

The background is a solid blue color. In the top-left corner, there are several overlapping rectangular bars in shades of orange, dark blue, light blue, and white, all tilted at an angle. A similar set of overlapping bars in the same color palette is located in the bottom-right corner.

Blockchain Use Cases

Key Concepts

Keys



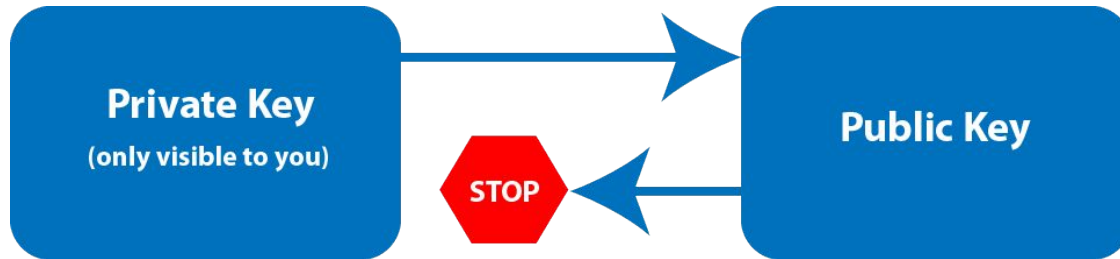
Private Keys

- Private Key is used to generate a signature for each transaction over the blockchain.
- The generated signature is used to confirm that the transaction has come from a specific user, and also prevents the transaction from being altered by any malign entity.
- In simple words - “Private Keys are used to sign the cryptocurrencies you send to others.”
- If someone obtains your private key, they would be able to send your cryptocurrencies to themselves, which has happened in most of the hacks around the world.
- Example: **L34EXrFCuxQCorfE66sxQe8Tyh71SyU8cc9z7HnbEWwW8YsgbvTw**

Public Keys

- The Private Key is used to derive the Public Key mathematically.
- Public Keys are practically irreversible, i.e., you can easily derive public key from the private key, but it would take millions of years to do the vice versa.
- Public Keys can be distributed to everyone.
- Example:

0237F49F4CCF760BF5FA993616E63B7B2A8611AB71AE7630386738B3BC4D1B84FD



Addresses

- A cryptocurrency address in a core is a representation of the public key.
- One-way cryptographic hash functions are used to derive address from the public key.
- For example in Bitcoin, the algorithms that are being used to generate a bitcoin address from the public key are the Secure Hash Algorithm 256 (SHA-256) and the RACE Integrity Primitives Evaluation Message Digest 160 (RIPEMD-160)
- The address appears typically in a transaction between two parties, with the address signifying the recipient of the funds.
- Example: **1JPgMJuAvYJU6mxxbJdmf1XBd7bBPdPV3a**

Private Key



Large, randomly
generated number

Public Key



Generated from
Private key

Address



Generated from
Public key

Transactions

- Transactions are records of data in chronological order
- Transactions are stored in a Merkle tree inside the Block.
- The transactions, when submitted, are picked up by the blockchain network and is inserted into a 'pool of unconfirmed transactions.' The transaction pool is a collection of all the transactions on that network that have not been confirmed yet.
- Miners on the network select transactions from this pool and add them to their 'block.'
- Transactions also contain metadata information which can be utilized to store data over the Blockchain.

What are Blocks?

- A Block is a container data structure which contains a set of confirmed transactions.
- A block could contain different information, and a chain of these blocks evolves into a blockchain as long as it links one and the other.
- The blocks are stored on the hard drives of many miners spread across the globe on a peer to peer network.
- In the Bitcoin algorithm, a block is created every 10 minutes. All the transactions happening over the network within 10 minutes interval are crunched into that block and added to the chain.

Structure of Blocks

All blocks in the Blockchain are composed of a header, identifiers and a long list of transactions. The structure of a block is as follows:

- Block Header
- Block identifiers
- Merkle Trees

Structure of Blocks

An Example of Bitcoin Blockchain

Field	Description	Size
Magic No	value always 0xD9B4BEF9	4 bytes
Blocksize	number of bytes following up to end of block	4 bytes
Blockheader	consists of 6 items	80 bytes
Transaction Counter	positive integer VI = VarInt	1 - 9 bytes
Transactions	The (non empty) list of transactions	Transaction counter-many transactions

Block Header

The header contains metadata about a block. There are three different sets of metadata:

- The previous block hash. In a blockchain, every block is inherited from the last block because we use the previous block's hash to create the new block's hash.
- Mining competition for the network. For every block to be part of the blockchain, it needs to be given a valid hash. This contains the values for the timestamp, the nonce, and the difficulty.
- Merkle tree root. This is a data structure to summarise the transactions inside the block.

