# The Mathematics behind Blockchain

Partho Sutra Dhor
parthosutradhor@gmail.com

**Abstract:** One of the technologies that have revolutionized the world in the last decade is Blockchain. It is a public ledger or distributed database where information is verified based on the opinions of the majority of participants. This blockchain technology is used in many cases, an excellent example of which is Bitcoin. Blockchain is democratizing and decentralizing the centralized economy and information system, which is truly an unprecedented breakthrough in digital security. This article will go through the mathematical details of Blockchain technology and its future.

## 1. Introduction:

We are very much dependent on a centralized system where all our sensitive information is stored that can be stolen, corrupted or modified. Blockchain [1] was first proposed in 1991. About two decades later, Satoshi Nakamoto [2] proposed a peer-to-peer transaction system called Bitcoin in 2009. Imagine a bank with no central party controlling the transaction and no central database. Centralized systems use client/server architecture where one or more client nodes are directly connected to a central server. But in this system, each client manages their own database. In a Blockchain system, it is almost impossible to alter the chain of blocks/databases because to alter a block, one has to use some computational power or proof of work called mining. Since all the block are connected by a chain, a change in a single block spoil all the subsequent blocks. In this article, we will investigate Blockchain's mechanisms and technical details.

## 2. Cryptographic Hash Function:

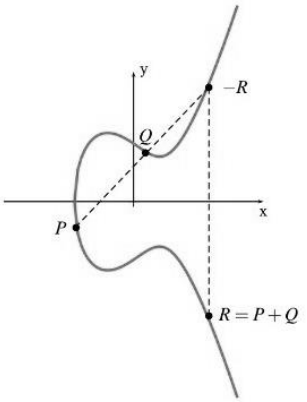
A cryptographic hash [5] function is a mathematical function whose domain is the set of all data bytes of any size, and the codomain is the set of all fixed-size data bytes or strings. A hash function must satisfy some rules:

1) Output must be fixed size
2) A small change in input creates an incredible difference in output
3) Theoretically impossible to find an inverse function.

Some commonly used functions are MD5, SHA1, SHA256 etc.

## 3. Elliptic Curve Cryptography (ECC):

An elliptic curve [6] is an equation such as $y^2 = x^3 + ax + b$. In Bitcoin [4] and most other implementations, a = 0 and b = 7, so this is simply $y^2 = x^3 + 7$. All coordinate points in an elliptic curve make an additive abelian group. Consider $E = \{(x, y)\ on\ elliptic\ curve\}$. The addition is well defined if $P = (p_1, p_2)$ and $Q = (q, q_2) \in E$ then $P + Q\ is$ the point on the elliptic curve, which is the reflection of the x-axis of the point where the joining line of P and Q intersect. Multiplication by an integer *m* is equivalent to adding *m* times itself.




The elliptic curve cryptography is algebraic cryptography over a finite field [3] $\mathbb{F}p$. All the algebraic field operations (additions and

multiplications) of two points in the elliptic curve result in another point on this curve. This process is done with modulo operation $y^2 \equiv x^3 + ax + b \pmod{p}$.
In ECC, we have:

- Elliptic curve (EC) over finite field $\mathbb{F}p$
- G is the generator point (fixed constant, a base point on the EC)
- k is the private key (integer)
- P = k * G is the public key (point)

Using the well-known ECC multiplication techniques in time $\log_2 k$, such as the "double-and-add algorithm", it is quick to calculate P = k * G. It will require a few hundred straightforward EC operations for 256-bit curves. Calculating k = P / G is extremely time-consuming and considered impossible for large k. The ECC cryptography, often known as the ECDLP [7] problem, is based on this asymmetry (rapid multiplication and impractical slow opposite operation).

# 4. Digital Signature Verification:

The elliptic curve digital signature algorithm (ECDSA) takes a message *msg* + a private key *PrivKey* as input and produces a signature {r, s} by this algorithm

1. $h$ = hash(*msg*)
2. Generate a random number $k$ (*Private Key*) securely in the range $[1, \dots n]$
3. $R = k * G$ and take its $x$-coordinate $r$
4. Signature $s = k^{-1} * (h + r * PrivKey) \ (mod\ n)$
5. Return the signature {r, s}.

To verify an ECDSA signature, take the signed message *msg* + the signature *{r, s}* produced from the signing algorithm + the public key *PubKey*, corresponding to the signer's private key, *PrivKey*. The output is a Boolean value: valid or invalid signature.

1. $h$ = hash(*msg*)
2. $R' = (h * s_1) * G + (r * s_1) * PubKey$ where $s_1 = s^{-1} \ (mod\ n)$
3. Take its x-coordinate $r'$ from $R'$
4. Calculate the signature validation result by comparing whether $r' == r$

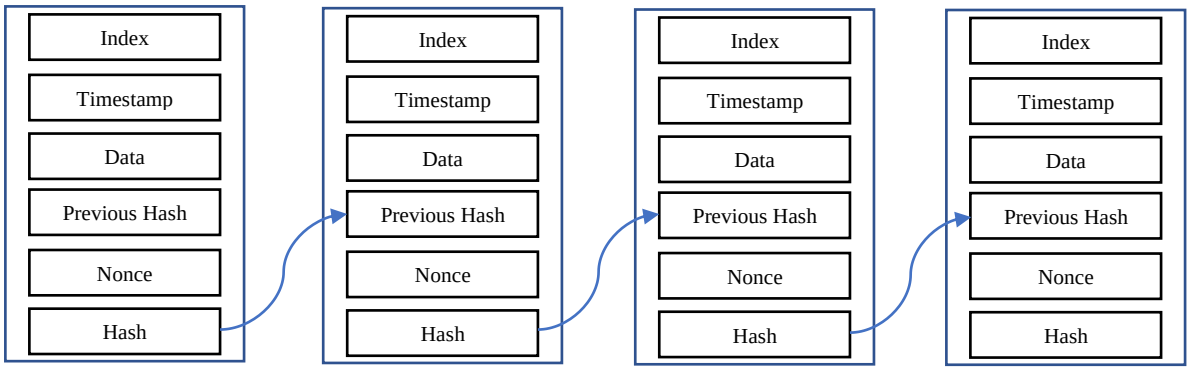
The basic principle behind signature verification is to use the public key to recover the point $R'$ and confirm that it is the exact point R produced randomly during the signing procedure.

