CRYPTOGRAPHY

CRYPTOCURRENCES

NO CRYPTOGRAPHY then **NO SECURITY**

By,

DINESH SERVAMSETTY

MSC-COMPUTER SECURITY

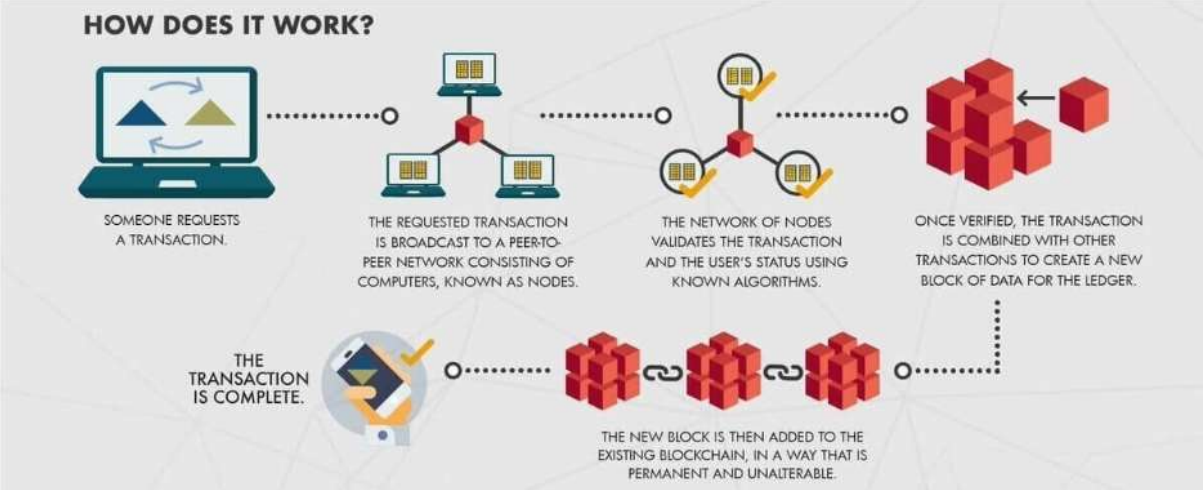
INDEX

This page is intentionally left.

1. What is cryptocurrency and its history.

- A cryptocurrency is a digital asset designed to work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets. Cryptocurrencies use decentralized control as opposed to centralized digital currency and central banking systems.

  Bitcoin represents the first decentralized cryptocurrency, which is powered by a public ledger that records and validates all transactions chronologically, called the Blockchain. Here's an overview of how the blockchain works:



History:

- Back to the early 1980s, an American cryptographer named David Chaum invented a "blinding" algorithm or anonymous cryptography called **e-cash** that remains central to modern web-based encryption. The algorithm allowed for secure, unalterable information exchanges between parties, laying the groundwork for future electronic currency transfers. This was known as "blinded money."

- By the late 1980s, Chaum enlisted a handful of other cryptocurrency enthusiasts in an attempt to commercialize the concept of blinded money. After relocating to the Netherlands, he founded DigiCash, a for-profit company that produced units of currency based on the blinding algorithm. Unlike Bitcoin and most other modern cryptocurrencies, Digi Cash's control wasn't decentralized. Chaum's company had a monopoly on supply control, similar to central banks' monopoly on fiat currencies.

- Digi Cash initially dealt directly with individuals, but the Netherlands' central bank cried foul and quashed this idea. Faced with an ultimatum, Digi Cash agreed to sell only to licensed banks, seriously curtailing its market potential. Microsoft later approached Digi Cash about a potentially lucrative partnership that would have permitted early Windows users to make purchases in its currency, but the two companies couldn't agree on terms, and Digi Cash went belly-up in the late 1990s.

- Around the same time, an accomplished software engineer named Wei Dai published a white paper on b-money, a virtual currency architecture that included many of the basic components of modern cryptocurrencies, such as complex anonymity protections and decentralization. However, b-money was never deployed as a means of exchange.

- Shortly thereafter, a Chaum associate named Nick Szabo developed and released a cryptocurrency called Bit Gold, which was notable for using the blockchain system that underpins most modern cryptocurrencies. Like Digi Cash, Bit Gold never gained popular traction and is no longer used as a means of exchange.

Cryptocurrency Examples:

I. Bitcoin:

Alice wants to buy apple which Bob has for sale. In return, she must provide something of equal value to Bob. The most efficient way to do this is by using a medium of exchange that Bob accepts which would be classified as currency.

Alice doesn't necessarily need to be in direct contact with bob in order for the funds to be transferred. She may instead transfer this value by first entrusting her currency to a bank who promises to store and protect Alice's currency notes. The bank gives Alice a written promise (called a "bank statement") that entitles her to withdraw the same number of currency bills that she deposited. Since the money is still Alice's, she is entitled to do with it whatever she pleases, and the bank (like most banks), for a small fee, will do Alice the service of passing on the currency bills to Bob on her behalf. This is done by Alice's bank by giving the dollar bills to Bob's bank and informing them that the money is for Bob, who will then see the amount the next time he checks his balance or receives his bank statement.

Each balance is simply associated with an address and its public-private key pair. The money "belongs" to anyone who has the private key and can sign transactions with it. Moreover, those keys do not have to be registered anywhere in advance, as they are only used when required for a transaction. Transacting parties do not need to know each other's identity in the same way that a store owner does not know a cash-paying customer's name.

A Bitcoin address mathematically corresponds to a public key and looks like this: 1PHYrmdJ22MKbJevpb3MBNpVckjZHt89hz

Each person can have many such addresses, each with its own balance, which makes it very difficult to know which person owns what amount. In order to protect his privacy, Bob can generate a new public-private key pair for each individual receiving transaction and the Bitcoin software encourages this behaviour by default. Continuing the example from above, when Charlie receives the bitcoins from Bob, Charlie will not be able to identify who owned the bitcoins before Bob.

