INTRODUCTION to BLOCKCHAIN

CHAPTER3: CRYPTOCURRENCIES

Dr. Noureddine Lasla

Chapter Overview

Objective:

Understanding Digital Assets and Their Role in Blockchain

Key Areas of Focus:

- Introduction to Money
- History of Money
- Introduction to Cryptocurrencies
- Types of Cryptocurrencies
- Key Features of Cryptocurrencies
- Bitcoin
- Altcoin

What is Money?

Money is any item or medium of exchange that symbolize perceived **VALUE**.

Functions of Money:

- Medium of exchange: Money eliminates the inefficiencies of **barter** systems by providing a commonly accepted method for exchanging goods and services.
- Unit of account: Money provides a standard measure of value, allowing people to compare the worth of different goods and services. (**Abstract units of measure.)**
- Store of value: Unlike perishable goods, which may deteriorate over time, money retains its value.
- Standard of Deferred Payment: It provides a mechanism for credit and financial contracts, making it possible to lend, borrow, and repay in a universally accepted medium.

History of money

Barter

trade depends on a "**double coincidence of wants**,"

Commodity Money

Early forms of money included items with **intrinsic value** (salt, livestock, shell)

Metal Coins

400 BCE: standardized, and their value was backed by the weight and type of metal

Paper Money

700 AD: represent a **promise to exchange** for metal coins or goods. To avoid carrying large amount of coins

Banking & Bills of Exchange

Middle Age: Method of settling accounts in international trade

Gold Standard

!9th Century: Money was directly **linked** to a specific quantity of **gold**

Fiat Money

20th Century: not backed by a physical commodity. Value is derived from the **trust to Gov.**

History of money

Monetary System	Examples	Backing	Main Feature
Fiat Money	U.S. Dollar, Euro, Yen	No backing (trust-based)	Government-issued currency
Commodity Money	Gold Standard (historical)	Gold, Silver	Backed by physical commodities
Representative Money	Gold certificates (historical)	Physical commodity (e.g., gold)	Claims to a commodity
Digital/Cryptocurrencies	Bitcoin, Ethereum	Decentralized (peer-to-peer)	Digital and decentralized
CBDCs (Central Bank Digital Currencies)	Digital Yuan, Sand Dollar	Government-issued fiat currency	Digital fiat currency, central bank control

Again…, what is money?

Units of currency are merely **abstract unit** of measurements.

Money is used to measure debt: It is an IOU (I owe you)

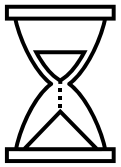
Technically any one can create money. But the hard part is to get that money **accepted**.

Introduction to Cryptocurrencies

What Are Cryptocurrencies?

- Digital or virtual currencies secured by cryptography.
- Operate on decentralized networks (blockchain).
- Enable peer-to-peer transactions without intermediaries (**electronic cash**).

History of Cryptocurrencies



- 2008: Bitcoin whitepaper published by Satoshi Nakamoto.
- 2009: Bitcoin network launched; first block mined (Genesis Block) *03-01-2009*.
- 2011: Emergence of altcoins (e.g., Litecoin).
- 2015: Ethereum introduced smart contracts.
- 2017: ICO (Initial Coin Offering) boom.
  - **Filecoin ICO (2017)** raised $205 million
- 2020s: Rise of DeFi, NFTs, and institutional adoption.

Types of Cryptocurrencies

- **Bitcoin** (BTC): First cryptocurrency, store of value, Limited supply (21 million coins).

- **Altcoins**: Ethereum (ETH), Litecoin (LTC), Ripple (XRP), etc.
  - Reduce volatility and act as a bridge between crypto and traditional finance.

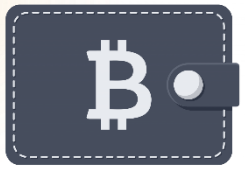
- **Stablecoins**: Pegged to fiat currencies (e.g., USDT, USDC).

- **Tokens**: Used within specific ecosystems (e.g., Binance Coin, linkchain, etc.).

Bitcoin

- Bitcoin transactions are transferred directly between users through a **peer-to-peer** network.
- Each transaction is recorded on the blockchain, ensuring that all Bitcoin exchanges are secure, transparent, and immutable.
- Bitcoin transactions rely on **asymmetric cryptography** for authentication and security.

Bitcoin Wallet



Digital tool that allows users to **store, send, and receive Bitcoin**.

- Private key: **proves ownership** of Bitcoins.
- Public key: **address** linked to the private key, used to receive Bitcoin.
- Types of Bitcoin Wallets:
  - Hot Wallets (online, mobile, desktop wallets): connected to the internet, for **quick** transactions.
  - Cold Wallets (offline, Hardware, paper wallets): providing enhanced security for **long-term** storage of Bitcoin.

