

BLOCKCHAIN FOR LEGISLATORS – A GUIDE

DISRUPTIVE TECHNOLOGY – PROMOTES PROGRESS

DISTRUCTIVE TECHNOLOGY – NOT SO MUCH...

iNo!® Informed – I know; Decision – I No!



Creptoe/Beezanteen Cartoon – Rock Sale - Copyright © March 2023, L.D. Killion, All rights reserved.

Table of Contents

SUMMARY	4
BLOCKCHAIN BOOTCAMP BASICS	8
PART 1	9
SPECIAL SECTION FUNDAMENTALS OF ENCRYPTION	48
ARTICLE 1	51
WHAT IS ENCRYPTION?	51
ARTICLE 2	59
WHY IS RANDOMNESS AND UNPREDICTABILITY IMPORTANT IN CRYPTOGRAPHY?	59
ARTICLE 3	63
WHAT IS SECURE SOCKET LAYER (SSL) AND TRANSPORT LAYER SECURITY (TLS)?	63
ARTICLE 4	69
WHAT IS HYPERTEXT TRANSFER PROTOCOL (SECURE)?	69
ARTICLE 5	71
WHAT ARE CRYPTOGRAPHIC KEYS?	71
ARTICLE 6	74
WHAT IS PUBLIC KEY INFRASTRUCTURE?	74
ARTICLE 7	79
WHAT IS A HASH?	79
ARTICLE 8	84
THE BIZZARE BYZANTINE GENERAL’S PROBLEM	84
AND THE PRACTICAL BYZANTINE FAULT TOLERANCE	84
– PROOF OF WORK – PROOF OF STAKE – DELEGATED PROOF OF STAKE SOLUTIONS	84
ARTICLE 9	109
WHAT IS A DIGITAL SIGNATURE?	109
ARTICLE 10	114
GENERATING PUBLIC AND PRIVATE KEYS USING THE RSA AND ELLIPTIC CURVE ENCRYPTION ALGORITHMS	114
What is the Discrete Logarithm Problem?	116
Trap door functions	125
Generating Primes – Seeking Randomness and Unpredictability	127
Carmichael’s totient function	127

Generating the public key	128
Generating the private key (which is mathematically linked to the public key as shown below)	129
Understanding Modular Inverses	130
RSA Summary:.....	142
Elliptic Curve Cryptography (ECC)	144
Point at infinity.....	151
Point negation (Facet 3).....	151
Point addition (Facet 1).....	151
Point doubling (Facets 2 and 4)	152
Elliptic Cryptographic Curves – How Public and Private Keys Are Generated	153
Find whole prime number points on the elliptic curve,.....	155
Determine a G Generator Point and.....	155
Generating example public and private keys,.....	155
ARTICLE 11	162
WHAT IS THE INTERNET?	162
ARTICLE 12	171
WHAT IS A ROUTER AND MODEM?	171
ARTICLE 13	175
HOW WILL BLOCKCHAIN PROMOTE THE NEW WEB3?.....	175
ARTICLE 14	178
WHAT IS QUANTUM COMPUTERS?	178
ARTICLE 15	184
WHAT IS HACKING ALL ABOUT? (not a cough)	184
PART 2	196
BLOCKCHAIN CONCEPTS	196
PART 3	209
BLOCKCHAIN LEGISLATIVE AND REGULATORY MATTERS	209
APPENDIX A.....	220
COMPUTER SCIENCE	220
BIBLIOGRAPHY	233

BLOCKCHAIN FOR LEGISLATORS – A GUIDE

DISRUPTIVE TECHNOLOGY – PROMOTES PROGRESS

DISTRUCTIVE TECHNOLOGY – NOT SO MUCH...

iNo!® Informed – I know; Decision – I No!

(March 2023)

SUMMARY




The goal of this Guide is an educational document (and accompanying slide presentation/video), from the eyes of an engineer/attorney...and not a computer scientist¹, describing in everyday plain English, what Blockchain technology is all about, when to use it, what is good about it, what is bad about it, and what legislators (and regulators) can do to protect the innocent from those who misuse and abuse it.