Each person, such as Alice or Bob, has one or more addresses each with an associated pair of public and private keys that they may hold in a wallet. Only the user with the private key can sign a transaction to give some of their bitcoins to somebody else, but anyone can validate the signature using that user's public key.

Suppose Alice wants to send a bitcoin to Bob.

- Bob sends his address to Alice.
- Alice adds Bob's address and the amount of bitcoins to transfer to a message: a 'transaction' message.
- Alice signs the transaction with her private key, and announces her public key for signature verification.
- Alice broadcasts the transaction on the Bitcoin network for all to see.

(Only the first two steps require human action. The rest is done by the Bitcoin client software.)

Looking at this transaction from the outside, anyone who knows that these addresses belong to Alice and Bob can see that Alice has agreed to transfer the amount to Bob, because nobody else has Alice's private key. Alice would be foolish to give her private key to other people, as this would allow them to sign transactions in her name, removing funds from her control.

Later on, when Bob wishes to transfer the same bitcoins to Charlie, he will do the same thing:

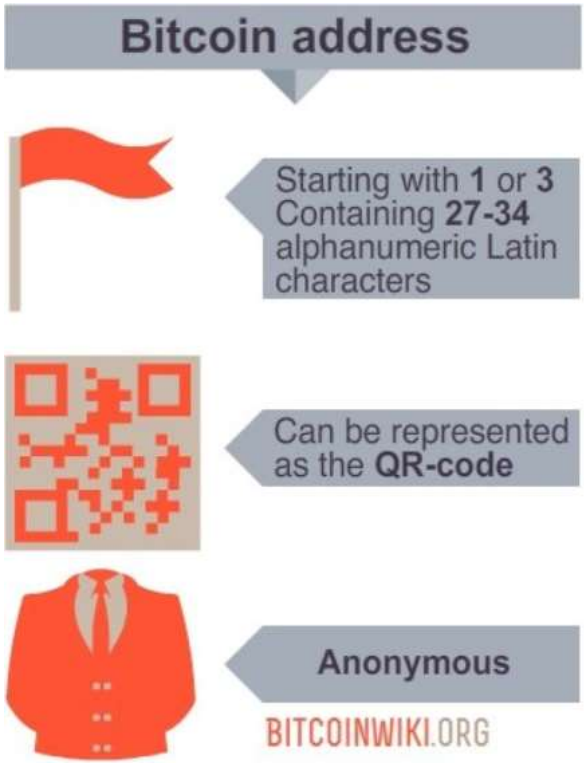
- Charlie sends Bob his address.
- Bob adds Charlie's address and the amount of bitcoins to transfer to a message: a 'transaction' message.
- Bob signs the transaction with his private key, and announces his public key for signature verification.
- Bob broadcasts the transaction on the Bitcoin network for all to see.

Only Bob can do this because only he has the private key that can create a valid signature for the transaction.

Eve cannot change whose coins these are by replacing Bob's address with her address, because Alice signed the transfer to Bob using her own private key, which is kept secret from Eve, and instructing that the coins which were hers now belong to Bob. So, if Charlie accepts that the original coin was in the hands of Alice, he will also accept the fact that this coin was later passed to Bob, and now Bob is passing this same coin to him.

Bitcoin Address: Bitcoin address is an identifier (account number), starting with 1 or 3 and containing 27-34 alphanumeric Latin characters (except 0, O, I). Bitcoin addresses can be also represented as a QR-code. The addresses are anonymous and do not contain information about the owner.

There are currently two bitcoin address formats in common use:
Common Pay-to-Pub key Hash (P2PKH) which begin with the number 1. Newer Pay-to-Script Hash (P2SH) type starting with the number 3,
eg: 35bSzXvRKLpHsHMrzb82f617cV4Srnt7hS.
Most Bitcoin addresses are 34 characters. They consist of random digits and uppercase and lowercase letters, with the exception that the uppercase letter "O", uppercase letter "I", lowercase letter "l", and the number "0" are never used to prevent visual ambiguity.



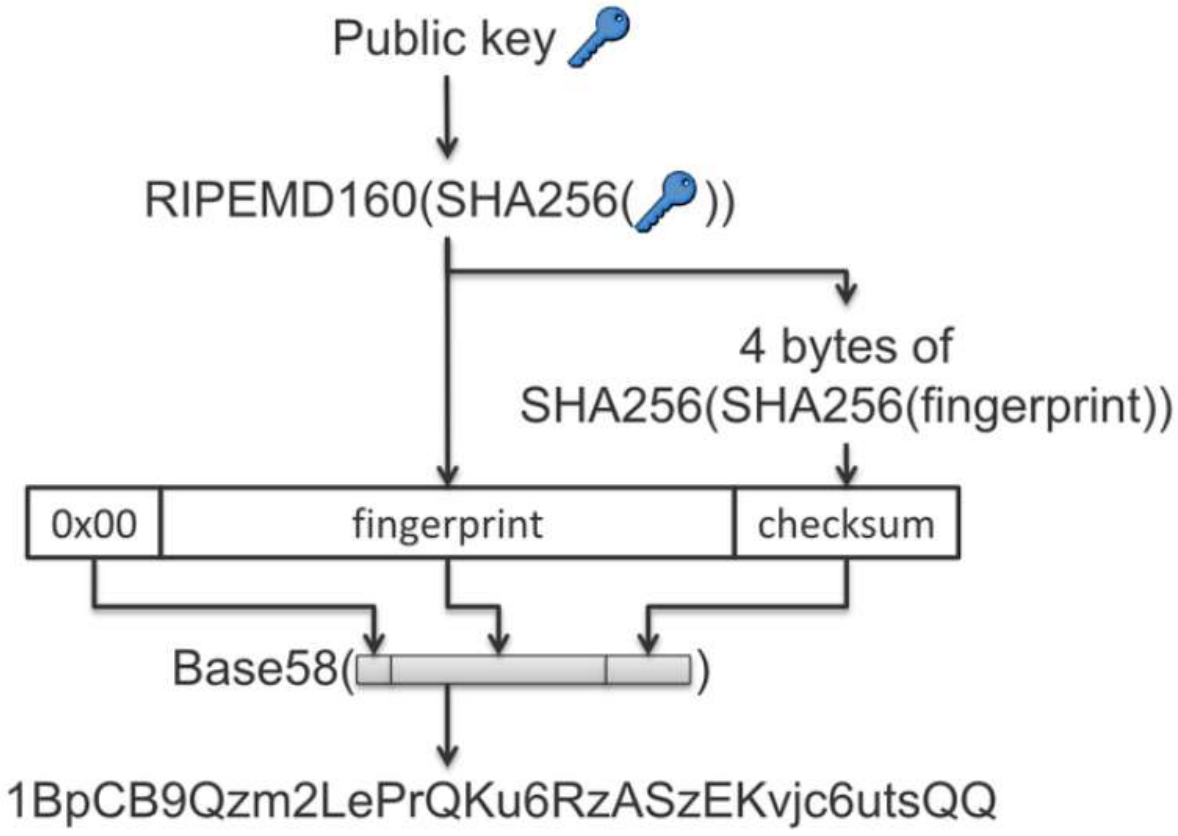
Some Bitcoin addresses can be shorter than 34 characters (as few as 26) and still be valid. A significant percentage of Bitcoin addresses are only 33 characters, and some bitcoin address length may be even shorter.

