Bitcoin: Uncovering the Hidden Mathematics

Michael Youniss

May 3, 2016

Abstract

This paper aims to give an overview of what the Bitcoin protocol is and discuss the underlying mathematics which secure the anonymous transactions.

1 Introduction

Bitcoin is an *online decentralized currency. Online* means there is no tangible currency, all transactions take place online between two anonymous people. *Decentralized* means there is no government or bank regulating the currency. *Currency* means Bitcoin transactions occur to trade goods and services.

The core concept underlying all of Bitcoin is the transaction. A transaction is represented in Bitcoin by a hash created by a wallet. There are no banks regulating Bitcoin, nor is there a government, thus in order to access bitcoins one must own a bitcoin wallet. Bitcoin wallets are not tangible, the same way bitcoins are not tangible; however, people own bitcoin wallets which allow them to store and spend bitcoins. All bitcoin transactions occur between two wallets.

Once a transaction occurs between two wallets, the transaction is broadcast out to a network of listening nodes, each node is called a miner. Miners add each broadcast transaction to a block of transactions. This block of transactions is then added to a chain of blocks of all the other transactions that have occurred in the Bitcoin protocol. This concept of adding transactions to a block and blocks to a chain is known as blockchain.

Blockchain allows users to see the history of all transactions and verify that a wallet received the bitcoins it is exchanging at one point in time. The blockchain protocol is essential to Bitcoin maintaining its decentralized nature. No bank is needed to prove an individual has money, rather, the history of all transactions will prove that. Due to its important nature, miners are rewarded bitcoins for successfully adding properly formatted blocks to the blockchain.

2 Transactions

The transaction is the core principle underlying the Bitcoin Protocol. A Bitcoin transaction takes place between wallets and contains the following information: a hash, meta data, ins, and out.

- Hash: Represents all the data in the transaction as a long number
- Meta data: Contains information about the transaction such as the version of the Bitcoin protocol being run, the number of inputs and number of outputs
- Ins: Number of input transactions
- Outs: Number of output transactions

Bitcoin wallets do not mix input transactions together. That is to say if a wallet received 3 bitcoins, 5 bitcoins, and 20 bitcoins as separate transactions, it will have 28 bitcoins but it stores the transactions separately within the wallet. Therefore, if the wallet were to pay 14 bitcoins, it would send out the 20 bitcoin transaction and ask for a new 6 bitcoin transaction to come back. Thus the number of Ins would be 1 (the 20 bitcoin transaction) and the number of outs would be 2 (the 14 bitcoins leaving, and the 6 bitcoins coming back). Inputs will always be the outputs of previous transactions.

Transactions must be signed by the payer to ensure that the bitcoins being paid are valid. Due to the online and decentralized nature of Bitcoin, these payers are anonymous and thus the only way to ensure a payer is attached to a transaction is through a digital signing algorithm. Bitcoin uses an Elliptic Curve Digital Signing Algorithm.

2.1 Elliptic Curve Digital Signing Algorithm (ECDSA)

The Bitcoin protocol chooses a finite field $\mathbb{F}_p$, an elliptic curve $(\mathrm{E}/\mathbb{F}_p)$ and a point $G \in E(\mathbb{F}_p)$ that has order q, where q is a large prime number.

In the example below lets assume Wallet A is sending Wallet B some bitcoins and wants to sign the transaction. The table below will walk through the math of ECDSA.

Wallet A	Public	Wallet B
	$\mathbb{F}_p$	
	E/$\mathbb{F}_p$	
	$G \in E(\mathbb{F}_p)$	
Choose a private key k		
$1 < k < q - 1$		
Calculate a public key V		
$V = kG \in E(\mathbb{F}_p)$	V	
Choose the transaction to sign		
$t \mod q$		
Calculate $tG \in E(\mathbb{F}_p)$		
$s_1 = f(tG) \mod q$		
where f(x,y) returns x value		
$s_2 = t + k * s_1$		
Signature = (s_1, s_2)	Signature	
		Calculate $v_1 = ts^{-1} \mod q$
		$v_2 = s_1 s_2^{-1} \mod q$
		Calculate $v_1 G + v_2 V \in E(\mathbb{F}_p)$
		verify $f(v_1 G + v_2 V) \mod q = s_1$

Bitcoin relies on the ECDSA for transaction receivers to verify that their transaction is coming from the proper wallet. Once the transaction is verified, it can be broadcast out to the network where there are miners listening and attempting to put it in a block and chain that block to the rest of the blockchain. Then once a part of the blockchain, the receiver will see the output of the transaction in their wallet.