# 5. Blocks and Blockchain:

The head of the block is divided into six components:

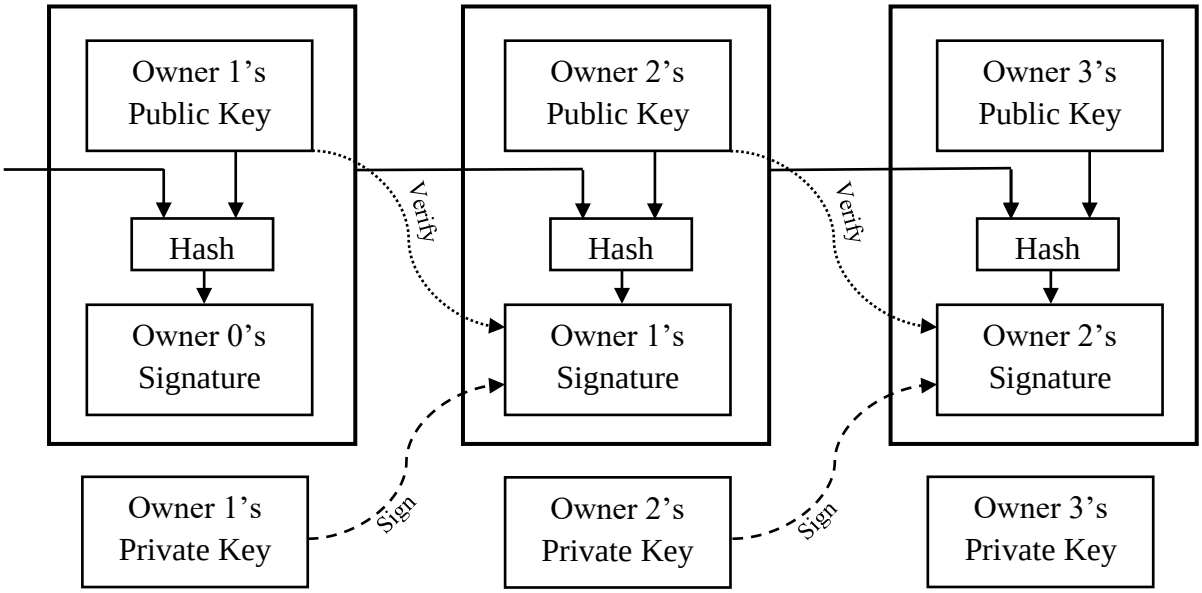
1. Index
2. The time in seconds since 1970–01–01 T00: 00 UTC
3. Data of transactions
4. Hash of the previous block
5. The Nonce
6. Hash of that block

The concept which makes the Blockchain unalterable is the proof of work. Here we use a nonce, a random number, to be determined to make the hash of the whole block start with specific numbers of zeros. In some crypto-currencies, the proof of work or finding nonce starts the hash with 19 zeros. All the preceding block hash must be mined if a block is altered. This needs a considerable amount of computational power that is practically impossible. The time goes by, and the chain of blocks gets larger. All the users agree that the ledger has the highest computational work.




## 6. Transactions:

The following mechanism makes transactions. Clients use the Elliptic curve cryptography method to sign their dealings with the private key and verify with the public key.




## 7. Is Blockchain the Future?

Blockchain stands on the strength of cryptography. So far, Blockchain is secure technology even though quantum computers can break RSA and EEC. The most popular public-key cryptosystems (like RSA and ECDSA) are quantum-broken. Most digital signature algorithms (like RSA and ECDSA) are quantum-broken. But Cryptographic hashes (like SHA2 and SHA256) are considered quantum-safe. If we can implement quantum-safe asymmetric key encryption, it will produce a reliable, uncensored, accessible global data and information storehouse. This feature will fuel the third generation of the internet. And for this reason, the internet's future lies with the Blockchain.

## References


[1] "Blockchain - Wikipedia." *Blockchain - Wikipedia*, en.wikipedia.org, 23 May 2016, https://en.wikipedia.org/wiki/Blockchain.

[2] Nakamoto, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*.

[3] Bhattacharya, P. B., et al. *Basic Abstract Algebra*. 2012, https://doi.org/http://dx.doi.org/10.1017/CBO9781139174237.

[4] "Bitcoin - Wikipedia." *Bitcoin - Wikipedia*, en.wikipedia.org, 9 Jan. 2009, https://en.wikipedia.org/wiki/Bitcoin.

[5] "Hash Function - Wikipedia." *Hash Function - Wikipedia*, en.wikipedia.org, 1 July 2010, https://en.wikipedia.org/wiki/Hash_function.

[6] "Elliptic-Curve Cryptography - Wikipedia." *Elliptic-Curve Cryptography - Wikipedia*, en.wikipedia.org, 8 Jan. 2020, https://en.wikipedia.org/wiki/Elliptic-curve_cryptography.

[7] Hankerson, D., Menezes, A. (2011). Elliptic Curve Discrete Logarithm Problem. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-5906-5_246