# Cryptography

## Encrypt and decrypt

Alice wants to send a message to Bob, and for Bob's eyes only:

• Bob gives Alice his public key

• Alice uses Bob's public key to encrypt the message

• Alice sends Bob the encrypted message

• Bob decrypts the message with his private key

## Sign and verify

Alice wants to make sure that Bob's public announcement is indeed from Bob:

• Bob gives Alice his public key

• Bob signs his announcement with his private key

• Bob sends Alice his announcement and its signature

• Alice verifies the signature with Bob's public key

Ethereum uses Elliptic curve as its basis to sign and verify - <http://mathworld.wolfram.com/EllipticCurve.html>

<https://anders.com/blockchain/public-private-keys/keys.html>

## Mix and match

It is possible to mix both ideas, whereby Alice encrypts her message with Bob's public key, then signs the encrypt file with her private key. Upon reception, Bob verifies the signature with Alice's public key, then decrypts the file with his private key.

## Key management, PKI

If you look again at the Alice and Bob examples, you will notice that there is a vulnerability in "Bob gives Alice his public key". A malicious Charlie could intercept Bob's public key and pass on his own public key to Alice. Because of this, key management and public key infrastructure (PKI) is an important part of cryptography.

# Cryptographic hash functions

## Such a hash function:

• converts an input, a.k.a. the message, into an output, a.k.a the hash

• does the conversion in a reasonable amount of time

• is such that it is practically impossible to re-generate the message out of the hash

• is such that the tiniest change in the message, changes the hash beyond recognition

• is such that it is practically impossible to find 2 different messages with the same hash

## With such a function, you can:

• prove that you have a message without disclosing the content of the message, for instance:

o to prove you know your password

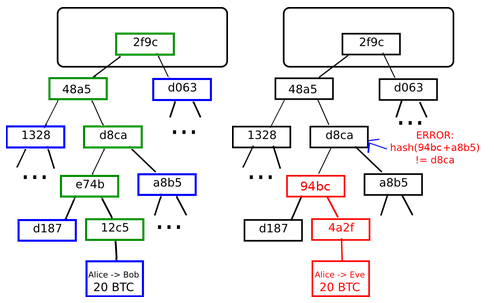
o to prove you previously wrote a message

• rest assured the message was not altered

• index your messages

Bitcoin uses SHA-256. Ethereum uses Keccak-256and Keccak-512.

# Merkle Tree



A Merkle tree is a tree where:

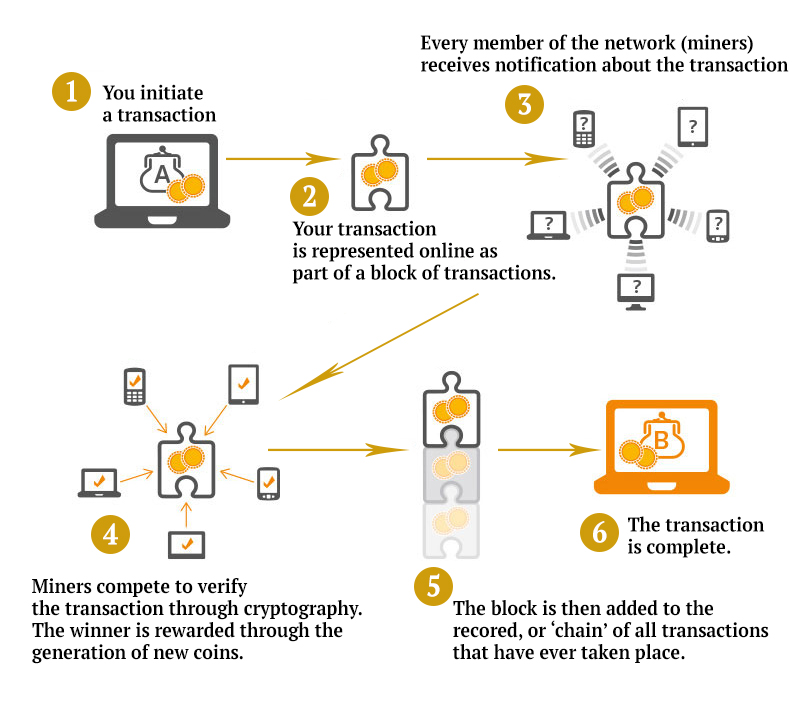
* each leaf node keeps a piece of the data.
* the complete data is obtained by collating the leaf nodes in a depth-first search fashion.
* each parent node keeps the computed hash of its children hashes.

With a Merkle tree, you can:

* keep data behind a single hash, just like a single-pass hash function would offer over the whole content.
* download and verify the integrity of data pieces as they come in random order.

In a blockchain environment, each block header keeps one or more hash(es) of the top leaf of one of more merkle tree(s).

# Mining



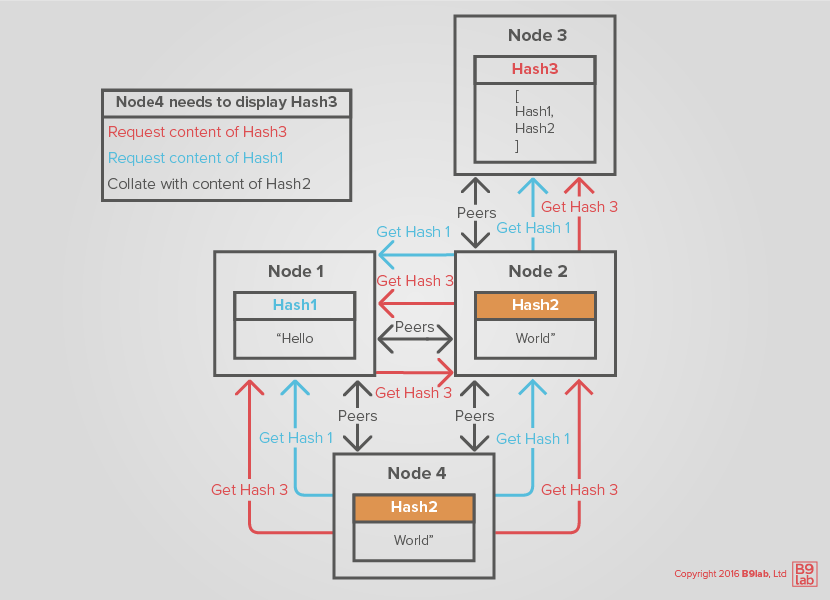
<https://anders.com/blockchain/hash.html>

<https://www.coinwarz.com/difficulty-charts/ethereum-difficulty-chart>

# Gossip protocol network

In the same fashion as in human social networks, a computer gossip network works by having each individual node relay freshly received information to its own peers. Such protocols vary in their aspects, such as latency or laziness. Nodes may also choose to drop new information.

In a Bitcoin context, a new transaction is broadcast in this way if and only if it is valid.



In a gossip network a node randomly chooses to whom to broadcast and it does not wait for acknowledgements

### Gas

Ethereum transactions are code and may include an infinite loop or a circular reference that never breaks, all of which can be seen as another type of attack. To protect against such an attack, Ethereum introduces another concept, that of gas. Gas is convertible from Ether, the currency of the Ethereum blockchain, is provided by the transaction creator and is consumed at each computational step:

* If the computation stops before all the gas is consumed, the consumed gas is collected by the winning miner, and the remainder returned to the transaction creator.
* If the gas provided is fully consumed before the computations have completed, the computation stops, all state changes dropped, and all gas goes to the winning miner.