Bitcoin UTXO

Bitcoin uses the **UTXO** (Unspent Transaction Output) model instead of a **balance-based** system.

In Bitcoin, every transaction consumes inputs (previously unspent outputs) and creates outputs (new UTXOs). These outputs can then be spent in future transactions.

Rather than maintaining a running balance for each address, Bitcoin tracks all individual UTXOs, each representing a specific amount of Bitcoin.

Bitcoin UTXO

UTXO (Unspent Transaction Output)

UTXO represents the amount of Bitcoin that has not been spent from a transaction.

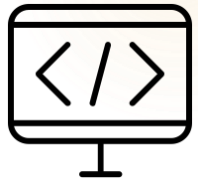
- Each Bitcoin transaction involves spending previously received UTXOs.
- Transaction **Inputs**: When you send Bitcoin, you are using one or more of your previous UTXOs as inputs.
- Transaction **Outputs**: The outputs are the new UTXOs created by the transaction. Any leftover Bitcoin after the transaction (i.e., **change**) is also returned as a new UTXO.

Bitcoin UTXO

How UTXOs Work:

- When you receive Bitcoin, you are essentially receiving UTXOs associated with a specific amount.
- When you make a transaction, you use these UTXOs as inputs. Bitcoin's system **doesn't allow** spending fractions of a UTXO, so if you're sending less than the full amount, the **remainder is sent back to you** as a new UTXO (often called "change").
- A transaction can have **multiple inputs** (coming from different UTXOs) and **multiple outputs** (sending Bitcoin to one or more recipients).

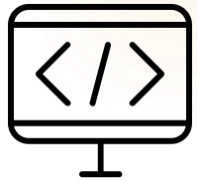
Bitcoin Scripting Language



What is Bitcoin script?

- Bitcoin Script is a **stack-based** programming language used to define the rules for how Bitcoin transactions are processed.
- It is **not** Turing-complete, meaning it's not designed for complex computations, but rather for simple transaction validation and authorization.
- Work with two types of instructions: **data instructions** and **OP_CODE**.

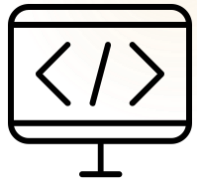
Bitcoin Scripting Language



Purpose of Bitcoin Script

- The goal is to define the conditions under which a Bitcoin transaction can be spent.
- It's a language for locking and unlocking Bitcoin during a transaction.
  - Lock script (**scriptPubKey**)
  - Unlock script (**scriptSig**).

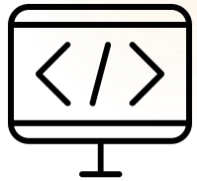
Bitcoin Scripting Language



Features of Bitcoin Script

- Stack-Based: uses a stack data structure where operations are performed by **pushing** and **popping** data to and from the **stack**.
- Non-Turing Complete: is **limited** in functionality to **prevent** complex operations that could cause security issues or make the system inefficient.
- Simple Operations: supports **basic** operations like addition, subtraction, comparison, and logical AND/OR operations.

