THE BLOCKCHAIN DATABASE

NOEL T. NEGUSSE

Abstract. The popularity of cryptocurrencies in recent years, particularly bitcoin, was not due to its applications as a digital currency or even its ability to carry out anonymous transactions. It was due to the advent of the blockchain: the transparent yet cryptographically secure, trust free and decentralized system. Blockchains allow people who do not know or trust each other build a robust ledger via an easily verifiable proof-of-work chain. Hence, the blockchain database takes the perfect stance in security, trust no one. This paper will explore the blockchain paradigm, it's pitfalls and prospects and will present a simple implementation of the core algorithm.

1. Introduction

Cryptocurrencies as we knew them were revolutionized in 2009 when 'Satoshi Nakamoto', creator(s) of Bitcoin, introduced the blockchain database and its implementation as a distributed public ledger. A profound concept that would allow people who did not know or trust each other to build a robust leger together through safely proven transactions. And through the added use of peer-to-peer networking and timestamp hashing, Bitcoin became the first decentralized and autonomous digital currency to ever exist.

The blockchain is a distributed public ledger that stores all transactions as a cryptographically secured chain of blocks. This cryptographic characteristics of blocks and the distributed public ledgers (blockchains) were both keys that enabled Bitcoin to become the first digital currency to solve the *double spending problem** without having to rely on the flawed "trust" based model so common in today's world. In other words, Bitcoin became a model that allowed a payment system to be based on cryptographic proof instead of trust. This allows any two given parties the ability to transfer bitcoins between each other without the need of a third party. It also inevitably fosters a more

open and transparent system for people to operate under without worry of a central ledger or power.

2. The Chain of Transactions

We define a transaction to contain an owner's public key, the sender's digital signature and a hash defining the amount being transacted. A single digital coin can be best defined as a singly linked chain of digital signatures. The coin is transferred from one owner (2) to the next (3) by signing the hash of the last transaction [1][see figure 1]. The chain of ownership can also be quickly verified by following the chain of signatures and verifying the owner [1].

Now, while this system is effective, it still allows for double spending. Since the payee can't confirm whether previous transaction owners signed an earlier version. In order to account for this, we introduce a global timestamp hash chain to our blockchain database. The timestamp hash chain will create and append a new hash to its chain when it sees a new block of items on the ledger [1][see figure 2] and publish it globally to the network.

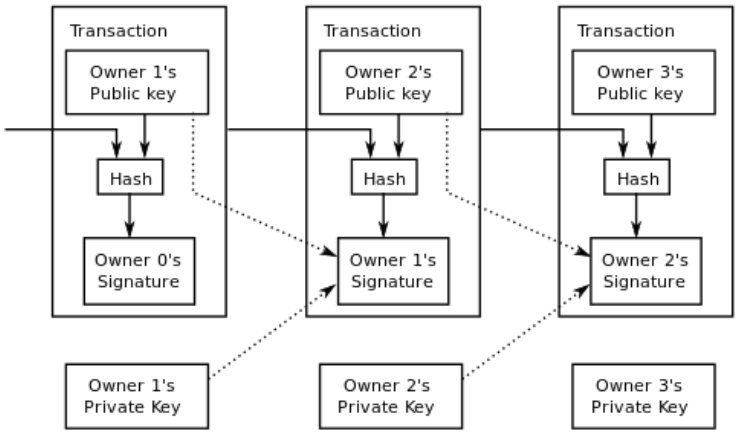


Figure 1

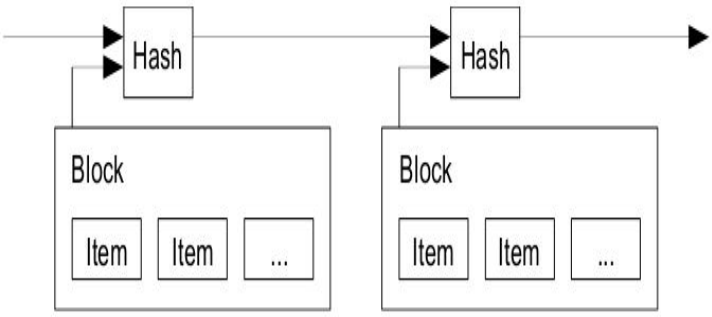


Figure 2

3. The Blockchain

Just to give a better overview of what exactly a bitcoin transaction would look like internally, here's a step-by-step activity workflow:

Step

1. Digitally signed transaction initiation [3]
2. Transaction is sent to miner who verifies transaction [3]
3. Transaction is broadcast to all connected nodes as block [3]
4. Network accepts transaction if data is valid [3]
5. Receiver receives the transaction [3]

In simplest terms, a blockchain is defined as a global ledger that records transactions in the form of a chain of blocks. More specifically, a blockchain database consists of two types of records: transactions and blocks [2]. Transactions are the pieces that form a block, and blocks are the pieces that form the Chain. When blocks of transactions are being constructed, *valid* transactions are hashed and encoded into a Merkle or hash Tree in order to be later verified [1][Figure 3].