Think of this discussion as a:

- Foreign language translator – converting the Blockchain *mystery* vocabulary into everyday plain English (the writer's language of choice); or
- A decipher – decoding² the Blockchain language secret code (a cipher³) into plaintext English.

CLASH OF THE ATOMS

My consumer alert antennae **blinks red** whenever I witness celebrity athletes and actors (presumptively compensated for doing so) advertise the get rich schemes associated with Blockchain cryptocurrency and wander from their normal promotional domains associated with athletic shoes or retirement plans.

When attending a cocktail  or tail gate party, a guest may casually mention words such as *Blockchain, smart contract, consensus, Proof of Work, Byzantine General Problem, cryptocurrency, algorithm, paradigm, Bitcoin, hash* and other associated words. And a listener will mentally process the words and  perhaps even engage in conversation about them. I do the same and often give the illusion that I know what I am talking about, when in truth, I haven't a clue what is going on inside the Blockchain Blackbox⁴. The use of Blockchain buzz words is a cool thing to do to give at least the impression 

¹ Occasionally I'll drop in a footnote that provides additional background information. While not critical to understanding Blockchain technology, it hopefully will supplement the discussion with some non-black box mystery insights. Admittedly, since I am not a computer scientist, I probably in some instances over-simplified the concepts discussed, but my defense (or excuse) is that I have the advantage of not being shrouded with computer scientist fraternity indoctrination and free to interpret in my simple language what I have studied and what I think I know and understand (*I may not always be right, but I am never wrong*, sort of thing). I tend to leave non-understandable mysteries isolated to areas such as magnetism, gravity, infinity, and subatomic quarks.

² Think of Ralphie's (from *A Christmas Story*) secret decoder token received in the mail (pocket watch looking device with a spinning disc top that when spinned, matches numbers with letters to decode a mystery number code into plaintext English letters and words), used to decode a secret message transmitted over Ralphie's favorite radio programme and instructed, to "*drink more Ovaltine*", a disappointing decoded advertisement.

³ Cipher - a secret or disguised way of writing; a code (see *National Treasure*, with Nicholas Cage, all about ciphers).

⁴ Those smart folk who programme the Blockchain Blackbox – I envy that intellectual capacity - who develop Blockchain computer programs (oddly enough called Developers) are a special breed.

of enlightenment and being savvy with today's technology. I am not criticizing that fantasy as it is part of being social.

But it is critical for the listener to know more than just how to pronounce the buzz words, when the listener is responsible for legislative and regulatory governance obligations and accountability to protect the innocent public against the bad folk⁵ (insiders as well as outsiders) who misuse or abuse Blockchain technology, especially when such bad acting extends beyond the boundary of *caveat emptor*, buyer beware - a consumer's otherwise personal risk assumption obligation.

My mission is for this Guide to be a useful go-to reference and to minimize the complications and black box mysteries surrounding Blockchain technology for the purpose of legislators and their staff being better informed what Blockchain technology is all about in order to progress relevant legislation and regulation⁶ to protect the public.

My explanations have hopefully erred on the side of elegant simplicity and not intellectual complexity. Even-so, I am affected by Einstein's view: "*keep things simple but not too simple*".

With knowledge, one has power as well as a little bit of sunshine is a great disinfectant.

This Guide is divided into three parts:

- **Part 1: What is Blockchain...the Good, the Bad, the Ugly...in simple terms.**
- **Part 2: Easy to read bite-size chunk discussion of important Blockchain concepts, which are banted about in the literature, and when better understood, helps immensely with understanding Blockchain technology⁷.**
- **Part 3: What legislators and regulators may wish to have on their legislative or regulatory to-do bucket lists regarding the reduction or elimination of adverse impacts on society (investors and consumers) caused by the destructive misuse or abuse of Blockchain / Cryptocurrency technology.**

⁵ My definition of "bad folk" is someone who does not respect the rights or property of others and takes unauthorized advantage of a circumstance to adversely interfere with those rights are wrongfully take someone's property.

