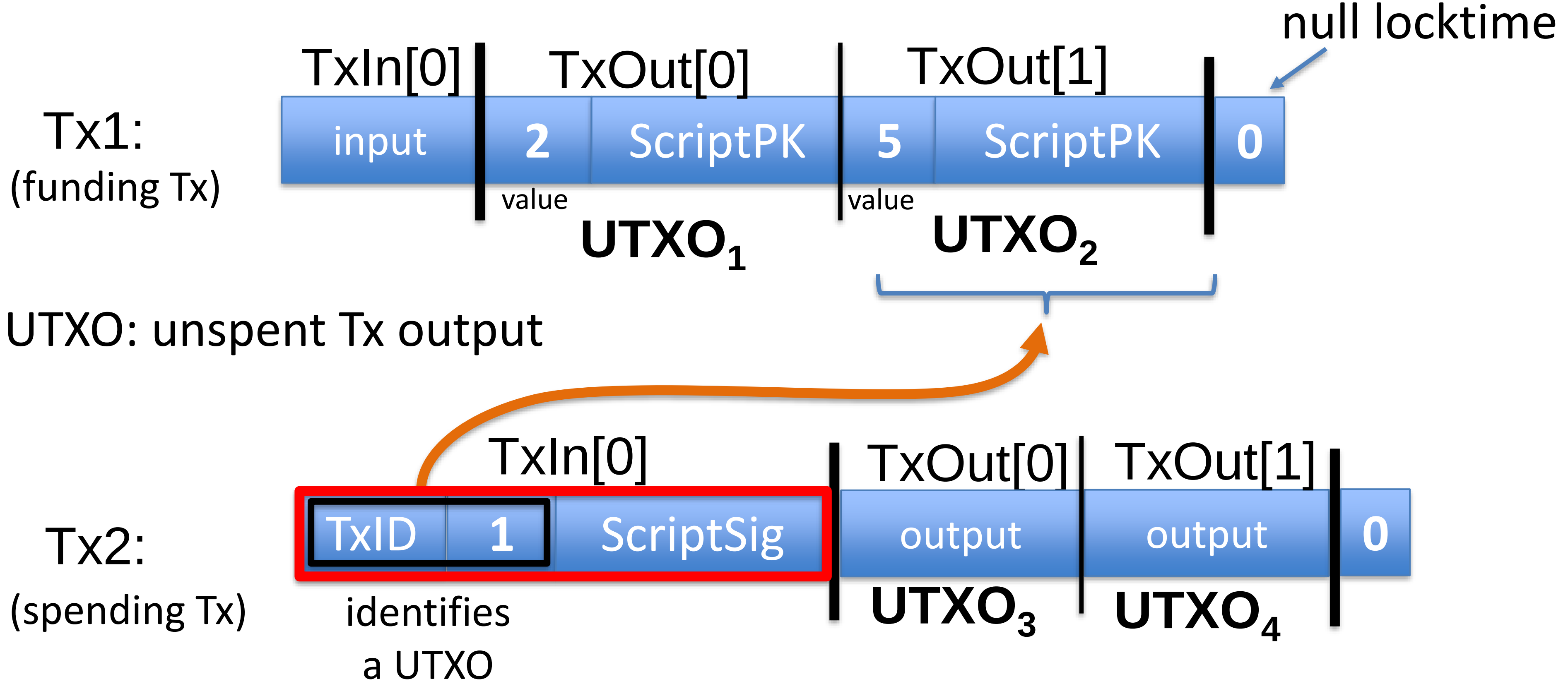
TxID	32 byte hash
out-index	4 byte index
ScriptSig	program
seq	ignore

output:

value	8 bytes
ScriptPK	program

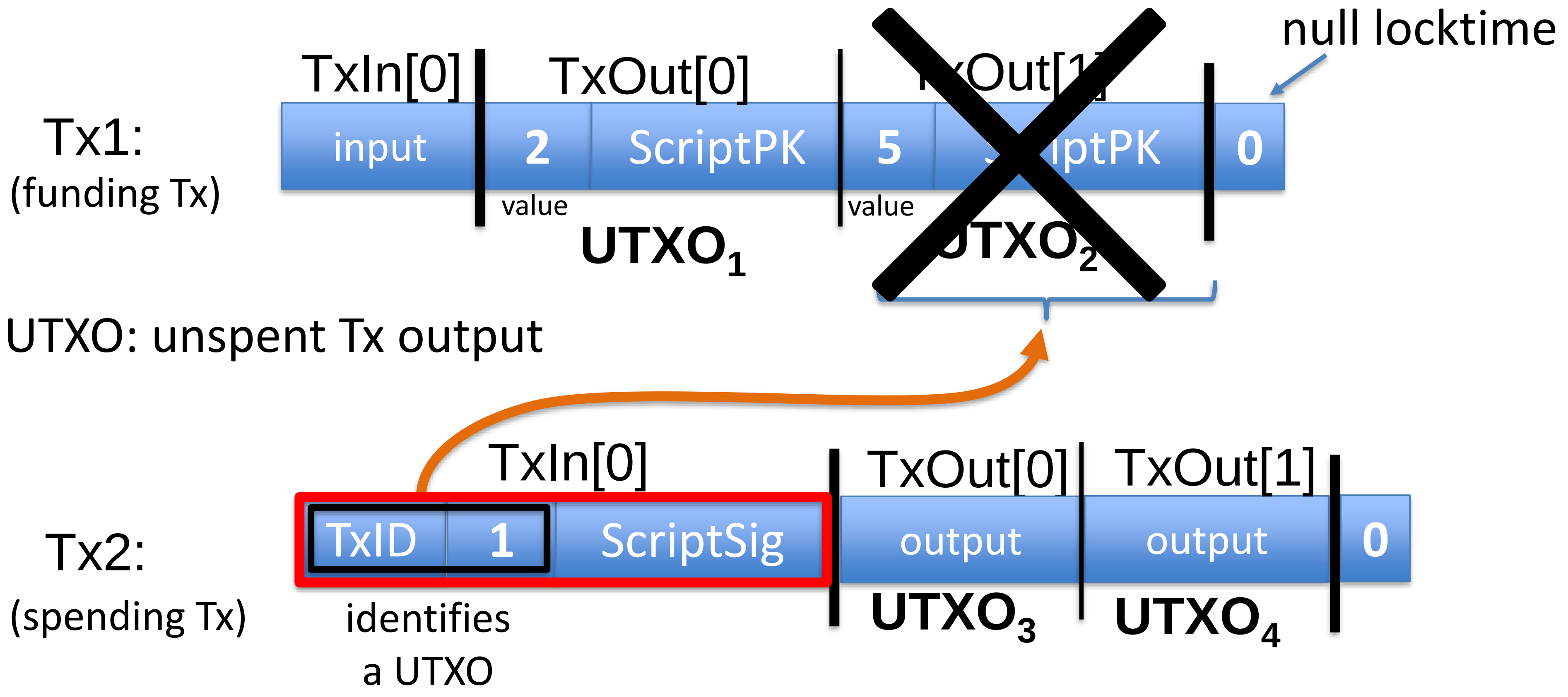
$\#BTC = value/10^8$

Example



null locktime
TxIn[0]
TxOut[0]
TxOut[1]
Tx1:
(funding Tx)
input
2
ScriptPK
5
ScriptPK
0
value
value
UTXO1
UTXO2
UTXO: unspent Tx output
TxIn[0]
TxOut[0]
TxOut[1]
Tx2:
(spending Tx)
TxID
1
ScriptSig
output
output
0
UTXO3
UTXO4
identifies
a UTXO

Example


null locktime
TxIn[0]
TxOut[0]
TxOut[1]
Tx1:
(funding Tx)
input
2
ScriptPK
5
ScriptPK
0
value
value
UTXO$_1$
UTXO$_2$
UTXO: unspent Tx output
TxIn[0]
TxOut[0]
TxOut[1]
Tx2:
(spending Tx)
TxID
1
ScriptSig
output
output
0
identifies
a UTXO
UTXO$_3$
UTXO$_4$

Validating Tx2

program from funding Tx:
under what conditions
can UTXO be spent

Miners check (for each input):

1. The program $\boxed{\text{ScriptSig | ScriptPK}}$ returns true

2. $\boxed{\text{TxID | index}}$ is in the current UTXO set

3. sum input values $\geq$ sum output values

After Tx2 is posted, miners remove $UTXO_2$ from UTXO set

An example (block 648493) [2826 Tx]