Every Bitcoin address stands for a number. These shorter addresses are valid simply because they stand for numbers that happen to start with zeroes, and when the zeroes are omitted, the encoded address gets shorter.

Several of the characters inside a Bitcoin address are used as a checksum so that typographical errors can be automatically found and rejected. The checksum also allows Bitcoin software to confirm that a 33-character (or shorter) address is in fact valid and isn't simply an address with a missing character.

Bitcoin uses an algorithm called SHA256 (also called SHA2–256) for organizing block data, for the block mining algorithm, and as part of the process for encoding transactions and user accounts (user accounts also use another hashing algorithm called RIPEMD-160)

Bitcoin uses the ECDSA elliptic curve algorithm for digital signatures.

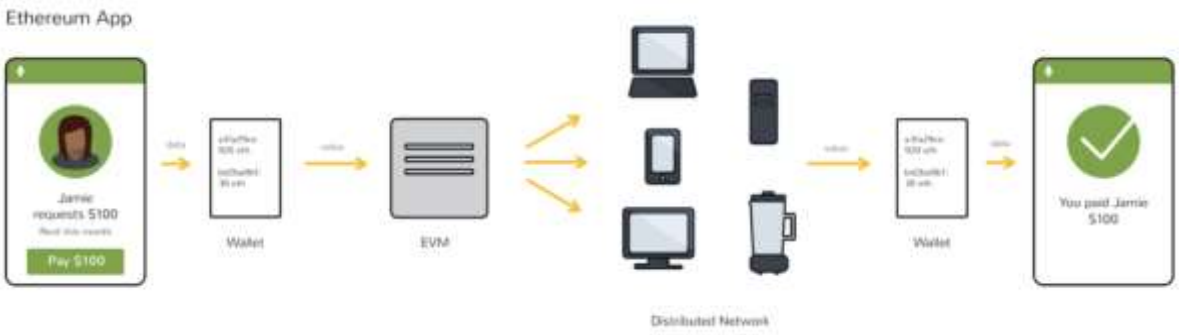


Pros	**Cons**
➢ There will only ever be 21 million Bitcoins.	➢ Bitcoin fluctuates a lot.
➢ Bitcoin is easier to liquidate than rival cryptocurrency types.	➢ Bitcoin may be replaced by a better cryptocurrency.
➢ More stores accept Bitcoin than other cryptocurrency types.	➢ People still use Bitcoin for crime.
➢ Bitcoin is the biggest cryptocurrency.	➢ No buyer protection

II. Ethereum:

Ethereum aims to use a chain of blocks to replace third parties on the Internet. Ethereum aims to function both as a kind of decentralized Internet and as a decentralized application store, supporting a new type of application (a "dapp") in the process.

Ethereum, if all goes as planned, would make the data control of these types of services to its owner and the rights of creation to its author. The idea is that an entity will no longer have control over your notes and that no one can suddenly ban the application itself, temporarily taking all your laptops offline. Only the user can make changes, not just any other entity. In theory, it combines the control that people had over their information in the past with the easily accessible information that we are used to in the digital age. Whenever you save changes, add or delete notes, each node in the network makes the change.



