BLOCKCHAIN: SMART CONTRACTS LECTURE 6 – **BITCOIN-PART1**

FLORIN CRACIUN

IMPORTANT

Some of the following slides are the property of

Dr. Emanuel Onica & Dr. Andrei Arusoaie

Faculty of Computer Science,

Alexandru Ioan Cuza University of Iași

and are used with their consent.

CONTENTS

1. Centralized money
2. Bitcoin - generalities
3. Bitcoin
    a. addresses & keys
    b. wallets
    c. transactions

What is money?

FROM GOLD TO CENTRALIZED MONEY

- Money are a form of value that people trust over time
  - Medium of exchange
  - Historically, people used: gold, salt, wheat, etc.
- "Paper" money: instead of carrying a bar of gold, people started to use paper money issued by a trusted authority
- Today fiat money are not even backed by physical value, e.g., gold; they only function as a legal tender enforced by a government

ISSUES WITH CENTRALIZED MONEY

- Corruption
- Mismanagement
- Lack of control

DIGITAL MONEY - A CENTRALIZED APPROACH

- Yet a centralized solution
- Online banking + specialized solutions (e.g., PayPal, Amazon Pay, Payline, Shopify Payments, TransferWise…)
- Banks keep a ledger on their servers
- People have accounts, and the ledger keeps the transactions corresponding to these accounts
- Issues:
  - we have to trust the bank, their engineers, their servers…
  - everything is kept secret
  - the **double spend problem**

WHAT IS BITCOIN?

- A decentralized solution
- A public ledger = transparency
- No central authority
- Accounts are anonymous
- Immune to censorship, confiscation
- Easy to ‘transport’ anywhere in the world

Whitepaper: https://bitcoin.org/bitcoin.pdf

BITCOIN - GENERALITIES

- The Bitcoin Protocol
- Based on blockchain
- Secured by mining: computing blocks (not easy)
- Mining incentive: current reward = 6.25 BTC/block
  - https://www.bitcoinblockhalf.com/
  - Last halving: May 11, 2020
- Block reward is halving every 210,000 blocks
  - It started with 50 BTC :-)
- Current price: https://www.coindesk.com/price/bitcoin

Bitcoin Units of Measure

1 Satoshi	= 0.00000001 ฿	
10 Satoshi	= 0.00000010 ฿	
100 Satoshi	= 0.00000100 ฿	= 1 Bit / µBTC (you-bit)
1,000 Satoshi	= 0.00001000 ฿	
10,000 Satoshi	= 0.00010000 ฿	
100,000 Satoshi	= 0.00100000 ฿	= 1 mBTC (em-bit)
1,000,000 Satoshi	= 0.01000000 ฿	= 1 cBTC (bitcent)
10,000,000 Satoshi	= 0.10000000 ฿	
100,000,000 Satoshi	= 1.00000000 ฿	

Source:
https://en.bitcoinwiki.org

BITCOIN - GENERALITIES

- It takes about 10 minutes to mine a block
  - Proof-of-work
  - High energy consumption
  - Mining rig:

MINING WORLD MAP

- Hashrate: unit of measurement for the processing power
  - calculations per second (orders: trillions per second)