3 Block Chain

In the Bitcoin protocol there is a history of every transaction stored in the blockchain. The blockchain is blocks of transactions, chained together, in chronological order from the beginning of Bitcoin to the most recent block added.

Blockchain records all the Bitcoin transactions for all users to see. With proper information, one can find a block holding a past transaction and verify the transaction was issued to the wallet claiming to have it. Blockchain is necessary for a decentralized currency, like Bitcoin, because this is the one way for the community of users to verify the legitimacy of users and transactions. Due to its importance, there is reward for the person who comes up with the next block that follows the rules as set out by Bitcoin and is accepted by the rest of the bitcoin users. The person who comes up with a block to add to the chain is called a miner. The miner compiles a list of recent transactions which have been broadcast out to the Bitcoin network. Next, the miner decides which transactions to include in the next block. The miner can ignore transactions of foes, the miner can reject illegitimate transactions, or the miner can reject transactions that do not include a transaction fee (an amount of bitcoin transferred to the miner who successfully includes that transaction into the blockchain). Finally, the miner adjusts a value known as the nonce accordingly until the hash of all the information in the block is less than a predetermined value by the Bitcoin protocol. This last step requires the most computation power, or luck, to successfully accomplish.

There are four main components in a block in the blockchain: the hash of the previous block, the hash of the reward transaction, the hashes of all transactions included in the block, and the nonce. The hash of the previous block ensures that from the most recent published block, one can trace the hashes all the way back to the first block published. The reward transaction is an amount of bitcoins (currently 25 bitcoins) rewarded to the miner who successfully adds the next block to the chain. The reason this act of adding a block is called mining is because of the bitcoin reward awarded to the miner responsible for adding the next block. The hashes of transactions included in the block are all the recent transactions that have been broadcast to the network that the miner decided to add to this block. Amongst these transactions might be transaction fees which will award the miner more bitcoins for including the transaction in the blockchain. Finally, the nonce is an integer value adjusted until the hashing function returns an acceptable hash.

The hashing for Bitcoin blocks uses Hashcash, which uses SHA-256[2] (a 256 bit hashing function) in its implementation in Bitcoin. All the information stored in the block is then fed into Hashcash which returns a hash. Call the Hashcash hashing function $H(x, n)$ where x is all the hashes (previous block, reward transactions, and transactions included), and n is the value of the nonce. H will return a 256 bit binary string that represents the entire block of data. Bitcoin protocol for an acceptable hash looks like the following equation: $H(x, n) < 2^{256-k}$ where k is a value adjusted by the Bitcoin protocol in real time. $2^{256} - 1$ is the value of a 256 bit binary string composed entirely of 1's. 2^{256-k} is a 256 bit binary string with k leading 0's and then $256 - k$ 1's. The Bitcoin protocol requires the hash result to be less than 2^{256-k}, or in other words, the Bitcoin protocol requires the hash function to have k leading 0's. There exist 2^{256} distinct binary strings that can be returned by the hashing function, and only 2^{256-k} valid binary strings. The odds of the hash function returning a valid hash are $\frac{1}{2^k}$. If k were set to 20, there would be about 1 in a million chance of the hash function returning a valid hash. Like all hashing functions, by slightly changing the nonce the hash result is drastically different than the hash produced using the previous nonce value; therefore, a miner must compute the hashes for millions of nonces before coming across a valid hash. Meanwhile, there are other miners competing with the same data to come up with a valid block and hash first. The power of miners and number of miners is constantly changing on the Bitcoin network; however, the rate at which blocks are added is supposed to be constant at one block per every ten minutes. Thus, Bitcoin protocol changes k based on the rate blocks have been added to make it easier or harder to successfully mine a block to ensure a constant rate of block addition. Mining a block successfully requires a lot of computation power, and for that power miners are rewarded.

4 Security

Bitcoin is a decentralized currency which means all nodes in the network have an equal amount of power. Bitcoin is thus susceptible to attack; however, Bitcoin has built in features to protect itself from a group of ill willed nodes taking advantage of the decentralized currency.

In the Blockcahin protocol each block has the hash of the previous block in it. When a miner attempts to make a valid hash, the hash of the previous block and all the transactions within it must be combined with a proper nonce to construct a well formatted hash. Due to the difficulty of creating this hash, it is inconceivable that an attacker could go back into the blockchain protocol and change any previous transactions or blocks because in so doing they would break the entire blockchain all other good willed users have subscribed to. Blockchain in turn protects against attackers trying to edit the past. The only possible attack Bitcoin is susceptible to is a dishonest node spending Bitcoin and reclaiming those bitcoin in the same transaction block. In so doing, the attacker receives a good or service without losing any bitcoin.