COINBASE (Newly Generated Coins)		→	1CK6KHY6MHgYvmRQ4PAafKYDrg1ejbH1cE	7.14047154 BTC
			OP_RETURN	0.00000000 BTC
			OP_RETURN	0.00000000 BTC
Tx0				
0.00000000 BTC			6.25 + Tx fees =	7.14047154 BTC

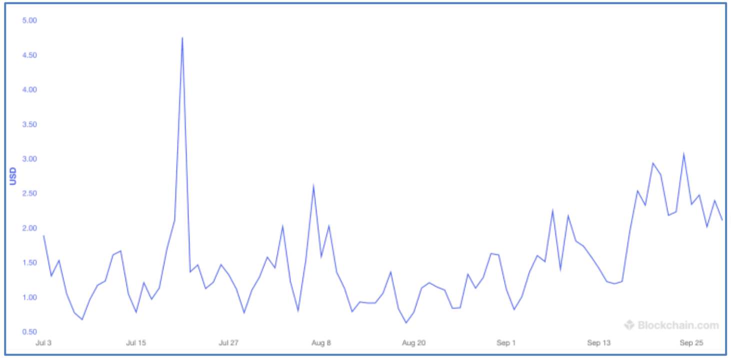
3PuJbxJS1pKxf8EdVR18yBkD1fPAbgUtyw	0.72333974 BTC	→	1E5Ao1VUnA5BhffvXf2Xmud6avUgwkFnJv	0.00917379 BTC
input	(input UTXO value)		bc1qr8k3e0vx06lpu3j7m858pa2ak9tyr56ttwvefk	0.61504199 BTC
			bc1qdrxve8kua3yz5dgx6wf3u95ngh0d3e648...	0.09290152 BTC
Tx1			14ZhjuXpQ5jCDjtAy7ZMu3hfEQCWewzLw7	0.00616444 BTC
0.00005800 BTC (Tx fee)			outputs	0.72328174 BTC

17MWze4Z1uP1jnvqvj7SAnGtxcoVq11H8A	0.05000000 BTC	→	3G3C2RFQ8gsf77EQpdR4ZReChWFKEHhxVU	0.04808000 BTC
Tx2				
0.00192000 BTC (Tx fee)				0.04808000 BTC

sum of fees in block added to coinbase Tx

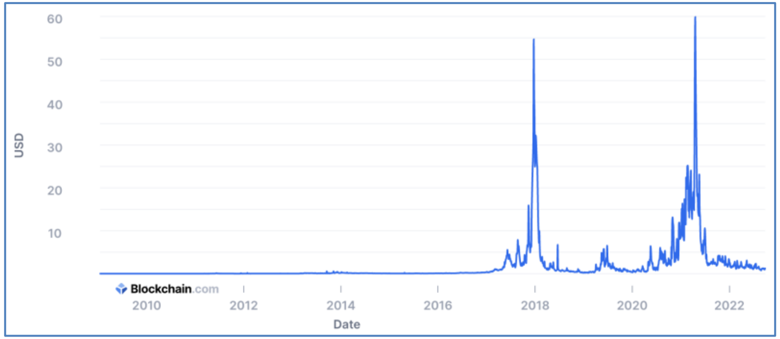
Tx fees

Bitcoin average Tx fees in USD (last 60 days, sep. 2023)