- Mining map: https://cbeci.org/mining_map

KEY CONCEPTS

- Addresses & Keys
- Wallets
- Transactions

Bibliography: *Mastering Bitcoin*, Andreas M. Antonopoulos

Online: https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/

ADDRESSES & KEYS

HTTPS://WWW.OREILLY.COM/LIBRARY/VIEW/MASTERING-BITCOIN/9781491902639/CH04.HTML

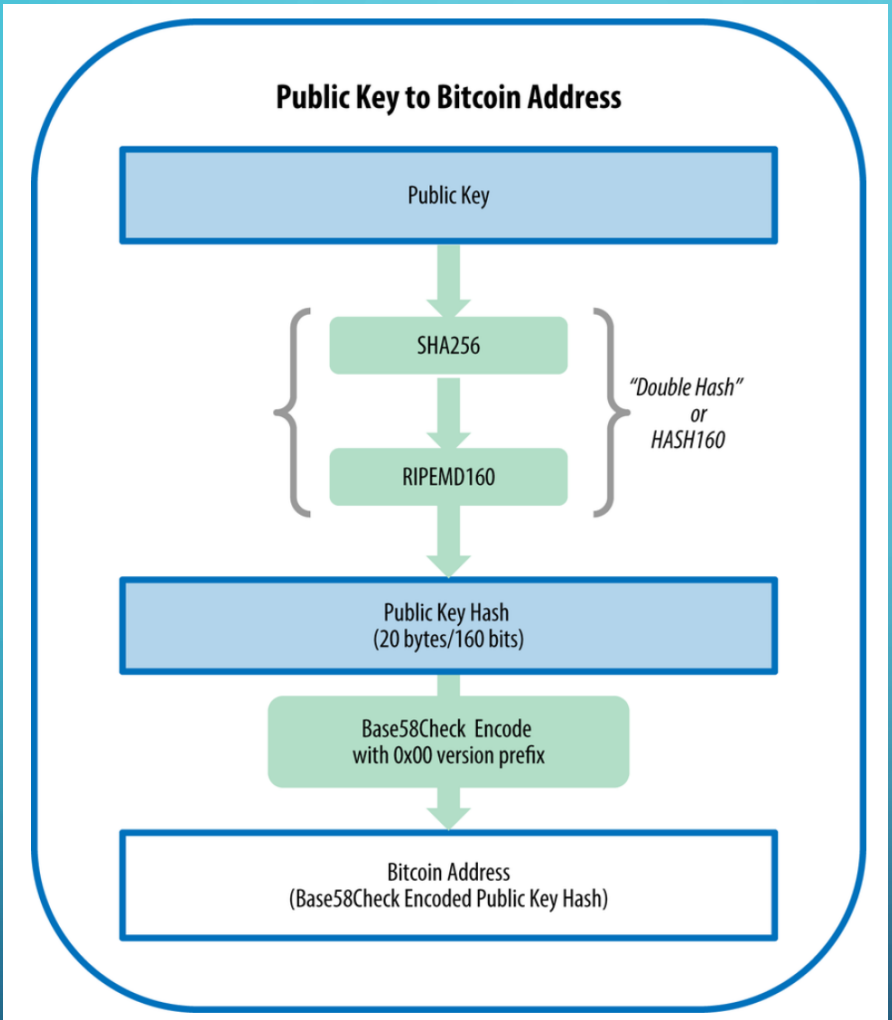
CREATING A BITCOIN ADDRESS

- **Private key (SK) generati**
  - 256 bit values => 2^{256}
  - How to generate: pick
    words) and then apply
- **Public key (PK) ge**
  - Uses Elliptic Curve Cryptography (E
  - PK = SK* G, where G is constant for secp256k1
  - Important: SK cannot be derived from the PK
- **Address:** compressed public key

More on Elliptic Curve Cryptography: https://unglueit-files.s3.amazonaws.com/ebf/05db7df4f31840f0a873d6ea14dcc28d.pdf#elliptic_curve

ADDRESS GENERATION

- There are multiple types of addresses:
  - P2SH: pay-to-script, addresses start with “3”
  - P2PKH: pay-to-public-key-hash, starts with “1”
  - Bech32: start with “bc1”, not recommended
- Bitcoin address: generated from the public key
  - Phase 1: hashing `ripemd160(sha256(publick key))`
  - Phase 2: a `base58check` encoding is performed
    - Allows the hash to be displayed in a more compact way
    - It avoids confusions and also performs checksums in order to make sure that the address is transmitted correctly
    - It uses a conversion table