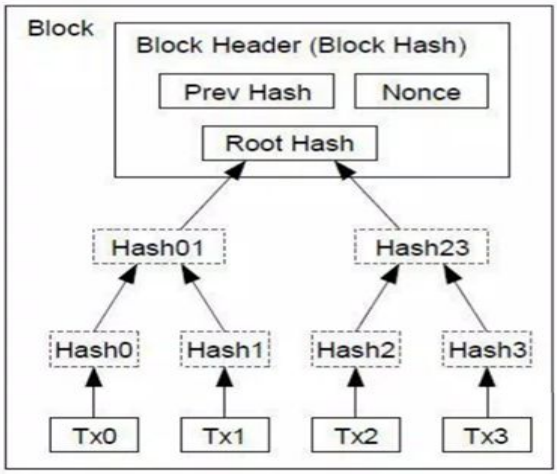


Figure 3 [1]

A blockchain can be thought of very much like a reversed linked-list, in that every block/node has a value (Merkle Tree) and a reference to the hash of the previous block in the chain. A reverse linked-list structure was chosen because when referencing or adding to a blockchain, one only cares about the most recent block/transaction of a chain.

4. Decentralization and Concurrency

A blockchain database is a highly concurrent, distributed system with a notably high *byzantine fault tolerance** . This is in part due to the extremely high difficulty of the proof-of-work necessary to add new blocks. And by virtue of a decentralized system, every node and miner in the network has a copy of the "official" blockchain. The data is distributed throughout the peer-to-peer network via massive data replication, meaning no user is 'trusted' more than any other. Additionally, if it does so happen that two blocks complete simultaneously, creating a concurrency issue and producing forked blocks, the system is designed to take the difficulty or *nonce* value in order to prioritize which to keep.

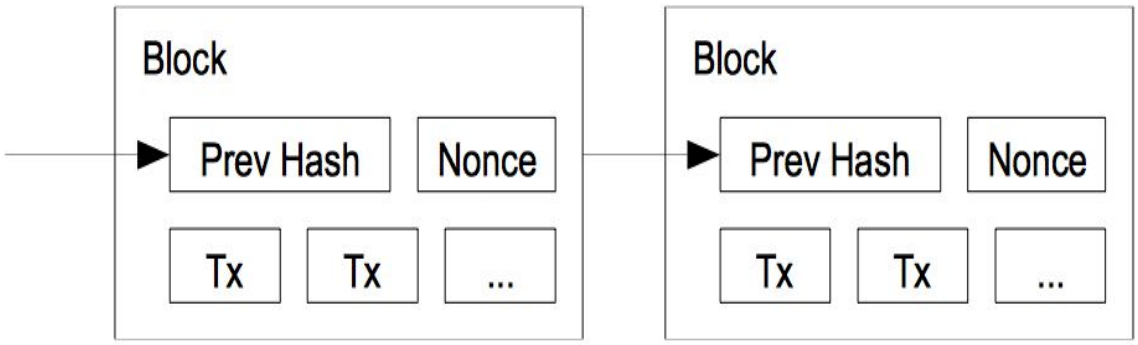
A decentralized distributed system can lead to a lot of unique concurrency issues and bugs, but it can also yield a lot of strong bonuses. By storing the data across the peer-to-peer network (much like your assets), the blockchain completely eliminates the risks and dangers of holding data in a centralized point. And as with all distributed system, the system has a much easier time dispersing and spreading load.

Blockchains also contain a simple but effective security system for individual parties who wish to locate or access their bitcoins (or digital assets). A public key cryptography cryptosystem is used, where the public key, in this instance, represents the address of the blockchain somewhere on the network which contains the owner's bitcoins and the private key acts as the password to retrieve said coins. This method works especially well since all data stored in a blockchain is treated as immutable before the head, and therefore incorruptible. This ensures the validity of the public keys address.

5. Proof-of-Work System

A proof-of-work system is normally a computation that is costly and time-consuming to produce but still easy for someone to verify. It's purpose historically was to deter or discourage DoS and spam by raising the cost (CPU cycles) of entry. Bitcoin, and consequently, blockchains use proofs-of-work systems, HashCash, for block generation in the global ledger (blockchain). In order for a block to be safely accepted by members of the p2p network, 'miners' must complete a proof-of-work that incorporates all of the data in the current blockchain. This is done by encoding all of the transactions in a given block into a Merkle or hash tree which will produce a given root or top hash. This top hash in conjunction with some random difficulty, $nonce$, will have to solve $Y = H(X)$ for X s.t. , $X < nonce$ where $Y = tophash$, H is the SHA256 hash function. At first glance it may seem that a computation this intense may take years in some cases, and it truly can. The idea here is that, despite the massive amount concurrent bitcoin miners, by creating a problem with a very low probability of success we can throttle the amount of blocks being written at once, as well as prevent malicious blocks from getting written onto the global transaction block chain.

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. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest miners. This means the only way for an attack to happen on our blockchain, would be to attempt to create a double spending problem by creating a fork in our local blockchain and then trying to outrun all the other miners solving proof-of-work puzzles. The longest chain not only serves as proof of the sequence of events witnessed, but is an inductive proof that it came from the largest possible 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 [1].



7. Conclusion

Blockchains revolutionized cryptocurrencies as we know them and set the stage for the open, peer-to-peer model of decentralized, autonomous digital currencies. Blockchain databases paved the way for simplistic and robust systems that for the first time, steered away from the idea of ‘trusts’, but onto more concrete cryptographic proofs and models.

Double Spending Problem: A failure mode of digital currency systems, which makes it possible to spend the same digital token twice.

Byzantine Fault Tolerance: A system that holds a tolerance to concurrency issues that prevent components of a system from reaching a common agreement, when necessary for correct operation

References

[1] Nakamoto, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*. *www.bitcoin.org/*. <https://bitcoin.org/bitcoin.pdf>.

[2] Economist Staff (2015-10-31). "Blockchains: The great chain of being sure about things". The Economist. Retrieved 18 June 2016.

[3] "Blockchain - Secured way of transaction". iFour Technolab Pvt. Ltd. Retrieved 2016-11-15.