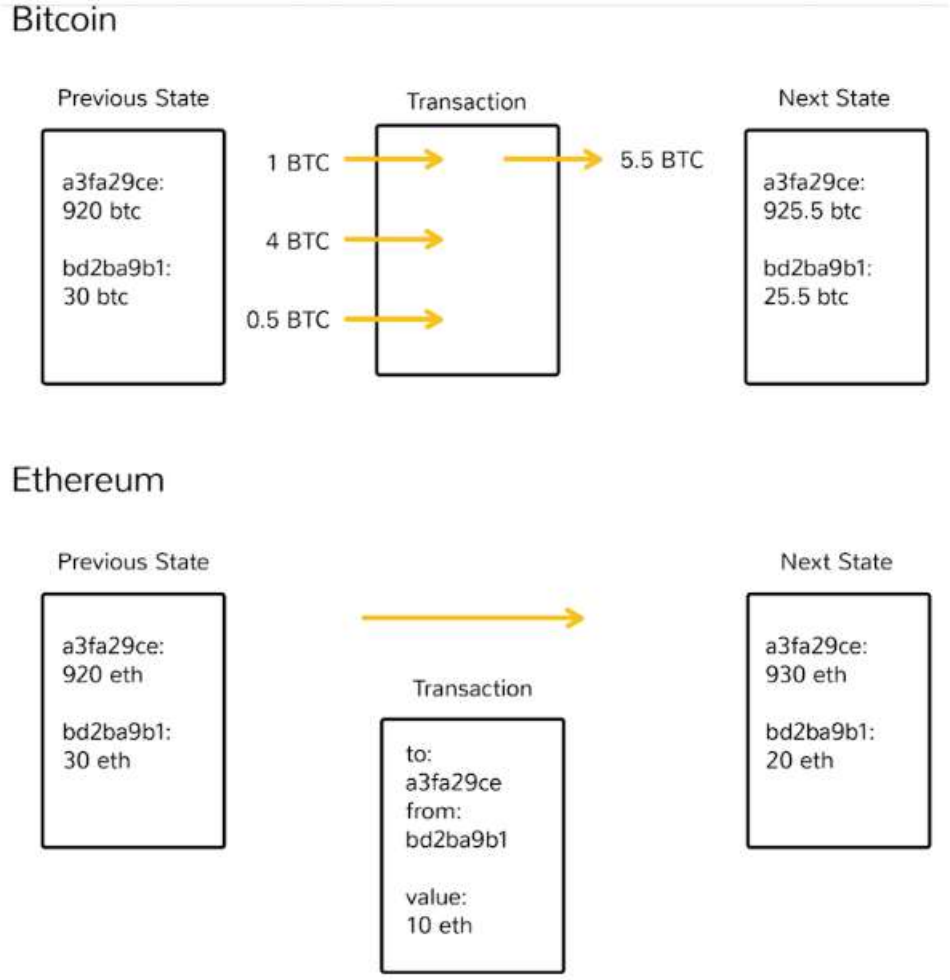
How Ethereum works:

Take the example of the notebook. Using Ethereum, the application does not require an entity to store and control its data. To do this, Ethereum strongly resembles the bitcoin protocol and its blockchain design, but adjusts it to support applications more than just a means of payment.

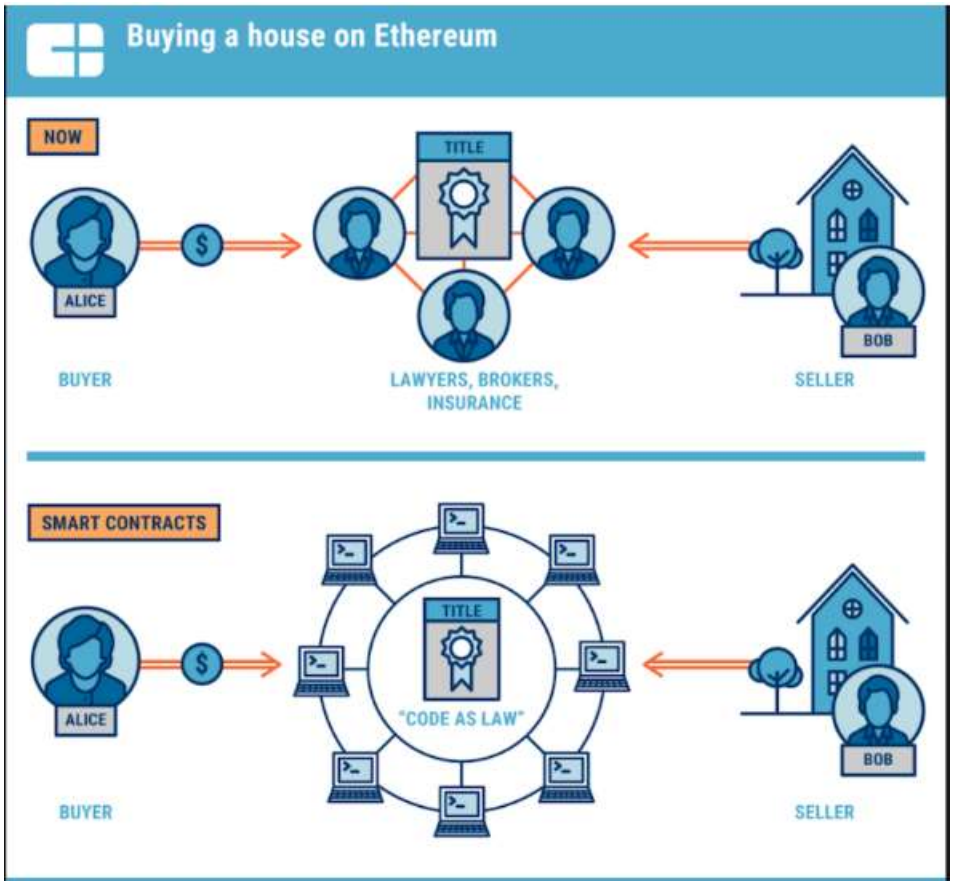
Ethereum's goal is to disregard Bitcoin's design so that developers can create applications or agreements with additional steps, new ownership rules, alternative transaction formats, or different ways to transfer.

The structure of the blockchain Ethereum is very similar to that of bitcoin, in that it is a shared record of the entire transaction history. Each network node stores a copy of this history. The big difference with the Ethereum is that its nodes store the latest state of every smart contract, in addition to all ether transactions. For each Ethereum application, the network must keep track of the status or current information of all these applications, including the balance of each user, all the code of the smart contract and where everything is stored. Bitcoin uses unspent transaction outputs to track a bitcoin.

Although it is more complex, the idea is quite simple. Whenever a bitcoin transaction is performed, the network cuts the total amount as if it is a paper currency, issuing return bitcoins in a way that makes the data behave the same way than physical parts. To make futures transactions, the bitcoin network must add up all your changes, which are classified as "spent" or "unspent". On the other hand, Ethereum uses the accounts. Like bank account funds, the ether chips are in one wallet and can be ported to another account. Funds are always a part, but do not have what you call a continuous relationship.