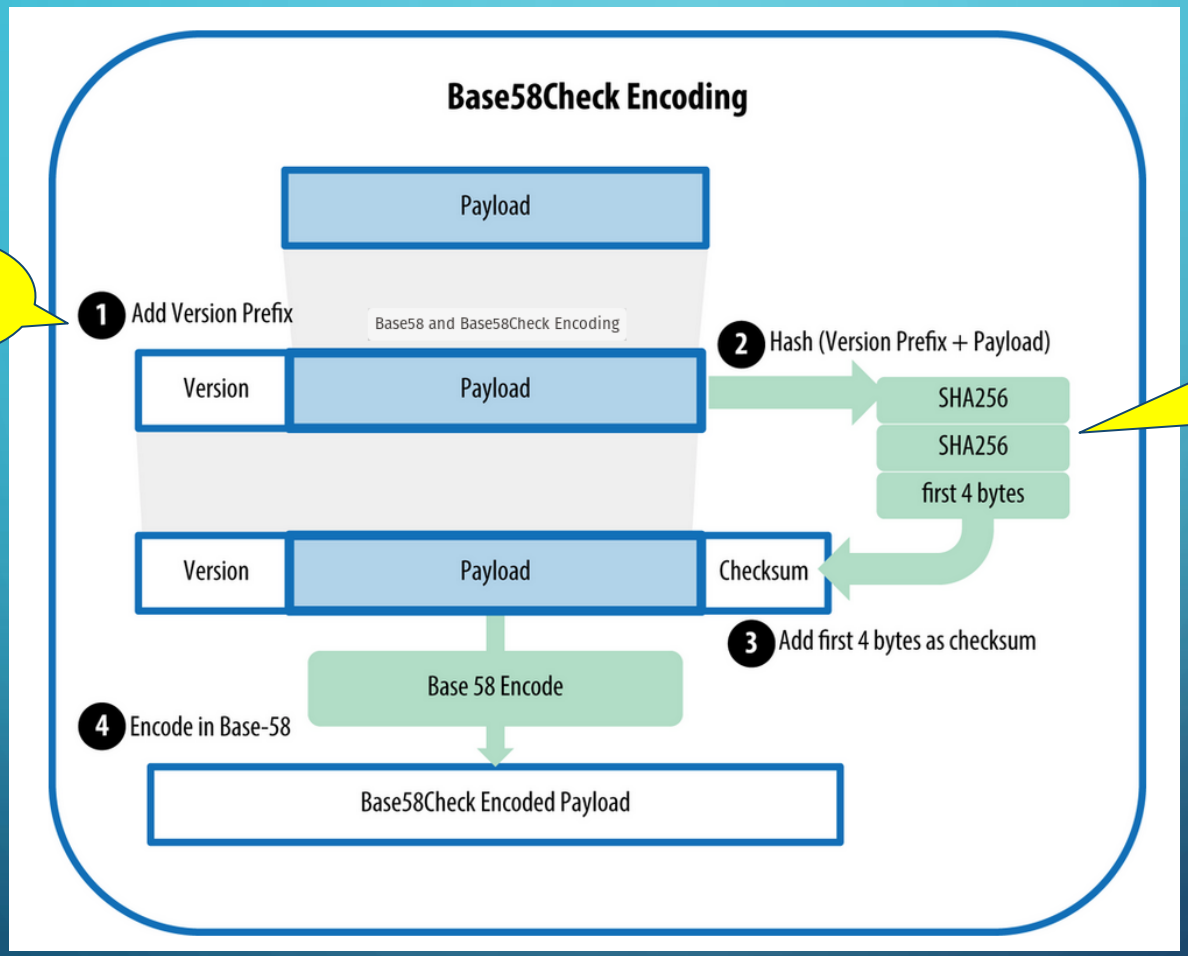


Source: Mastering Bitcoin, Andreas M. Antonopoulos

BASE58 ENCODING TABLE

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Source: Mastering Bitcoin, Andreas M. Antonopoulos