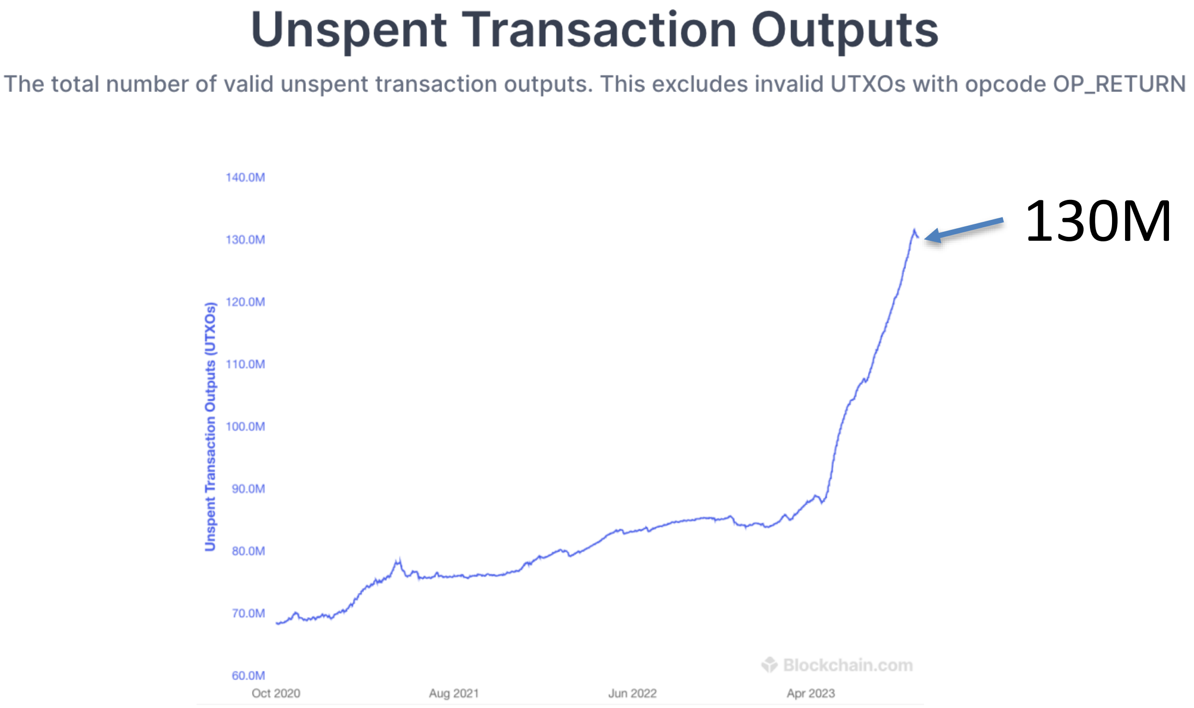


$2.11

Bitcoin average Tx fees in USD (all time)



All value in Bitcoin is held in UTXOs



Sep. 2023: miners need to store ≈130M UTXOs in memory

Focusing on Tx2: TxInp[0]

from UTXO
(Bitcoin script)

Value	0.05000000 BTC
Pkscript	OP_DUP OP_HASH160 45b21c8a0cb687d563342b6c729d31dab58e3a4e OP_EQUALVERIFY OP_CHECKSIG
Sigscript	304402205846cace0d73de82dfbdeba4d65b9856d7c1b1730eb401cf4906b2401a69bdc90220589d36d36be64e774c8796b96c011f29768191abeb7f56ba20ffb0351280860c01 03557c228b080703d52d72ead1bd93fc72f45c4ddb4c2b7a20c458e2d069c8dd9e

from TxInp[0]

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide
    1. There should be a unit of currency/money.

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide
  1. There should be a unit of currency/money.
  2. There should be a standard way of keeping accounts, i.e., keeping track of how much money each person owns, and transferring money between accounts.

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide
  1. There should be a unit of currency/money.
  2. There should be a standard way of keeping accounts, i.e., keeping track of how much money each person owns, and transferring money between accounts.
  3. No user should be able to create new money from thin air. There should be a fixed amount of money in the system at any given time, and new money should be introduced in a systematic manner.

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide
  1. There should be a unit of currency/money.
  2. There should be a standard way of keeping accounts, i.e., keeping track of how much money each person owns, and transferring money between accounts.
  3. No user should be able to create new money from thin air. There should be a fixed amount of money in the system at any given time, and new money should be introduced in a systematic manner.
  4. A user should not be able to spend more money than he/she owns. There should be a way to verify whether or not this happens.

Currency/banking system

- In any currency/banking system, there are some basic requirements that the system must provide
  1. There should be a unit of currency/money.
  2. There should be a standard way of keeping accounts, i.e., keeping track of how much money each person owns, and transferring money between accounts.
  3. No user should be able to create new money from thin air. There should be a fixed amount of money in the system at any given time, and new money should be introduced in a systematic manner.
  4. A user should not be able to spend more money than he/she owns. There should be a way to verify whether or not this happens.
  5. One user should not be able to spend someone else's money (at least, not without their permission).

Bitcoin and Satoshi

- The basic unit of currency in the Bitcoin system is Bitcoin, and the smallest denomination is called a Satoshi, which is equal to 10^{-8} Bitcoins.
- All transactions in Bitcoin must be an integer multiple of a Satoshi.