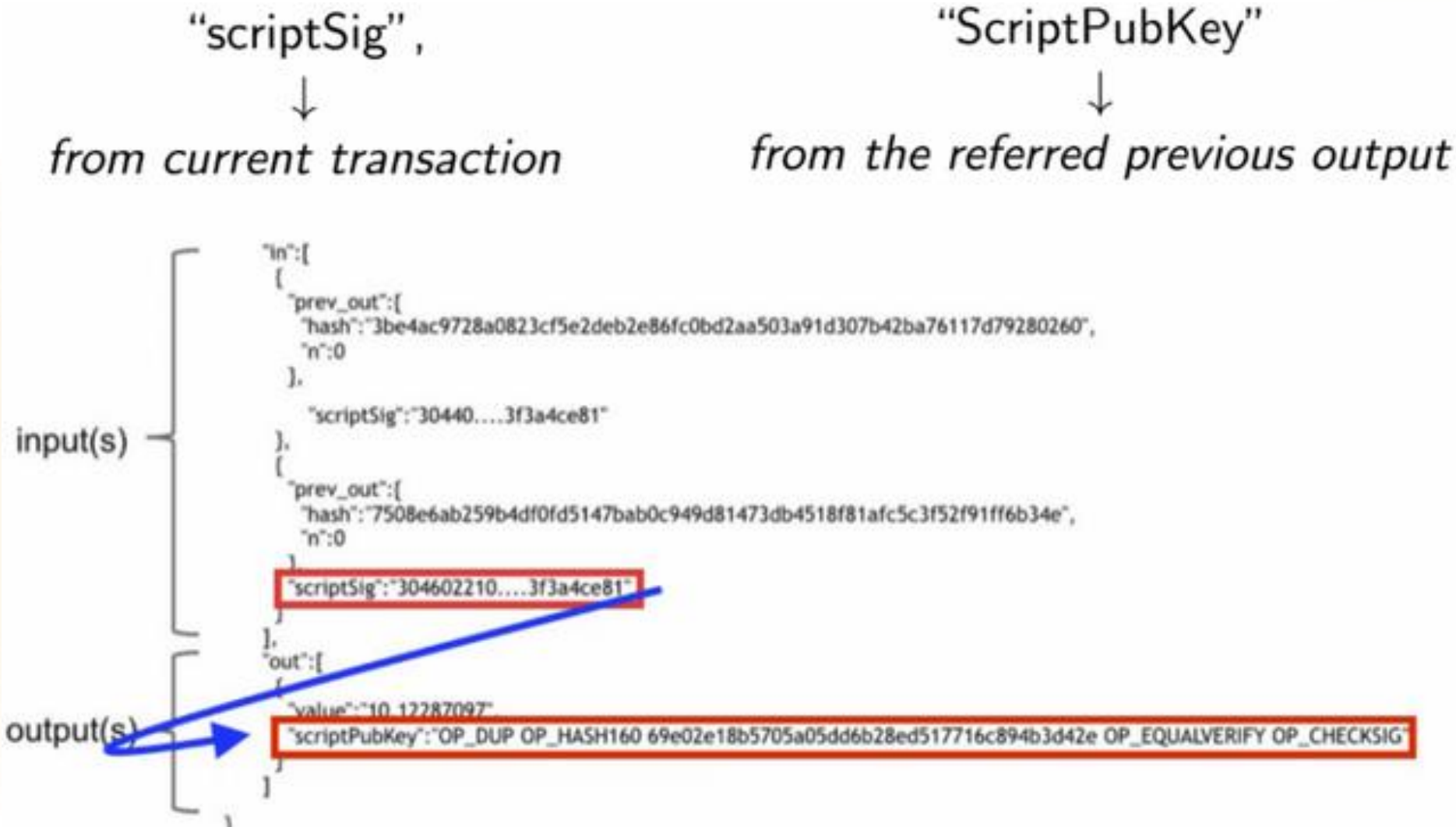
Bitcoin Scripting Language



How Bitcoin Script Work

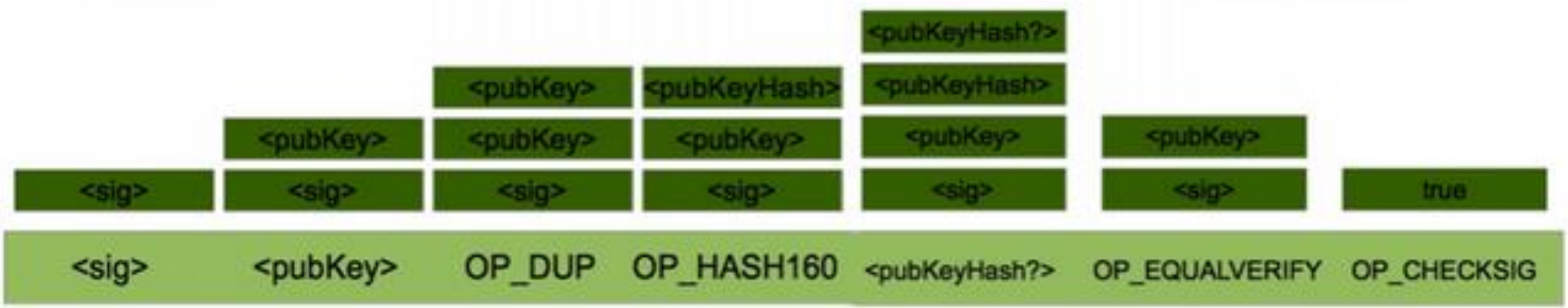
- Locking Bitcoin:
  - When a Bitcoin is sent, the sender specifies conditions under which it can be spent by creating a **scriptPubKey** (a locking script).
- Unlocking Bitcoin:
  - When the recipient wants to spend the Bitcoin, they must provide an unlocking script (**scriptSig**) to satisfy the conditions in the **scriptPubKey**.