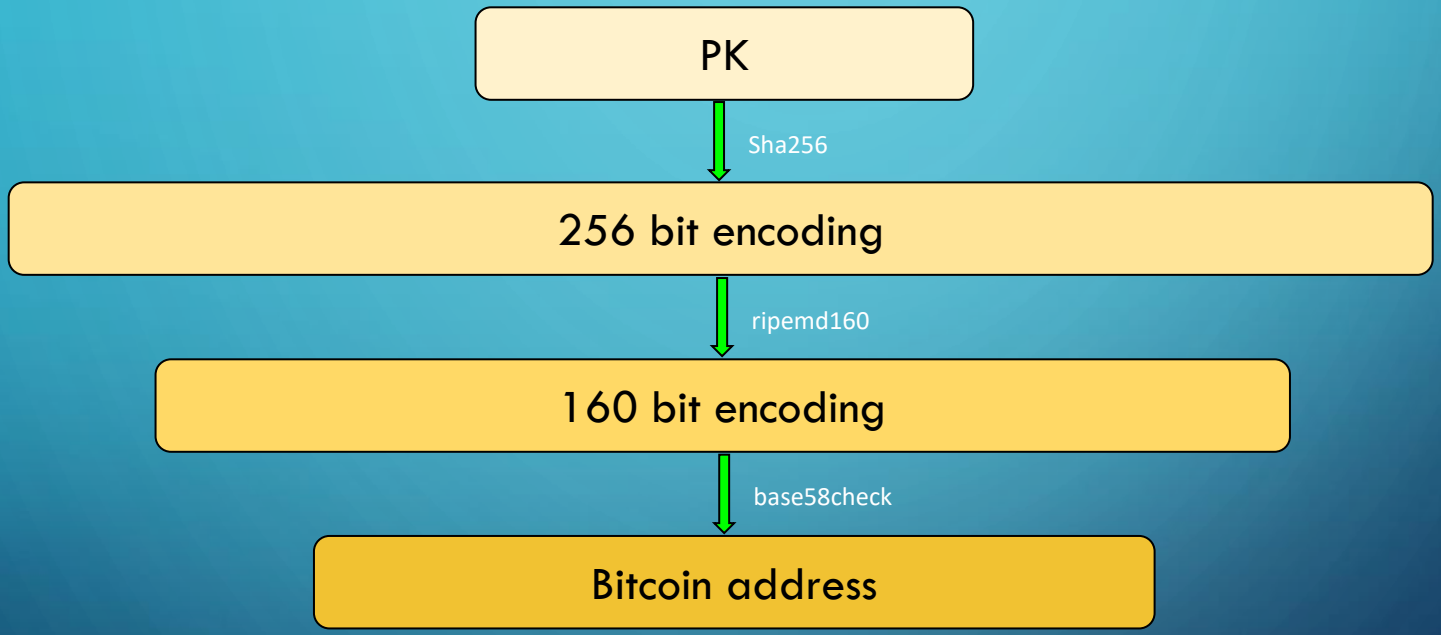


Source: Mastering Bitcoin, Andreas M. Antonopoulos

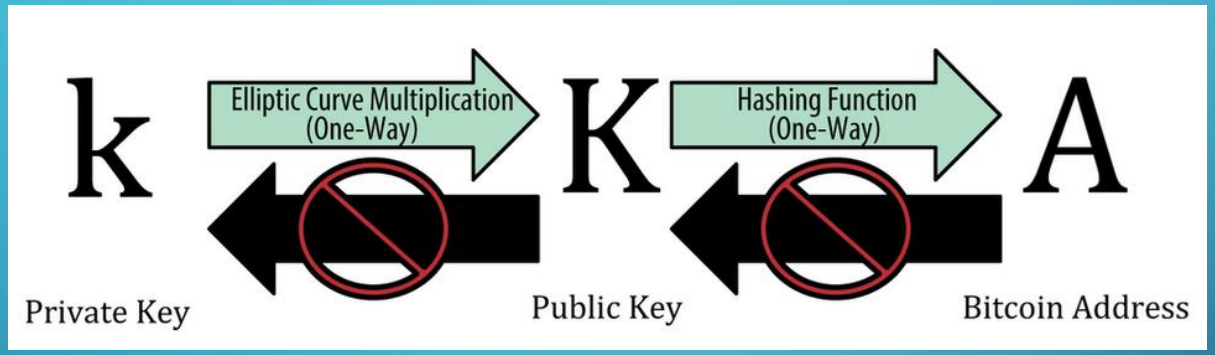
VERSION PREFIXES

Type	Version prefix (hex)	Base58 result prefix
Bitcoin Address	0x00	1
Pay-to-Script-Hash Address	0x05	3
Bitcoin Testnet Address	0x6F	m or n
Private Key WIF	0x80	5, K or L
BIP38 Encrypted Private Key	0x0142	6P
BIP32 Extended Public Key	0x0488B21E	xpub

ADDRESS GENERATION OVERVIEW



Source: Mastering Bitcoin, Andreas M. Antonopoulos



Source: Mastering Bitcoin, Andreas M. Antonopoulos

WALLETS

https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch04.html

WALLETS

- Wallet: enables the sending and receiving coins
  - It holds a combination of private/public key (SK/PK)
    - The public key is used to generate the address
    - The private key is required to transfer values to addresses
  - There are mechanisms that allow users to generate new SK from a master/seed key
  - New receiving addresses are created for privacy reasons
    - The coins are still managed by the same wallet!

METHODS OF KEY GENERATION FOR WALLETS

- Non-deterministic (random) wallets
  - a.k.a type-0 Non-deterministic wallets
  - Bitcoin Core Client pre-generates 100 random private keys
  - Obsolete; problems: address re-usage, hard to manage, backup, and import
- Deterministic (seeded) wallets
  - Keys are derived from a common seed using one-way hash functions
    - The seed is combined with random data or chain code => SK

MNEMONIC CODES

- A series of words that encode a *random number* = *seed*
- Mnemonic codes are sufficient to re-create the seed and recover the wallet
- Mnemonic codes are defined in BIP0039
  - Bitcoin Improvement Proposal 39
    - Create random sequence (128 -> 256 bits) -> seed
    - Create a checksum by taking the first bits of sha256(seed)
    - Add the checksum to the end of seed
