Blockchains e criptomoedas

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

O que uma moeda digital descentralizada precisa?

http://www.weidai.com/bmoney.txt

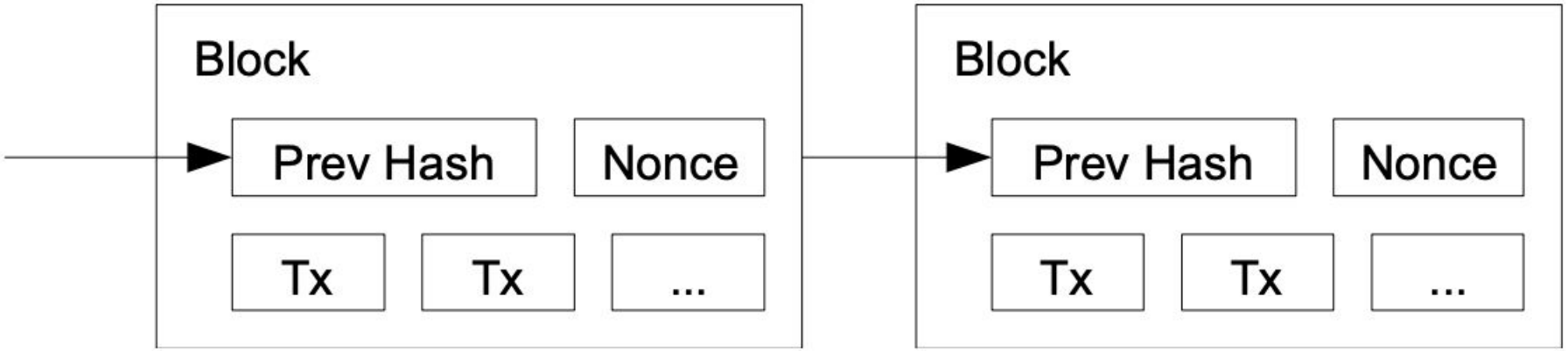
1. Uma forma de criar dinheiro.
2. Uma forma de transferir dinheiro.
3. Uma forma de armazenar todas as transações que seja pública e segura.

O que uma moeda digital descentralizada precisa?

http://www.weidai.com/bmoney.txt

1. Uma forma de criar dinheiro.
2. Uma forma de transferir dinheiro.
3. **Uma forma de armazenar todas as transações que seja pública e segura.**

Armazenamento de transações em cadeia



Usamos a hash para conferir se ocorreu modificação!
Replicamos essa corrente em vários nós.

Como inserimos um novo bloco? Proof of work

Como inserimos um novo bloco? dificuldade = 4

Simulação

https://andersbrownworth.com/blockchain/block

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Simulação

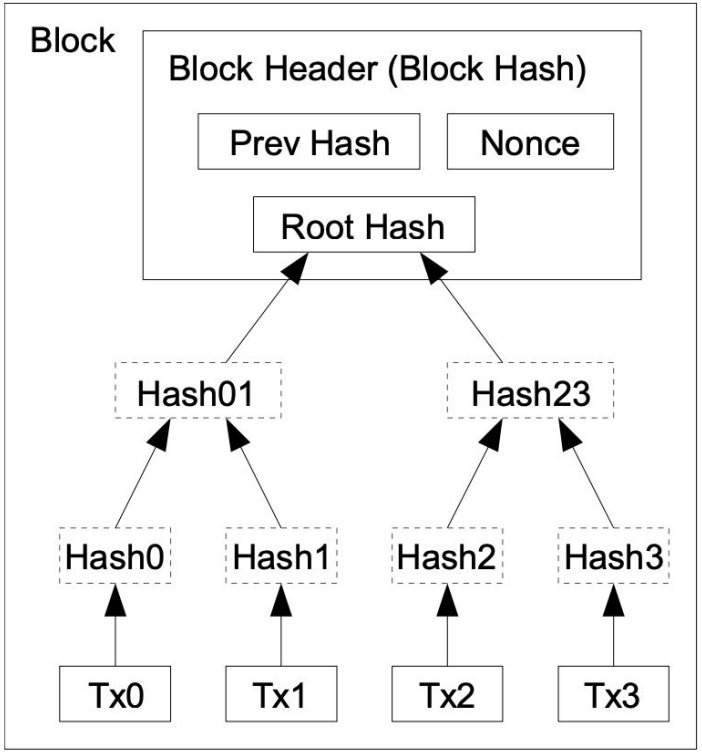
https://andersbrownworth.com/blockchain/block

The steps to run the network are as follows:

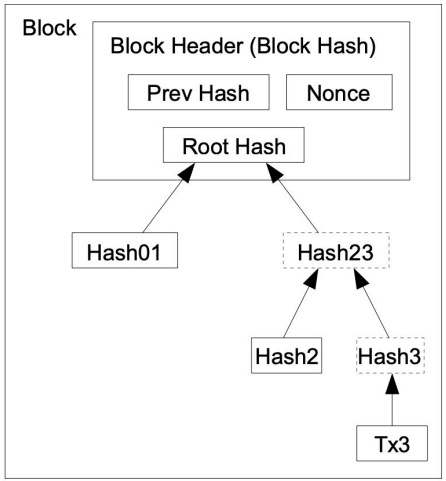
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