With ethereum, every time a program is used, a network of thousands of computers processes it. Contracts written in programming languages specific to a smart contract are compiled into 'bytecode', which can be read and executed a feature called 'ethereum virtual machine' (EVM). All nodes execute this contract using their EVMs.



Remember that each node in the network has a copy of the network's transaction history and smart contracts, in addition to tracking the current status. Whenever a user performs an action, all nodes in the network must agree that this change has occurred.

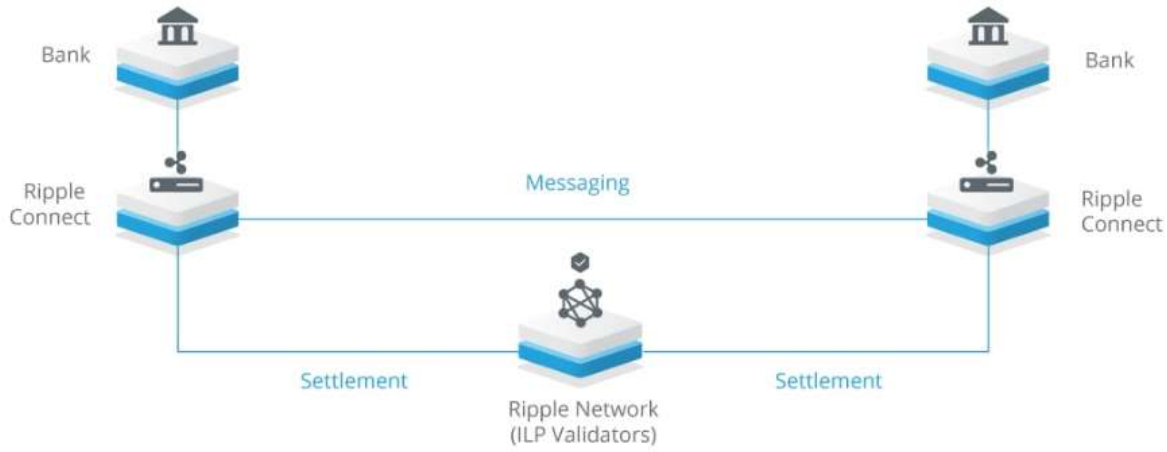
The goal here is for the network of miners and nodes to take responsibility for transferring the transfer from one state to another, rather than an authority like PayPal or a bank. Bitcoin minors validate the transfer of ownership of bitcoins from one person to another. The EVM executes a contract with all the rules initially programmed by the developer

Pros	**Cons**
➢ Users of dApps built on Ethereum will always need Ether.	➢ There are many more Ether coins than there are Bitcoins.
➢ Many new projects are being built on Ethereum.	➢ Delay In Shifting to Proof of Stake Consensus Protocol.
➢ Speed.	➢ Excessive Dependence on Buterin's Fame.

III. Ripple:

Ripple is a blockchain that is designed to be used by banks to make their payments faster. It is known as the banker's coin, and there are many partnerships with global banks currently being worked on.

Ripple focuses on their role in the "global settlement network," which simply allows financial parties, like banks, to reduce transaction costs. At the same time, Ripple also offers an improved service with direct and instant transactions. The image below shows how Ripple allows payments to cross the globe.



Ripple is primarily looking for solutions to streamline (international) transactions between banks. To do this, they have developed three platforms: xRapid, xCurrent and xVia. The XRP accompanying virtual currency can be used on the Ripple blockchain, making banking transactions even faster.

Traditional transfers are considered ineffective. Transactions can take several days and often involve high fees. If banks join one of Ripple's platforms to process transactions, they will be much more efficient.

It works pretty much like this:

Step 1: You are sending an international payment.

Step 2: Your bank buys XRP.

Step 3: Your bank sends the XRPs to the receiving bank.

Step 4: The receiving bank sells the XRPs.

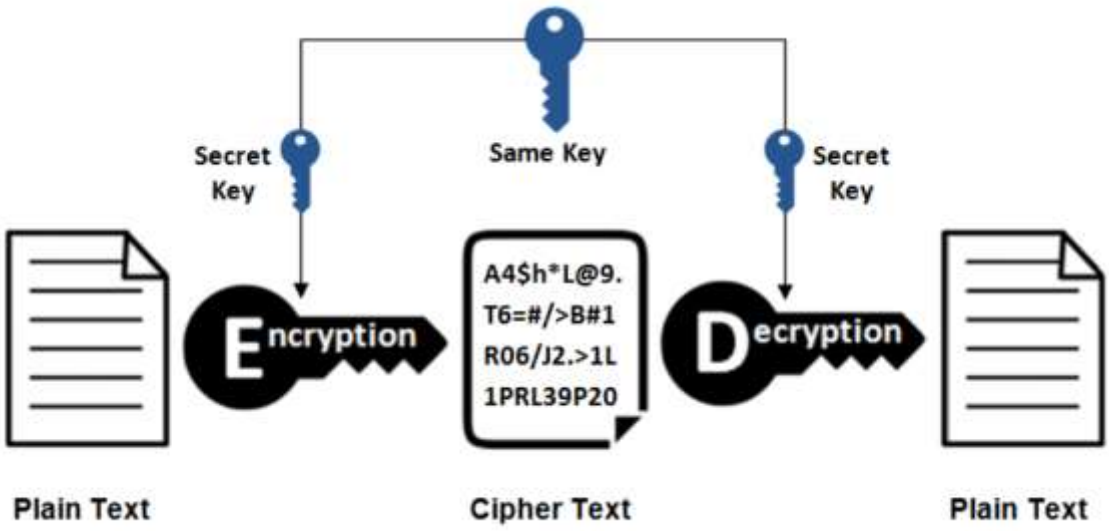
Step 5: The amount is deposited in the beneficiary's account.