    - Divide the sequence into sections of 11 bits and use them to index a dictionary of 2048 pre-defined words
    - Produce 12-24 words representing the mnemonic code
  - https://iancoleman.io/bip39/

HIERARCHICAL DETERMINISTIC WALLET

- BIP0032 (https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki)
- Keys are derived in a tree structure;
  - Root: is derived from a seed
  - Each parent can derive a sequence of children keys
- Advantages:
  - Express additional organization meaning in the tree
    - a brach for incoming payments, special departments, subsidiaries
  - Users can create a sequence of public keys without access to the corresponding private keys
    - Enables the use on insecure servers;
    - Public key for each transaction

WALLET TYPES

- Desktop wallets: installed on local computer
- Online wallets: run on the cloud
- Mobile
- Hardware: various devices that store the private key (e.g., USB sticks)
- Paper: a piece of software that is used to securely generate keys which are then printed and scanned

DESKTOP WALLETS

+ The environment enables complete control over funds
+ Some desktop wallets offer hardware wallet support or can operate as full nodes

- Difficult to utilize QR codes for creating transactions
- Susceptible to bitcoin stealing (malware, viruses)

ONLINE (WEB) WALLETS

+ Easy to access via web browser
+ Funds can potentially be recovered

- Service disruptions makes it difficult to access funds
- If the wallet platform is hacked… your funds are at risk

MOBILE WALLETS

+ Portable, convenient, ideal for face-to-face transactions
+ Designed to use QR codes to create quick transactions

- Dependency on the wallet app; updates or maintenance issues
- Loss or damage of the device may lead to loss of money

HARDWARE DEVICES

+ One of the most secure methods to store funds
+ Ideal for storing large amounts of bitcoin