It is not possible to have conflicting transactions in the same block. Instead, once a transaction that an attacker wants to reverse is accepted into a block, the attacker begins constructing a parallel chain. The first block in the parallel chain will contain the dishonest transaction where the attacker does not lose any bitcoins. The attacker hopes to build up blocks as quickly as the rest of the network to get the new parallel chain adopted. In the case the chain is adopted, the sender then does not lose the bitcoin hoping the good or service has already been delivered and it is too late for the supposed to be recipient of the bitcoin to do anything about losing the bitcoins.

In order to understand the likelihood of such an attack, lets examine the math:

Let t be the probability an honest node discovers the next block.
Let a be the probability an attacker finds the next block (This value can also be thought of as the percentage of miners working maliciously together to discover the next block. This gives each miner an equal chance of discovering the next block but the attackers an advantage by working together).
Let u be the number of blocks in the accepted chain the attacker is behind.
Let a_u be the probability the attacker will catch up to the accepted chain if behind by u blocks.

Assume $t > a$ because it is our assumption more than half the participants in the Bitcoin network are honest users. Because of the generation of public keys at the time of transaction, the attacker cannot begin work on his attack until the moment the transaction occurs. Once the transaction is processed, the recipient of the attackers transaction needs to wait u blocks before transferring the good or service. As Satoshi points out in his paper, the expected value of the attacker's a malicious chain that could be adopted by the network is: $\lambda = u\frac{a}{t}$. Satoshi goes on to show that the probability that the attacker catches up from u blocks behind is: $1 - \sum_{k=0}^{u} \frac{\lambda^k e^\lambda}{k!}(1 - \frac{a}{u}^{(u-k)})$. In order to ensure a $< 0.1\%$ chance of the attacker successfully taking their money back, the following table shows the values for a and u.

a	u
10%	5
15%	8
20%	11
25%	15
30%	24
35%	41
40%	89
45%	340

If 10% of users are working together to build malicious chains, this gives the attackers a 10% chance of discovering the next node. To ensure their failure, a recipient of bitcoins only needs to wait about an hour to deliver the good or service. This gives the attackers less than a 0.1% chance of stealing their bitcoins back and receiving the good or service for free. One block is published every ten minutes, and the attacker needs to be behind by five blocks, which is roughly an hours worth of time. As the percentage of anticipated attackers working together to steal back bitcoins increases, the amount of time a receiver of bitcoins must wait before exchanging a good or service to ensure an attacker cannot steal back bitcoins. A receiver of bitcoins can all but ensure their bitcoins are secured through delaying delivery according to their opinion of the percentage of attackers working together on the network.

5 Conclusion

Bitcoin is built on top of mathematics. Cryptography algorithms are used to transfer Bitcoins from one wallet to another. A hashing function is used to record each every transaction in the history of Bitcoin and ensure the recording is done at a steady rate. Probability is used too demonstrate the strength of bitcoin against a potential attacker. This online, decentralized currency can only exist securely due to its underlying mathematical architecture.

6 Bibliography

An Introduction to Mathematical Cryptography — Jeffrey Hoffstein — Springer. (n.d.). Retrieved April 27, 2016, from http://www.springer.com/us/book/9781493917105

How a Bitcoin Transaction Works. (2014). Retrieved April 27, 2016, from https://www.cryptocoinsnews.com/bitcoin-transaction-really-works/

How bitcoin mining works - CoinDesk. (n.d.). Retrieved April 27, 2016, from http://www.coindesk.com/information/how-bitcoin-mining-works/

How the Bitcoin protocol actually works. (n.d.). Retrieved April 27, 2016, from http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Main Page. (n.d.). Retrieved April 27, 2016, from https://en.bitcoin.it/wiki/

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved April 27, 2017, from https://bitcoin.org/bitcoin.pdf

The Math Behind Bitcoin - CoinDesk. (2014). Retrieved April 27, 2016, from http://www.coindesk.com/math-behind-bitcoin/

The Proof-of-Work Concept Daniel Krawisz. (n.d.). Retrieved May 03, 2016, from http://nakamotoinstitute.org/mempool/the-proof-of-work-concept/#selection-143.378-143.386