2. Cryptography in cryptocurrencies.

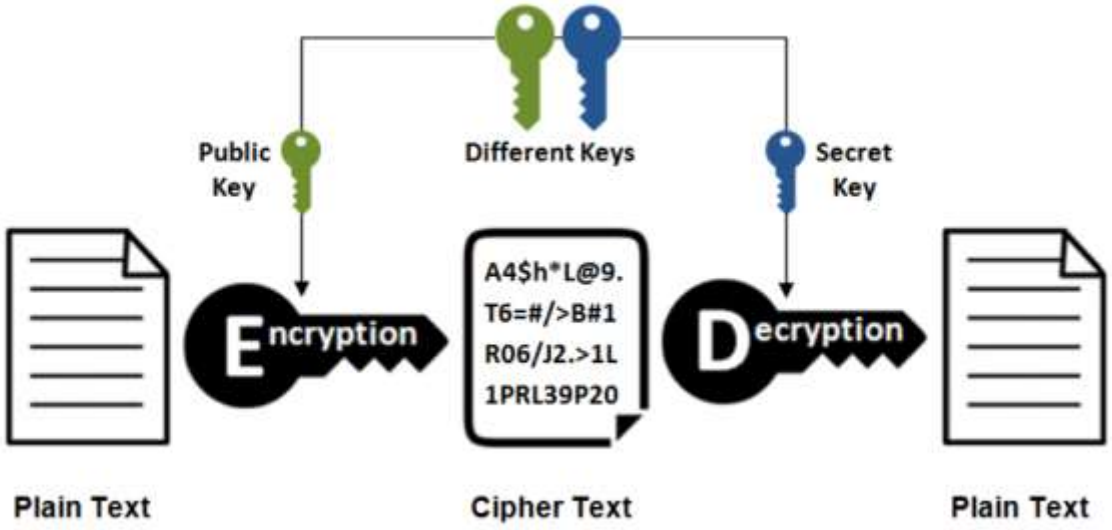
Methods used: Multiple methods exist for encryption in cryptography.

Symmetric Encryption Cryptography: It uses the same secret key to encrypt the raw message at source, transmit the encrypted message to the recipient, and then decrypt the message at the destination. A simple example is representing alphabets with numbers – say, 'A' is '01', 'B' is '02', and so on. A message like "HELLO" will be encrypted as "0805121215," and this value will be transmitted over the network to the recipient(s). Once received, the recipient will decrypt it using the same reverse methodology – '08' is 'H', '05' is 'E', and so on, to get the original message value "HELLO." Even if unauthorized parties receive the encrypted message "0805121215," it will be of no value to them unless they know the encryption methodology.

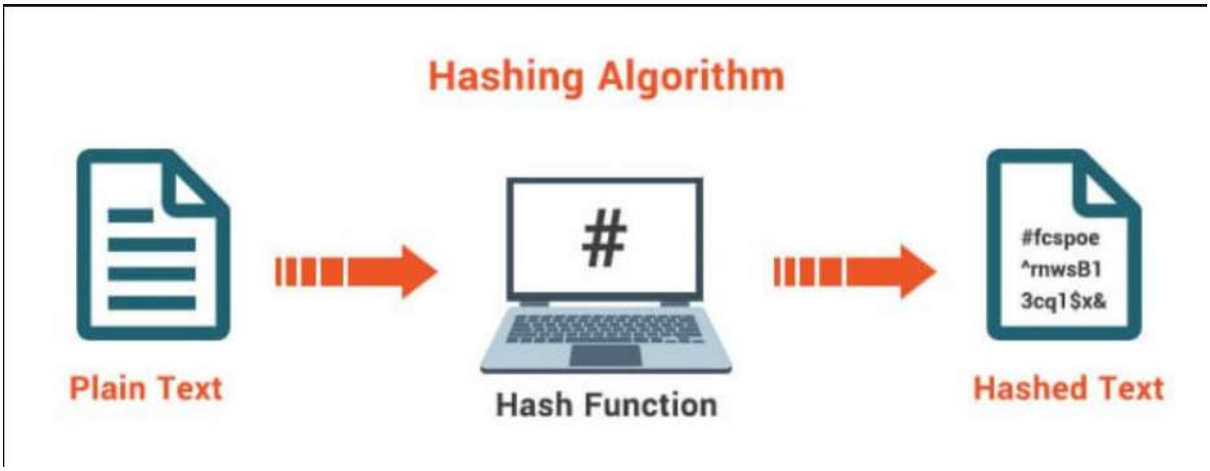
The above is one of the simplest examples of symmetric encryption, but lots of complex variations exist for enhanced security. This method offers advantages of simple implementation with minimum operational overhead, but suffers from issues of security of shared key and problems of scalability.



Asymmetric Encryption Cryptography: It uses two different keys – public and private – to encrypt and decrypt data. The public key can be disseminated openly, like the address of the fund receiver, while the private key is known only to the owner. In this method, a person can encrypt a message using the receiver's public key, but it can be decrypted only by the receiver's private key. This method helps achieve the two important functions of authentication and encryption for cryptocurrency transactions. The former is achieved as the public key verifies the paired private key for the genuine sender of the message, while the later is accomplished as only the paired private key holder can successfully decrypt the encrypted message.



Hashing: It is used to efficiently verify the integrity of data of transactions on the network. It maintains the structure of blockchain data, encodes people's account addresses, is an integral part of the process of encrypting transactions that occur between accounts, and makes block mining possible.