- Difficult to use compared to mobile devices
- Not designed for scanning QR codes
- Loss of device: unrecoverable funds

PAPER WALLETS

+ Also secure to store funds
+ Tamper resistant (physically protected)
+ Perfect for giving :-)

- Needs QR scanning
- Loss of the paper = loss of funds

PROTECT YOUR WALLETS

- Offline wallets are more secure than online wallets
- Protection:
  - Backup
  - Keep up to date your wallet
  - Add security layers: complex password, protect any operation by asking for the password, etc.
- Wallets can store more than one cryptocurrency
- https://www.thebalance.com/best-bitcoin-wallets-4160642

TRANSACTIONS

https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch05.html

TRANSACTIONS

- Encode the transfer of value between participants
- They are registered on a public ledger
- Lifecycle:
  - A transaction is signed = authorize to spend the funds
  - Then it is broadcasted to the network
  - Finally, it is verified by a mining node and included in a block which is eventually added to the blockchain

CREATING TRANSACTIONS

- Transactions can be created online or offline
- For Bitcoin, they include various informations:
  - Version number, no of inputs, input transactions, no of outputs, output transactions, lock_time, scripts, etc.
- Transactions indicate a source of funds and a destination
  - References to previous (unspend) transactions
- If properly formed and signed -> miners execute the transfer of the funds

TRANSACTION STRUCTURE

Size	Field	Description
4 bytes	Version	Specifies which rules this transaction follows
1–9 bytes (VarInt)	Input Counter	How many inputs are included
Variable	Inputs	One or more transaction inputs
1–9 bytes (VarInt)	Output Counter	How many outputs are included
Variable	Outputs	One or more transaction outputs
4 bytes	Locktime	A Unix timestamp or block number

If 0 then execute immediately.
If <500 million then the field is interpreted as block height, and it will not be executed prior to that block.
Otherwise, it is interpreted as unix epoch timestamp and transaction is not executed before that time.

Source: Mastering Bitcoin, Andreas M. Antonopoulos

UTXO

- UTXO = unspent transaction output
- Indivisible chunks of bitcoin locked to a specific owner, recorded in the blockchain

- Scenario:
  - Alice has UTXO1 with 30 BTC and UTXO2 with 20BTC
  - Wants to send 40 BTC to address B
  - She creates 1 transaction with 2 outputs (locktime can be 0):

No of inputs	2	
Inputs		UTXO1, UTXO2
No of outputs	2	
Outputs		40 BTC -> B, 10 BTC -> Alice (address)

TRANSACTION OUTPUTS - DETAILS

Transaction outputs:

- An amount of BTC denominated in satoshis
  - 1 BTC = 100 million satoshi
- A locking script: *locks the amount by specifying the conditions to spent the output*

Size	Field	Description
8 bytes	Amount	Bitcoin value in satoshis (10^{-8} bitcoin)
1-9 bytes (VarInt)	Locking-Script Size	Locking-Script length in bytes, to follow
Variable	Locking-Script	A script defining the conditions needed to spend the output

Transaction Outputs

Source: Mastering Bitcoin, Andreas M. Antonopoulos

TRANSACTION INPUTS - DETAILS

Transaction inputs:

- References to UTXO
- Unlocking script: *that satisfies the spending conditions set by the UTXO*

Wallets handle transaction creation, inputs, outputs automatically.

Size	Field	Description
32 bytes	Transaction Inputs on Hash	Pointer to the transaction containing the UTXO to be spent
4 bytes	Output Index	The index number of the UTXO to be spent; first one is 0
1-9 bytes (VarInt)	Unlocking-Script Size	Unlocking-Script length in bytes, to follow
Variable	Unlocking-Script	A script that fulfills the conditions of the UTXO locking script.
4 bytes	Sequence Number	Currently disabled Tx-replacement feature, set to 0xFFFFFFFF