A máquina que encontrou o nonce é recompensado com uma taxa sobre as transações: dinheiro sendo minerado

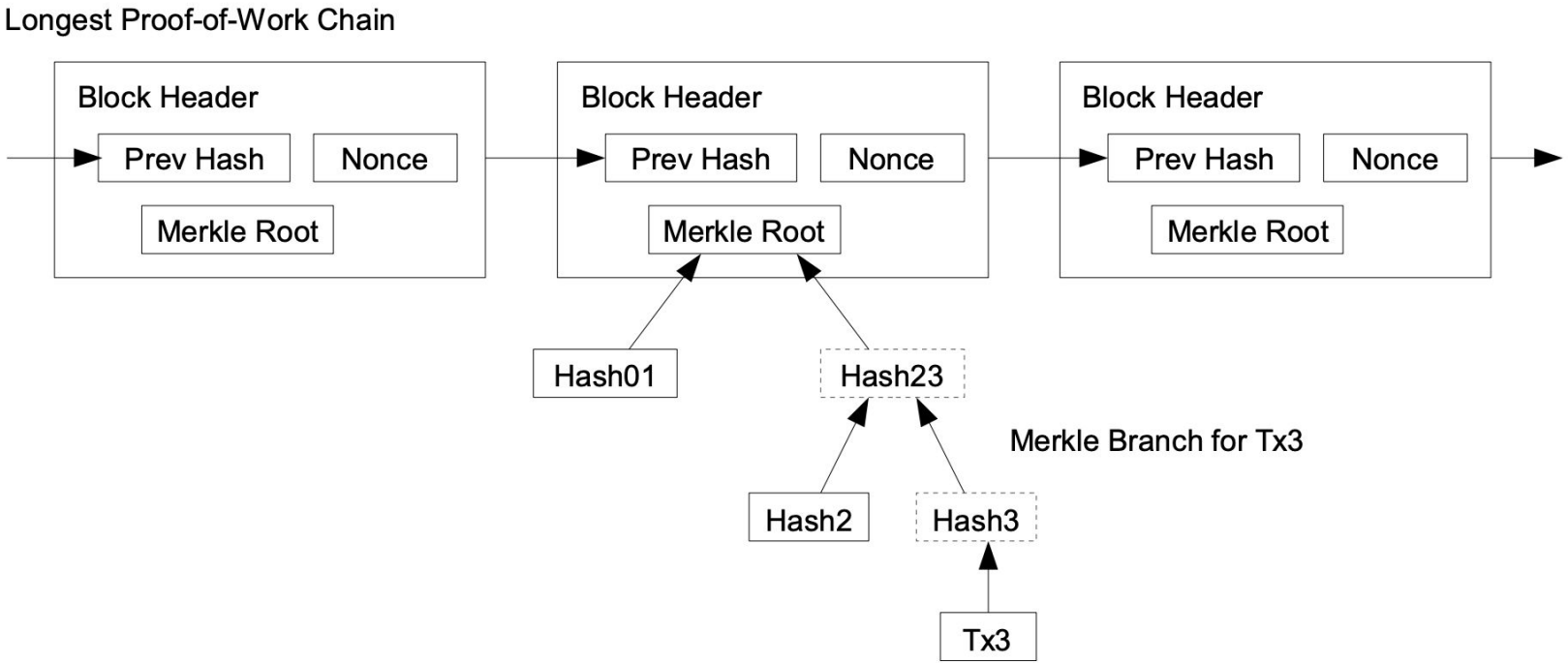
Reduzindo o espaço gasto: Árvore de Merkle