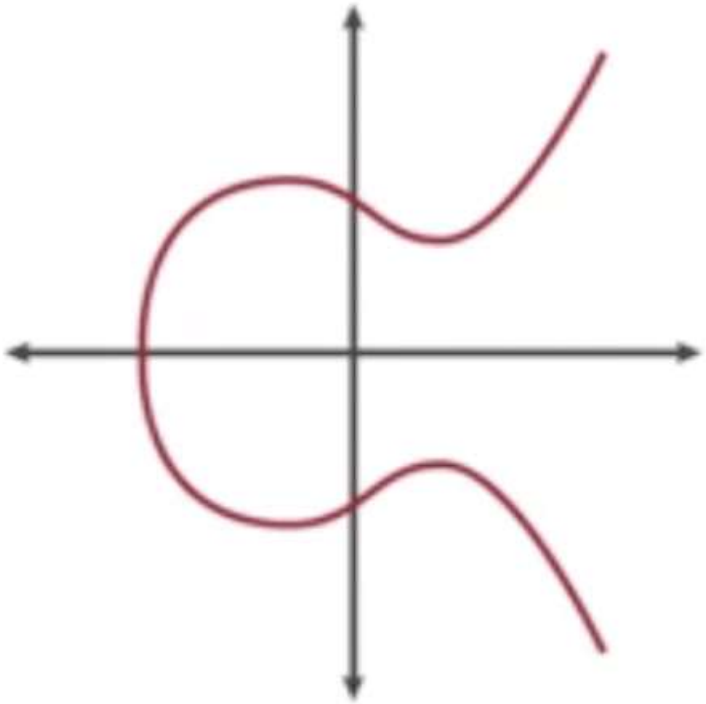


3. Elliptical curve cryptography in bitcoins, Ethereum etc.

- Elliptical curve cryptography is what is used by bitcoin, Ethereum etc. for their encryption purposes. So what is an elliptical curve? An elliptical curve is any curve that satisfies the following equation:
  Y^2 = x^3 + ax + b

  Where (x,y) is a point on the curve and a and b are constants.

  There are infinite curves that you can make. The following is how one of these curves, in general, look like:



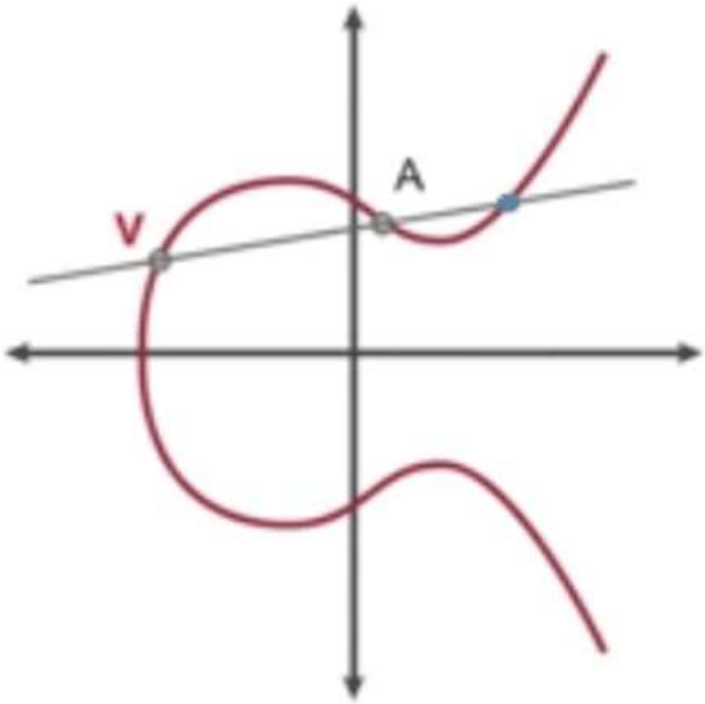
The properties of an elliptic curve?

- The curve is symmetric across the x axis.
- Any line that goes through 2 points on the curve will intersect the curve on a third point.
- Any tangent on the curve will intersect the curve on one more point.

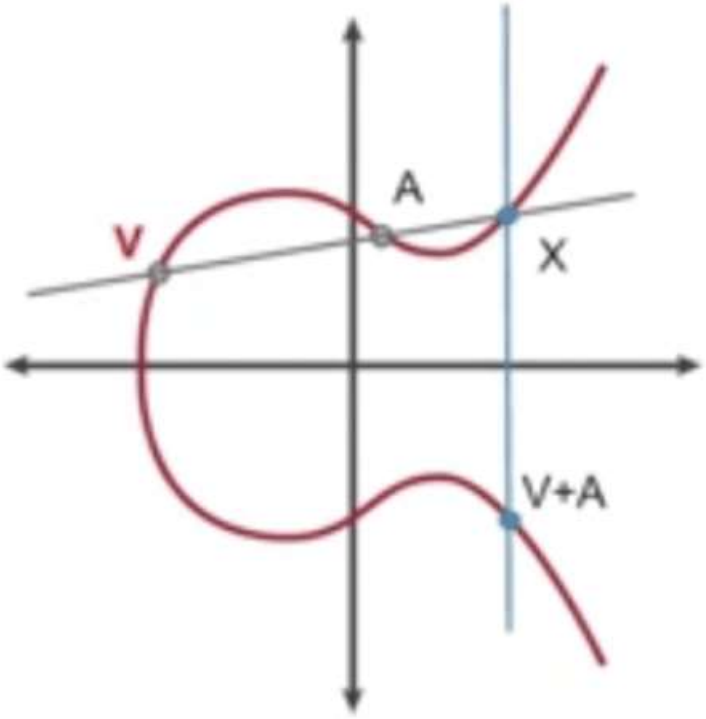
Performing maths on the curve.

Addition property of the curve

Suppose there are two points on the curve V and A. Let's trace those on the curve and put a line through them. This will intersect the curve on a third point.