Source: Mastering Bitcoin, Andreas M. Antonopoulos

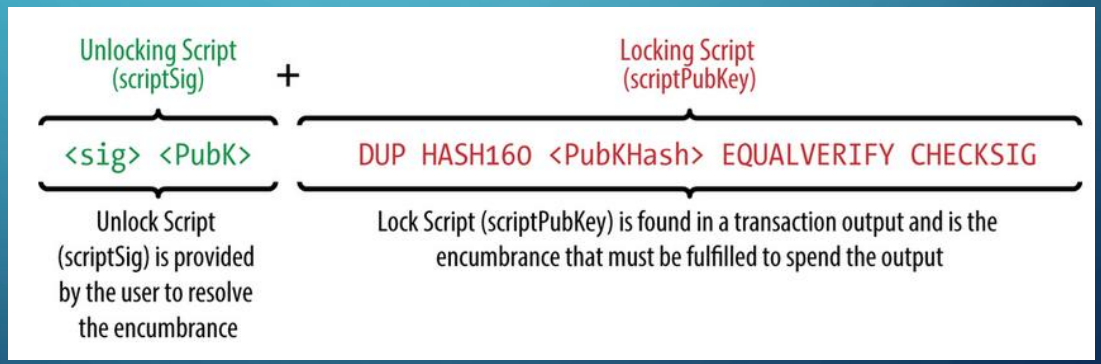
LOCKING AND UNLOCKING SCRIPTS

Bitcoin script:

- Stack-based language, “assembly”-like, Turing incomplete
- Programs = list of instructions which are processed from left to right
- OpCodes:
  - Constants (`OP_0`, `OP_TRUE`, `OP_PUSHDATA1`, `OP_NEGATE...`)
  - Flow control (`OP_IF`, `OP_VERIFY`, `OP_RETURN`, `...`)
  - Stack (`OP_DUP`, `OP_ROT`, `OP_SWAP`, `...`)
  - Splice (`OP_CAT`), Bitwise logic (`OP_INVERT`, `OP_EQUAL`, `...`)
  - Arithmetic (`OP_ADD`, `OP_BOOLAND`, `OP_LESSTHAN`, `...`)
  - Crypto (`OP_SHA256`, `OP_CHECKSIG`, `OP_CHECKSIGVERIFY`,`...`)
  - …

PAY-TO-PUBKEY-HASH: P2PKH

- Locking script, a.k.a. scriptPubKey:
  - `OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG`
- Unlocking script, a.k.a. scriptSig:
  - `<sig> <pubKey>`
- Combine:



PAY-TO-PUBKEY-HASH: LOCK + UNLOCK

Stack (stack top is on the left)	Locking Script + Unlocking Script	Obs.
	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig scriptPubKey
<pubKey> <sig>	OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Constants go in the stack
<pubKey> <pubKey> <sig>	OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Duplicated top
<pubHash> <pubKey> <sig>	<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Top hashed
<pubKeyHash> <pubHash> <pubKey> <sig>	OP_EQUALVERIFY OP_CHECKSIG	Constants go in the stack
<pubKey> <sig>	OP_CHECKSIG	Equality of the first 2 items is checked
true	Done	Signature is checked

INSPECT REAL TRANSACTIONS

Blockcypher: https://live.blockcypher.com/btc/

TRANSACTION CHAINING

- Transactions are linked together: UTXOs are inputs for other transactions
  - Therefore, transactions depend on each other
  - There is a child-parent relationship
- In the network, child may arrive before the parent
  - Child is kept in a temporary pool until parent arrives
  - *Orphan transaction pool*

TRANSACTION FEES

- Fees are incentives for miners
- They are calculated based on the size of the transaction in kilobytes
- Transaction fees affect the processing priority
- Typically, wallets compute fees automatically
- Fees = Sum(Inputs) - Sum(Outputs)
  - You have to include a transaction output for change
  - Don't forget to pay the fees

BIBLIOGRAPHY

Further reading about transactions:

- *Mastering Bitcoin*, Andreas M. Antonopoulos
  - Chapter 5: https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch05.html