Bitcoin Scripting Language

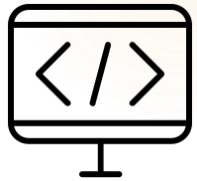


Bitcoin Script Language

Bitcoin Script Language



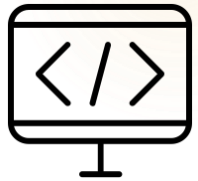
Bitcoin Script Language



Summary of the different types of Bitcoin scripts

Type of Script	Description	Example
Pay-to-PubKey-Hash (P2PKH)	The most common script type. The funds are locked to a public key hash (Bitcoin address). Only the owner of the private key corresponding to the public key hash can spend the funds.	**scriptPubKey**: OP_DUP OP_HASH160 <public_key_hash> OP_EQUALVERIFY OP_CHECKSIG
Pay-to-Script-Hash (P2SH)	The payer sends funds to a script hash, which can then be spent using any valid script matching the hash. Commonly used for multi-signature addresses and more complex scripts.	**scriptPubKey**: OP_HASH160 <script_hash> OP_EQUAL
Multi-Signature (Multisig)	A script type where multiple signatures are required to unlock the funds. Allows the creation of joint wallets, where several participants must sign to spend funds.	**scriptPubKey**: OP_2 <pubKey1> <pubKey2> <pubKey3> OP_3 OP_CHECKMULTISIG
Pay-to-Witness-PubKey-Hash (P2WPKH)	A SegWit-based script where the funds are locked to a public key hash and spent using a **witness**. It reduces transaction size and improves efficiency.	**scriptPubKey**: 0 <public_key_hash>
Pay-to-Witness-Script-Hash (P2WSH)	A SegWit-based script that locks funds to a script hash. Similar to **P2SH**, but designed to take advantage of SegWit.	**scriptPubKey**: 0 <script_hash>

Bitcoin Script Language



Summary of Bitcoin Script opcodes

Opcode	Description
OP_DUP	Duplicates the top item on the stack.
OP_HASH160	Applies **SHA-256** followed by **RIPEMD-160** hashing to the top item.
OP_CHECKSIG	Verifies that the digital signature is valid for the provided public key and data.
OP_EQUAL	Checks if the top two items on the stack are equal.
OP_EQUALVERIFY	Like **OP_EQUAL**, but also removes the top two items if they are equal.
OP_CHECKMULTISIG	Verifies multiple signatures for a **multi-signature** transaction.
OP_RETURN	Marks a transaction output as **unspendable**. Used for embedding data in the blockchain.
OP_IF / OP_ELSE	Conditional logic: executes script based on the truth value of the preceding condition.
OP_VERIFY	Verifies that the top stack item is **true** (non-zero). If not, the script fails.
OP_NOP	Does nothing, often used for backward compatibility or future extensions.

Bitcoin TX Example

How to transfer some BTC between Alice, Bob and Joe?

TxID:42b6a77cf6096f52fc7513cfdf861bcc1e841d7b05eb48d095a764df79501c7d

INPUTS From			OUTPUTS To	
(from previous transaction Alice has received):				
Alice	0.1005 BTC	⇒	Output #0 Bob's address	0.1000 BTC (spent)
			Transaction fees	0.0005 BTC

1. Bob's address (receiver): hash(Bob's public key)
2. Proof of ownership: transaction signed by Alice's private key (unlock)
3. Propagate the transaction on the Bitcoin network
4. Validate the transaction (rules):
   - check syntax, size
   - check double spending
5. Mining: the transaction becomes part of the blockchain

Bitcoin TX Example

Transfer from Bob to Joe

TxID: 30076701180f4dd48a7f1bb37027c6e791e950eaa44b6449f6b5d660b523f967

INPUTS From			OUTPUTS To	
42b6a77cf6096f52fc7513cfdf861bcc1e841d7b05eb48d095a764df79501c7d:				
Bob	0.1000 BTC	⇒	Output #0 Joe's address	0.0150 BTC (spent)
			Output#1 Bob's address (change)	0.0845 BTC (unspent)
			Transaction fees	0.0005 BTC

Bitcoin uses a scripting language (Turing incomplete)

The script for a Bitcoin transfer to dest. Bitcoin address D simply encumbers future spending of the bitcoins with two things:

- the spender must provide a public key that, when hashed, yields destination address D embedded in the script, and
- a signature to prove ownership of the private key corresponding to the public key just provided.

Altcoins

Altcoins are any cryptocurrencies that are not Bitcoin. The term "Altcoin" is short for "**alt**ernative **coin**," meaning they are alternatives to Bitcoin.

- Altcoins offer a wide range of applications and use cases.
  - Smart contracts (Ethereum), DeFi (Uniswap, Aave)
  - Privacy-focused transactions (e.g., Monero, Zcash)
  - Stable value (e.g., Tether, USDC)
- Many altcoins are at the forefront of blockchain innovation, exploring new consensus algorithms, scalability solutions, and other features.

Altcoins

Popular Altcoins

Altcoin	Use Case	Blockchain
Ethereum (ETH)	Smart contracts, dApps	Ethereum
Binance Coin (BNB)	Exchange token, DeFi	Binance Chain
Cardano (ADA)	Scalable smart contracts	Cardano
Solana (SOL)	High-speed transactions	Solana
Polkadot (DOT)	Blockchain interoperability	Polkadot
Dogecoin (DOGE)	Meme coin, payment method	Dogecoin