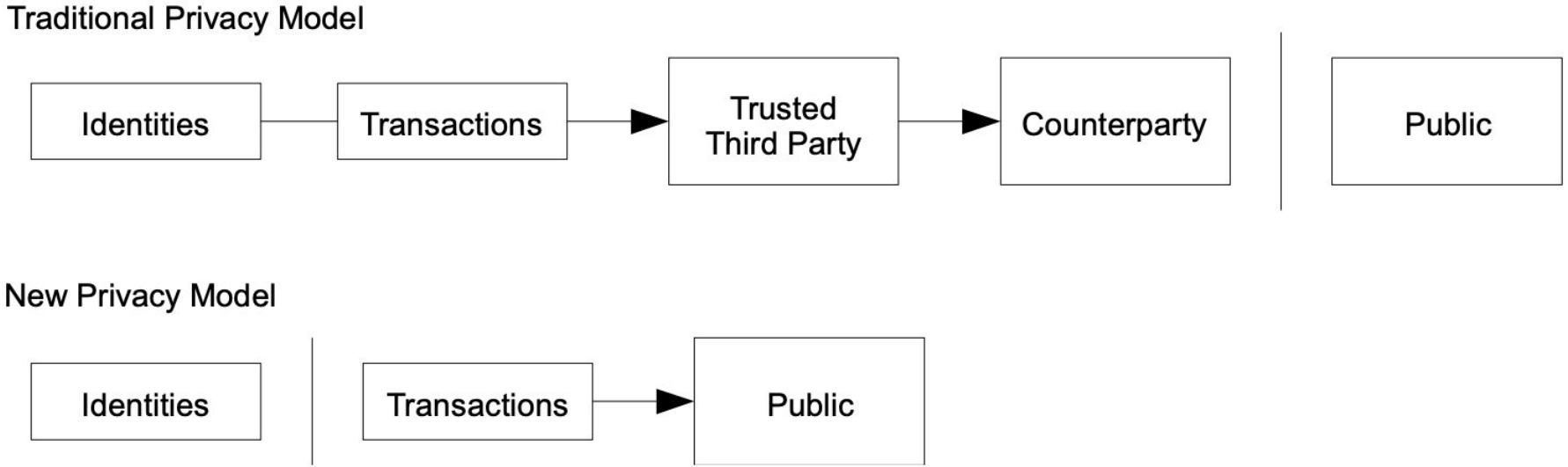
Conforme o tempo passa, blocos antigos podem ser compactados:



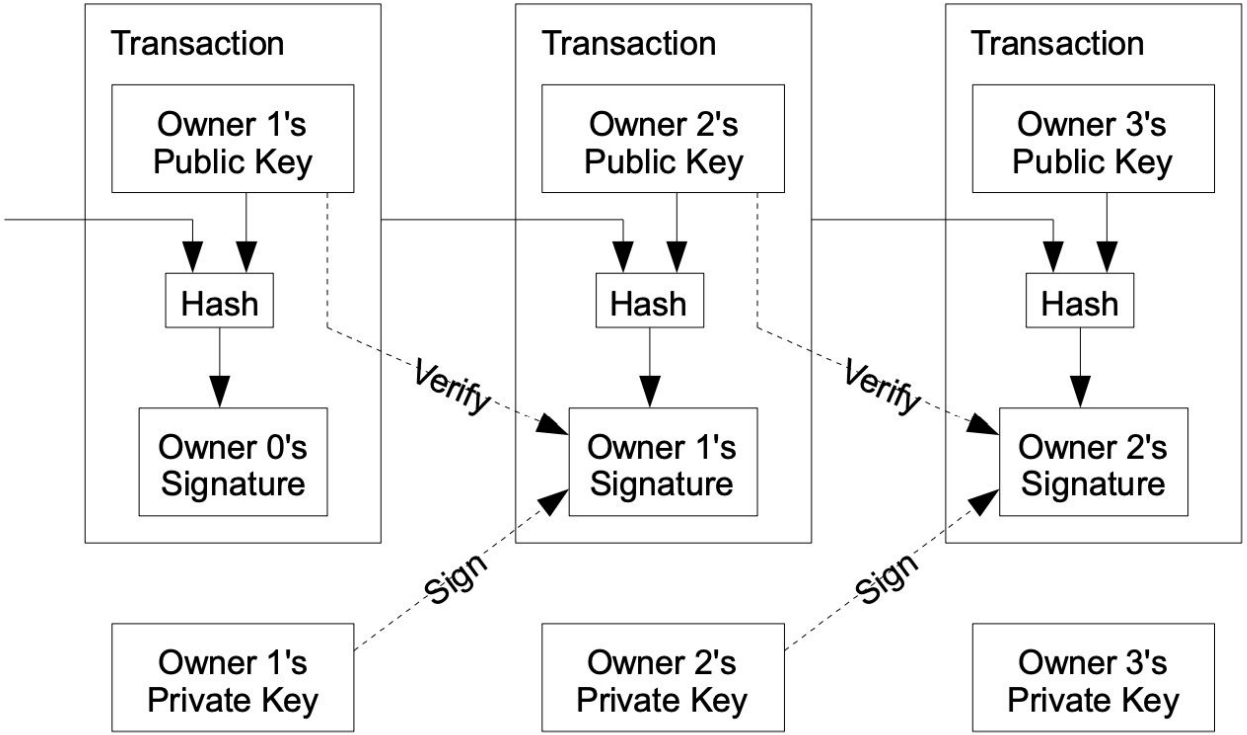
Verificando transações



Sobre privacidade



Transaçōes devem ser assinadas!



Trabalho

Objetivo

Desenvolver uma aplicação em C que implemente uma blockchain funcional. A blockchain deve utilizar o algoritmo **Proof of Work (PoW)** para validação de blocos e deve incorporar uma **Merkle Tree** para estruturar as transações de cada bloco. A aplicação deve incluir uma funcionalidade que permita verificar se uma transação específica está incluída em um bloco através do processo de **Proof of Inclusion**.

Grupos de 3 pessoas. Entrega do código + apresentação até dia 10/02