Transactions

- The term transaction refers to an exchange of something of value.
- In the context of Bitcoin and cryptocurrencies
  - A transaction is simply a message that specifies the transfer of money from one entity to another.
- Transactions are the data values that get recorded on the blockchain.
- The blockchain as a ledger is therefore an ordered list of transactions.

Coinbase transactions

- Introduction of New Bitcoins:
  - New Bitcoins are introduced into the system as a reward for miners who successfully mine a new block.
  - Every block includes a special transaction known as the ”coinbase transaction,” which allows the miner to claim a fixed number of newly created Bitcoins.

Coinbase transactions

- Block Reward Halving:
  - Initially, when Bitcoin was launched, the block reward for miners was 50 BTC per block.
  - Approximately every 210,000 blocks (about four years), the block reward is halved through an event known as ”halving.”
    - This means that after each halving, miners receive half the number of Bitcoins as the previous reward.

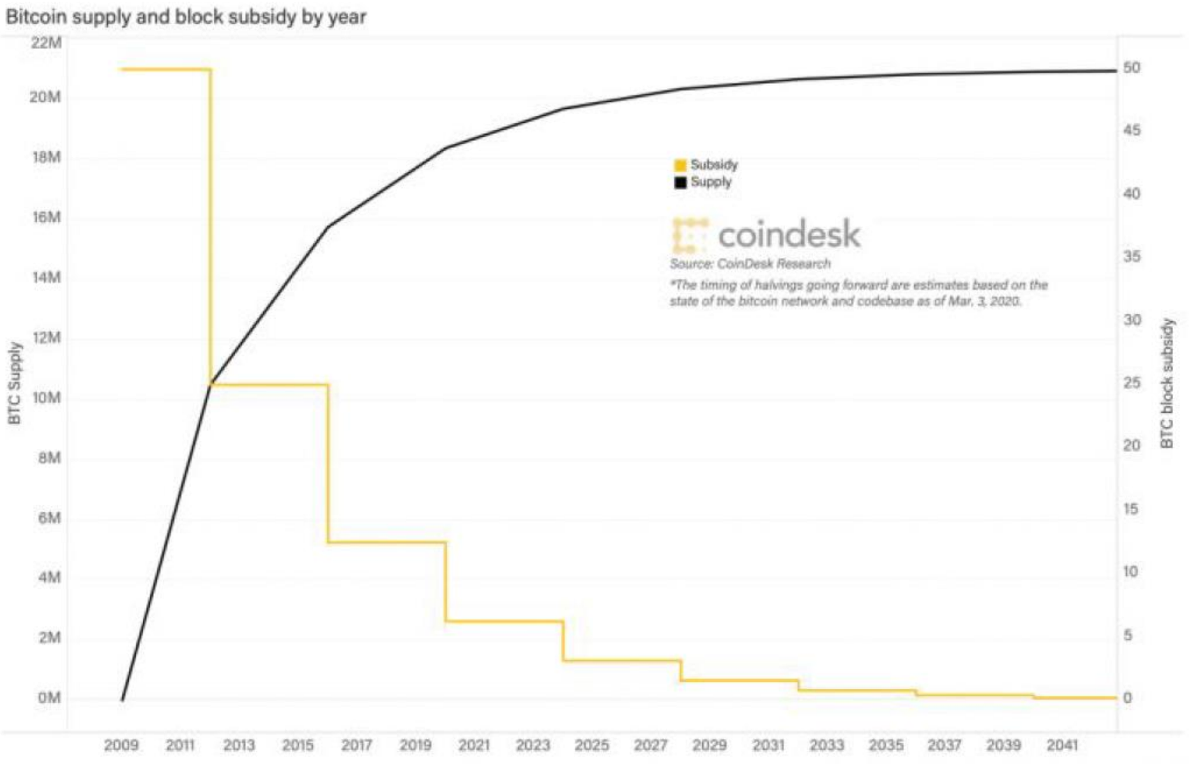
Coinbase transactions

- Current Block Reward:
  - As of now, there have been four halvings, and the current block reward for miners is 3.13 BTC per block.
  - The block rewards will continue until the year 2140 when the total supply of Bitcoin will reach its cap.

Coinbase transactions

- Fixed Supply of Bitcoin:
  - The total supply of Bitcoin is capped at 21 million coins.
  - This fixed supply ensures that there will never be more than 21 million Bitcoins in circulation.
  - Approximately 19.9 million coins are already in circulation, and the remaining Bitcoins will be gradually introduced through coinbase transactions until the cap is reached.

