BASIC BLOCKCHAIN FUNCTIONING

As an element of universal business order, the basis of information is supplied by the organizations that facilitate its transportation; the accuracy of information and the speed at which it can be transmitted are extensive and vital subsets of all transactional mediums. Blockchain, the central engine that drives the trade and utility of cryptocurrency, is a publicly accessible and immutable ledger that translucently oversees network transactions—a software that, although immaterial, has the capacity to efficiently track both tangible and intangible assets, securing the commercial process for all participants while recording the history of all transactions made on it (including the traceroute). The virtue of the blockchain's unrivaled security proves critical utility for those who interact with cryptocurrency-based transactions, yet not many of them fully realize the mathematical concision resting behind the screen. The remainder of this article explains how, exactly, blockchain functions as it does, overviewing the algorithmic structure of the software in synoptic brief. Subdividing the mathematical foundations of the blockchain into three primary pieces, we seek to inform the reader of hash tables, RSA encryption, and the ECDSA.

Consistent with the decentralized nature of the blockchain, hash tables are a means of storing information. Since the blockchain cryptographically links blocks through hashed records of previous blocks, timestamps, and other data, hash tables are useful for maintaining information about every historic transaction and serve as the fundaments of all cryptocurrencies' storage systems. The tables are arranged in nodes, with each one hosting a set of key-value pairs. To construct a strong understanding of this deep storage process, we must introduce the concepts of both hash functions and keys. Primarily, hash functions serve to accept data (sets of inputs of whose length does not matter) and translate them into entries in a matrix or hash table. Consider three pieces of data being fed into the hash function, resulting in the following tabular output:

INDEX	**VALUE**
0	Primary Input (Data 1)
1	Secondary Input (Data 2)
2	Tertiary Input (Data 3)

As a result of the hash function, keys are produced. They act as unique identifiers in hash tables, which can be used, through a precise process, to allow operation on a cryptographic hash of specific data. To conclude a brief introduction to hash tables, they are systems of organization that allow computers to operate mathematically instead of having to sift through a list of values, one at a time, before arriving at a desired location or empty slot (the former is a much more efficient method, especially when tasked with searching through extremely extensive records).

On the frontend and backend spheres, public and private keys are utilities that parties may use to receive a message in its entirety, unencoded. Suppose Alice intends to send a message to Bob (these names are classical monikers in cryptographic scenarios), each of which represent a party on one end of a blockchain transaction—for example, the exchange of USD to BTC or purchase of an NFT. In this situation, Alice and Bob each have both a publicly-accessible key and a private one. For each party, the public key acts as an encryption function, whereas the private key acts as a decryption function. If Alice wanted to send a message to Bob, she would first use his public key to encrypt her message. On Bob's end, he would use his private key to decrypt that message, allowing for an intelligible reading of the contents that were sent to him. Mathematically, the keys are inverse functions to one another, acting as complex forms of encryption and decryption that are nearly impossible to bypass or intercept from an outside vantage point.

After the completion of digital transactions through blockchain, all parties use an elliptic curve digital signature algorithm (ECDSA) to certify the acceptance of the financial interaction. Using elliptic curves instead of modular exponentiation and the discrete logarithm problem, the ECDSA—an elliptic-curve cryptosystem variant of the digital signature algorithm (DSA)—uses private keys to facilitate signed message-sending. These curves are of the form

$$y^2 = x^3 + ax + b$$

for some coefficients in a field K. With unique functions, certain characteristics of elliptic curves make their use in key recognition and message encryption so promising; one of these, for instance, is that a nonvertical line intersecting an elliptic curve at two known points will again intersect it at a third, calculable point. With the ECDSA, a public key is of the form of an elliptic curve function and a point that lies on that curve, while the private key is a number. In essence, the algorithm encodes messages by protecting them with high security levels and with the elliptic curve system. With up to 256 bits of security, attackers and maleficent programs would need to process 2^{256} operations to gain access to private keys.

As the most significant driver of cryptocurrencies and online exchange networks, the blockchain is an exceptionally algorithmic and systematically put-together software that will, undoubtedly, further evolve to mold the future needs of the digital economy.