We will call this third point X, and we will reflect it on the curve like this:

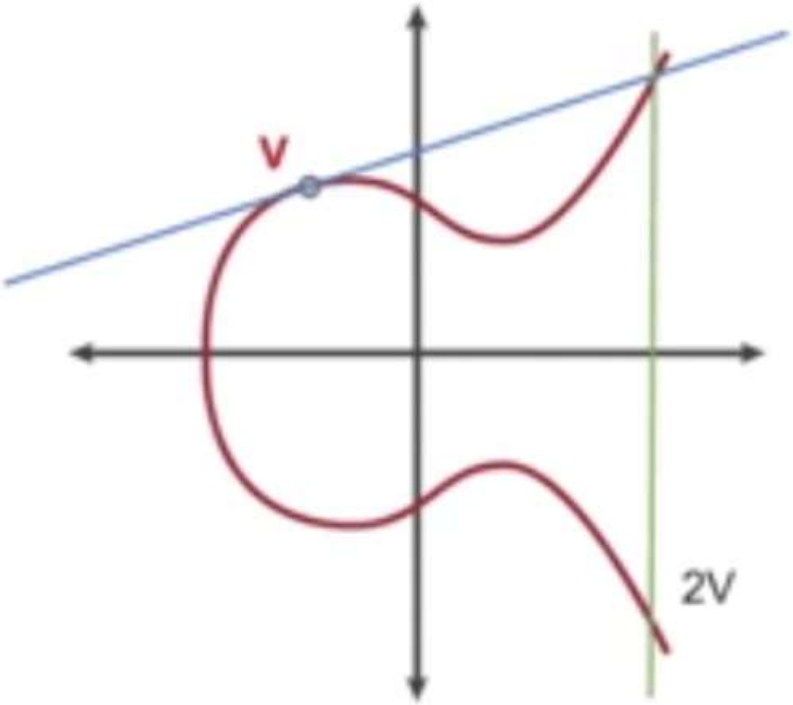


The reflection of X is a point which will incidentally be (V+A). This is the additive property of the elliptical curve.

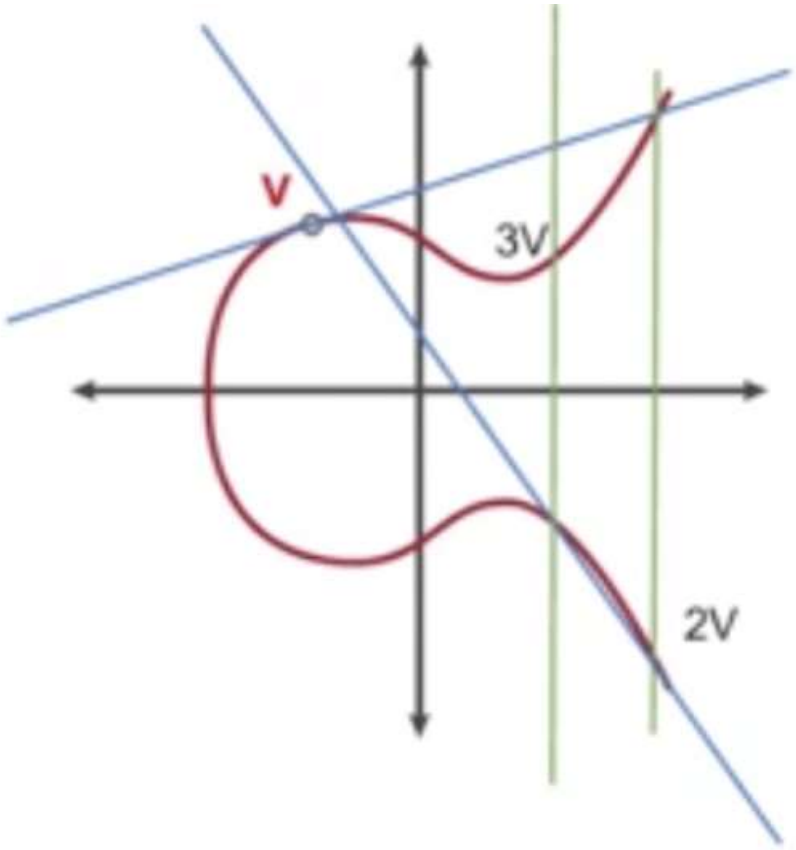
Interesting note. If we add two reflections with each other aka if we were to add X and V+A in the graph above, we will get infinity. The reason for that is that the line through X and (V+A) will intersect the curve at infinity.

Multiplication property of the curve

Now, what if we want to add a number to itself? Like suppose we have a point V, what do we do to find 2V? We will run a tangent through V and intersect it at a point in the graph and then find the reflection of the point on the curve. That reflection will be 2V.



This is also the multiplicative property of the graph because we are finding points which are basically the multiplication of an integer with the point itself. Now suppose we want to find 3V. We will join V and 2V and then reflect the point of intersection, like this:



You see how the points cycle across the graph? This is what gives it its security.

Mathematical properties of an elliptical curve

Property #1: The points on the curve form an Abelian group

The properties of the Abelian group are as follows:

- They have identity.
- The have inverses aka reflections.
- The points are associative meaning for three points A, B and C on the curve: (A+B) + C = A + (B+C).
- The points are closed on the curve.
- The points are commutative meaning for two points A and B. A+B = B+A.

Property #2: Multiplication on the curve is fast

All multiplication done on the curve can be done very fast. Now suppose we have a point P and we want to find 100P. Instead of adding the number to itself 100 times we can do the following:

- Add the point P to itself to get 2P.
- Add 2P and P to get 3P.
- Add 3P to itself to get 6P.
- Add 6P to itself to get 12P.
- Add 12P to itself to get 24P.

- Add 24P and P to get 25P.
- Add 25P to itself to get 50P.
- Add 50P to itself to get 100P.

So, instead of going through 99 steps you cut short the entire thing to just 8 steps.

Property #3: Division on the curve is slow

Whilst multiplication is fast, the division is very slow. Suppose we have Q = nP and we want to find the value of n by dividing Q by P. We can't really do that. We will have to manually go through the numbers one by one to find a value which satisfies the equation. This makes it very slow. This is called the discrete logarithmic problem and this is what gives the curves its trapdoor function i.e. it is easy to multiply n and P to get Q but given Q and P it is infeasible to get n.