Coinbase transactions



Coinbase transactions

- Incentive Mechanism:
  - Coinbase transactions, along with transaction fees, serve as incentives for miners to actively participate in the network and secure the blockchain through the proof-of-work process.
  - In the early years of Bitcoin, coinbase transactions formed a significant portion of the rewards for miners, but over time, the contribution of transaction fees has increased.

Addressing

- Bitcoin Address and Its Generation:
  - In Bitcoin, the notion of traditional accounts is replaced with addresses.
  - An address is simply the hash of a user's public key. It is a unique alphanumeric string that serves as a destination for receiving bitcoins.
  - Addresses are also used to determine where bitcoins will be sent in a transaction.
  - New pairs of public and private keys, and thus new addresses, can be generated at will by a single user.

Addressing

- Receiving vs. Spending Coins:
  - To receive coins, a user only needs to share their Bitcoin address with others. The address serves as a public identifier for receiving funds.
  - However, to spend coins, a user must also reveal the corresponding public key associated with the address.
    - This is because spending requires providing proof of ownership through a digital signature, which is created using the user's private key.
  - The idiosyncrasy lies in the fact that while an address (hash of the public key) is publicly visible and used for receiving the associated public key is only revealed during the spending process for cryptographic verification.

Transaction inputs and outputs

- Transaction Inputs:
  - Each transaction input represents the amount of Bitcoin being spent from a specific address.
    - When a user initiates a transaction, they reference one or more previous unspent transaction outputs (UTXOs) from the blockchain that they have the right to spend.
      - These UTXOs serve as the inputs to the new transaction and determine the source of the funds being spent.
      - Each input includes a reference to the UTXO's transaction ID and its output index, along with a cryptographic signature to prove ownership.

Transaction inputs and outputs

- Transaction Outputs:
  - Each transaction output represents the amount of Bitcoin being received by a specific address.
  - When a transaction is created, it typically includes multiple outputs, each specifying the amount of Bitcoin and the recipient's address.
  - Each output **locks** the specified amount of Bitcoin to the recipient's address using a locking script that can only be **unlocked** with the corresponding private key.

Transaction inputs and outputs

- Balance Consistency:
  - In every valid transaction, the total amount of Bitcoin being spent (sum of inputs) must equal the total amount being received (sum of outputs).
  - This ensures that the transaction preserves the overall balance of the Bitcoin system, i.e., no new bitcoins are created, and no bitcoins are lost during the transaction process.

Signatures on transactions

- Signing Transactions for Safety:
  - Each transaction in Bitcoin must be signed by the users who are spending money.
    - This process is a fundamental safety feature that prevents unauthorized access to someone's funds.
  - For every transaction input, the user creating the transaction must sign it using the corresponding private key associated with the address from which the funds are being spent.
  - By signing the transaction, the user proves ownership of the private key and authorizes the transfer of funds.

Signatures on transactions

- One-to-One Correspondence: Public and Private Keys:
  - As mentioned earlier, each address in Bitcoin has a one-to-one correspondence with a public key, and each public key has a corresponding private key.
  - When a user wishes to spend Bitcoins associated with a particular address, they create the transaction and then sign it using the private key linked to that address.

Signatures on transactions

- Verification of Signatures:
  - After a user signs a transaction, they broadcast it to the network along with the corresponding public key (not the private key).
  - Anyone who sees the signed transaction can verify its authenticity by checking whether it was signed with the private key associated with the public key provided.
    - By doing so, other users can validate that the transaction was indeed signed by the rightful owner of the address (and the coins in that address).

Signatures on transactions

- Multiple Addresses, Multiple Signatures:
  - In transactions that spend Bitcoins from multiple addresses, there must be signatures corresponding to each of these addresses.
    - Each input requires a valid signature to prove ownership and authorization for spending the funds associated with that particular address.

UTXO (Unspent Transaction Output)

- UTXO Model for Validating Transactions:
  - In the UTXO model, every transaction input must be a transaction output of a previous transaction, linking the spending of funds to their source.
    - When a new transaction output is created, it is considered unspent until it gets consumed as an input in a future transaction when the funds are spent.
  - A valid transaction can only include unspent transaction outputs (UTXOs) as its inputs, providing proof that the address indeed has sufficient funds to spend.