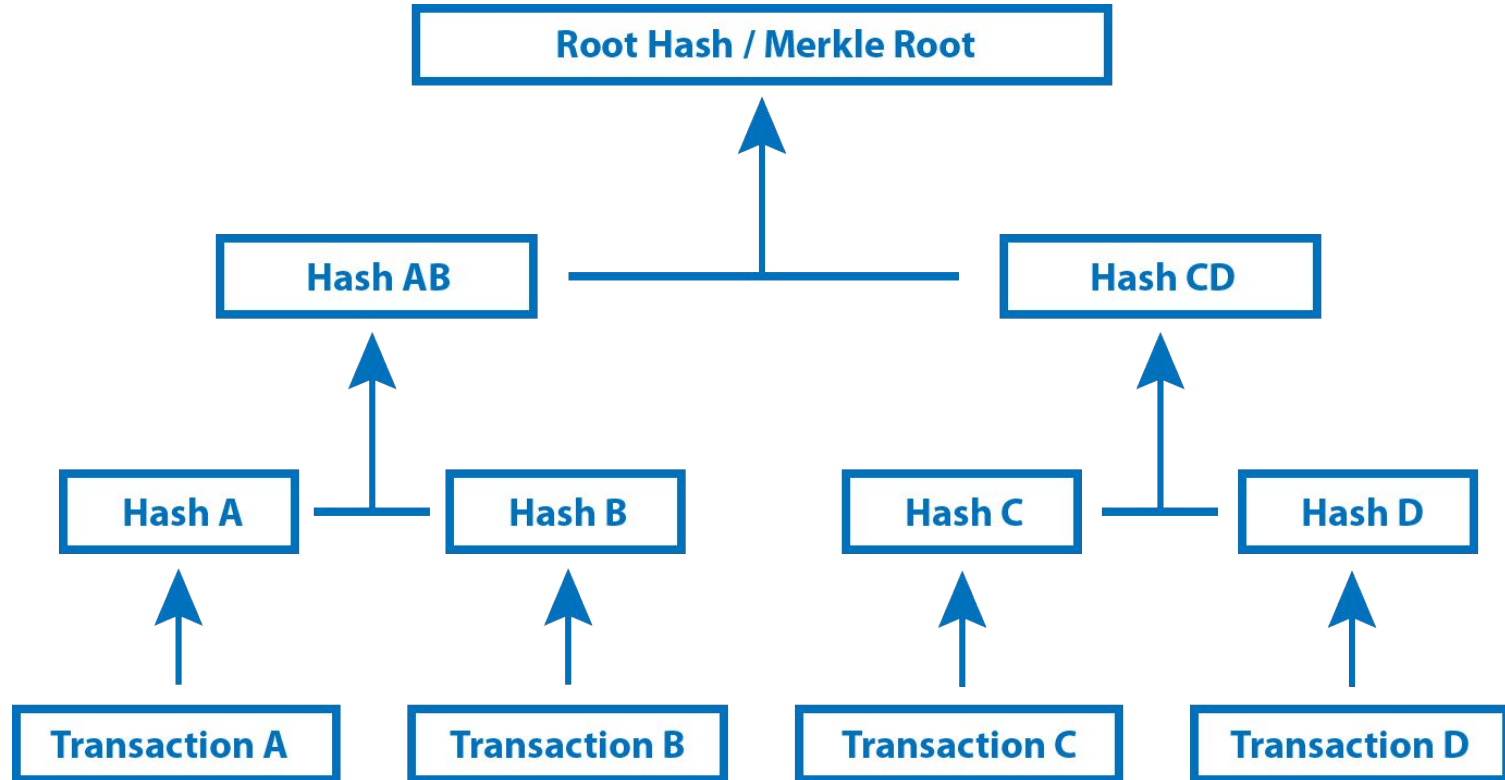
Block Identifier

- To identify a block, we need to have a cryptographic hash, a digital signature. This is created by hashing the block header twice with the SHA256 algorithm in case of Bitcoin Blockchain. You can use different hash functions for your Blockchain.
- Every block uses the last block's hash to construct its hash.
- Another way to identify a specific block is the block height. This is the position of the block in the blockchain.
- For example, if we say the block is in the 7312 position. This means that there are 7311 blocks before this one.

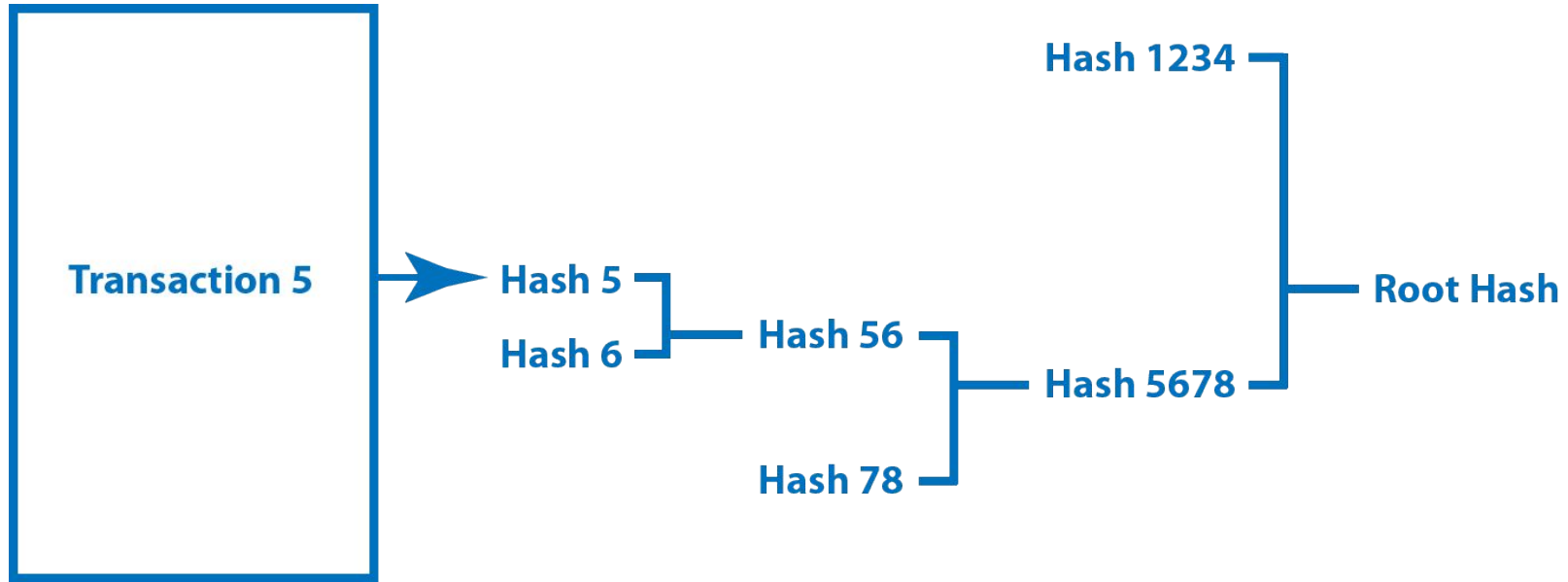
Merkle Tree

- A Merkle tree summarizes all the transactions in a block by producing a digital fingerprint of the entire set of transactions.
- The user can verify whether or not a transaction is included in a block.
- Merkle trees are created by repeatedly hashing pairs of nodes until there is only one hash left which is called the root hash.
- Each leaf node is a hash of transactional data, and each non-leaf node is a hash of its previous hashes.
- Merkle trees are binary and therefore require an even number of leaf nodes.
- If a single detail in any of the transactions or the order of the transaction's changes, so does the Merkle Root.

Merkle Tree



Merkle Tree



ADDITIONAL CONCEPTS



HD Private Key

- Hierarchical deterministic is a type of deterministic cryptocurrency wallet derived from a known seed, which allows for the generation of child keys from the parent key.
- The child key is generated from a known seed. There is a relationship between the child and parent keys that is invisible to anyone without that seed.
- The BIP 32 protocol can generate a nearly infinite number of child keys from a deterministically-generated seed from its parent.
- You can recreate those same child keys as long as you have the seed.
- The child key can operate independently, and the parent key can monitor and control each child key.

Mnemonics Seed

- A mnemonic seed is used to substitute either a 12, 18 or 24-word phrase for the private keys which can easily be memorized by human mind compared to hex encoded format.
- Mnemonic word phases are tied with the private keys and support wallet restoration.
- This provides additional security for the user, as well as a convenient solution to recover a wallet.
- BIP 39 introduced the mnemonic wallet implementation.
- The English wordlist for BIP 39 contains 2048 words, so to crack a 12-word phrase, it would require figuring out $2048^{12} = 2^{132}$ possible combinations under a shield of 128-bit security.



Smart Contracts

- Smart Contracts are the digital contracts signed between two parties and stored over the immutable ledger.
- Smart contracts help you exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman.
- Contracts can be encoded on any blockchain, but Ethereum is mostly used since it gives unlimited processing capability.
- Hyperledger is also providing chain codes which are very similar to Smart Contracts.
- Example: Renting an apartment.



THANK YOU

For more information contact
info@we2blocks.com