UTXO (Unspent Transaction Output)

- Preventing Double-Spending:
  - By keeping track of UTXOs at all times, honest users can validate every new transaction against this set to prevent double-spending attempts by dishonest users.
  - To spend their money, users must provide a valid chain of ownership through the UTXO history, ensuring that each input is a previously unspent output.

UTXO (Unspent Transaction Output)

- Multiple Transaction Outputs (Outputs and Change):
  - A transaction may have multiple outputs, allowing users to send funds to multiple recipients in a single transaction.
  - For example, if a user owns an address with 2 Bitcoins in an unspent output and wants to spend 1 Bitcoin, the transaction will have two outputs: one for the recipient and one for themselves (the change).
    - The change output sends the remaining 1 Bitcoin back to the user's address or a new address.

UTXO (Unspent Transaction Output)

- Multiple Transaction Inputs:
  - Users can include multiple transaction inputs in a single transaction to combine funds from different unspent outputs for larger transactions.
  - For instance, if a user owns an address with 1 Bitcoin from two separate unspent outputs and wants to pay 2 Bitcoins to another address, they can include both unspent outputs as inputs in the transaction.

Transaction fees

- Total Value in Inputs and Outputs:
  - In a Bitcoin transaction, the total value in the inputs (UTXOs being spent) must be equal to or greater than the total value in the outputs (newly created UTXOs).
  - The difference between the total value in inputs and outputs is known as the transaction fees.

Transaction fees

- Transaction Fees as Miner Incentives:
  - The transaction fees are claimed by the miner who successfully includes the transaction in a block and adds that block to the blockchain.
  - Miners compete to add transactions to their blocks because the fees they collect are an additional reward on top of the block reward (newly minted Bitcoins).

Transaction fees

- Transaction Prioritization and Speed:
    - Transactions with higher fees are more attractive to miners because they earn more rewards for including them in their blocks.
        - Miners prioritize transactions with higher fees, leading to faster inclusion of these transactions in the blockchain.
        - Transactions with lower fees may take longer to be confirmed since they are lower on the priority list.

Transaction fees

- Automatic Fee Calculation:
  - Wallets automatically calculate transaction fees based on a particular fee rate, measured in Satoshi per kilobyte (Satoshi/kB).
    - The fee rate determines the amount of Satoshi to be paid per kilobyte of transaction data. Why?
    - Higher fee rates result in faster confirmation times, while lower rates may lead to delayed confirmations.

Transaction fees

- Dynamic Fee Variation:
  - Transaction fees can vary over time due to fluctuations in network demand and block space availability.
  - During periods of high network congestion, transaction fees may increase as users compete for limited block space.

Transaction mempool

- Transaction Propagation:
  - When a user wants to make a Bitcoin transaction, they create a transaction message, sign it using their private key, and broadcast it across the Bitcoin network.
  - The transaction message is a few kilobytes in size and contains information about the transaction inputs, outputs, and the corresponding digital signatures.
  - Miners in the network receive these transactions and verify the signatures before adding them to their memory pool, known as the "**mempool**."

Transaction mempool

- The Mempool:
  - The mempool is a temporary storage area where miners keep pending transactions that they have received and verified.
  - Transactions in the mempool are waiting to be included in a block and added to the blockchain.
  - Miners prioritize which transactions to include in their working block based on factors like the transaction fee rate (higher fee with smaller size).

Transaction mempool

- Block Size Limit:
  - Each block in the Bitcoin blockchain has a maximum size limit, currently set at 1 megabyte (MB) in the Bitcoin network.
  - Miners aim to maximize the total transaction rewards in their block while adhering to this size limit.

Transaction mempool

- Confirmation Latency:
  - There is a certain latency between the time a transaction is issued and when it is confirmed on the blockchain.
    - This latency is influenced by two factors:
      1. The time it takes for a transaction to be included in a block (which can be reduced by offering a higher transaction fee)
      2. The time it takes for that block to be buried deep enough in the longest chain for the transaction to be considered confirmed.

Transaction mempool

- Blockchain Design Trade-offs:
  - The trade-off between latency and security is specific to the Bitcoin blockchain design.

Resources

- ECE/COS 470, Pramod Viswanath, Princeton 2024
- CS251, Dan Boneh, Stanford 2023