⁶ For the avoidance of doubt, don't get me wrong...I am not a big fan of big government laws and regulations, but accept the fact there are instances when such are necessary components of governing society because of the cunning ability of bad folk preying on innocent consumers who do not have a practical alternative of self-defense, especially when the mysteries behind black-box technology can be manipulated by insightful and crafty bad folk.

⁷ My experience is that many authors writing about Blockchain casually use non-defined technical terms and abbreviations (and if defined, the definition is itself incomprehensible at least in plain English) in their discussion, the author assuming the naive reader has the background understanding associated with the terms. I hopefully have avoided that pitfall in this guide...at least that's my aspiration. I have found when explaining a term, it is useful to do two things: (1) be a little repetitive with explaining a terms meaning – while helpful, the downside is that this adds more pages to the Guide and (2) provide an example and picture stories which for me, accelerates the learning curve, hence there are many examples (and yes more page volume). The ole' show me vs tell me experiential learning method - experiencing something is more impactful than just listening – the ah ha! moment... Eureka...I have found it! (Lawyers take the position that the inclusion of examples in documents are not necessary if the written words are crafted accurately. As a lawyer AND engineer, I have not conceded to that message, at least for me, an example picture is worth a thousand crafted words, and would probably reduce contract interpretation litigation – which I figure is not a bad thing).

- Some have selfishly argued that consumer and investor protection public policy regulations (affecting cryptocurrency for example) have the downside of adversely affecting many parties as a result of such regulations setting up barriers of entry into a business practice, while those that have the resources (big banks/security firms?) and can manage the restrictions and regulations, enjoy an unfair privileged inferred monopolistic position. While in a pure idealistic intellectual theoretical sense that view has a symbolic ring tone of self-justification (and candidly that argument could apply to any regulatory environment), the more honest argument is that if a business practice (example cryptocurrency) is exempted from consumer and investor protection regulatory practice, would have the absolute result of emphatically and discriminatorily disadvantaging those (big banks/security firms?) who have the contrary obligation to comply with relevant regulations. Requiring one set of business entities to comply with honorable public policy regulations and another exempt is contrary to notions of justice for ALL, fair play, and non-discriminatory objectives.

Supplemental reference material includes:

- Appendix A, a concise layman's guide of what computer science⁸ is all about. This is important because Blockchain technology does not function without computers and the internet. Appendix A can be skipped or scanned if the reader is primarily interested in just a better understanding of Blockchain technology, discussed in Parts 1, 2 and 3. However, the reader may wish to review at least the section in Appendix A on what **Modem's** do and how they work, since we all have those little black boxes in our homes and offices. Without them, our communication with the internet would all but stop.
- Slide and video presentations of the Guide's messages (a summary guided tour of the Guide).

CAUTION! THERE IS A LITTLE MATH IN THE GUIDE...HOPEFULLY I HAVE TAKEN THE STING OUT OF SUCH BY CRAFTING STEP BY STEP EXPLANATIONS AND LOTS OF ILLUSTRATIONS – THE PICTURE TELLS A 1000 WORD SORT OF THING.

For The Avoidance of Doubt ... And An Up-front Additional Cautionary Message - ONE SIZE DOES NOT FIT ALL!...

Blockchain technology is much more than just using it for investing in online digital convertible virtual currency (aka cryptocurrency or crypto - Blockchain's first use), such as Bitcoin (which is just one of many brands of 'cryptocurrency'). The technology can be useful in many many other non-finance related applications and transactions.

Broader Use of Blockchain Applications

Example potential non-cryptocurrency applications of Blockchain include:

- **Financial sector (fintech)**

⁸ I suspect I have shoes older than most of today's computer experts (IT -information technologists, computer scientists and engineers) and I was writing Fortran code (that should date me) computer programming, early day U.S. Government thermodynamic engineering applications, long before many of those experts were born.

- Efficient, low cost, auditable, secure exchange of value (currency, stocks, bonds, etc) between parties in a private environment; payroll payments in cryptocurrency; cryptocurrency investment
- Can apply to business or personal transactions; an alternate to traditional banking
- **Real Estate**
 - Public record keeping of real estate transactions (history, proof of title, recordation)
 - Free up dead capital by making non-financeable land, financeable if title to the poor is confirmed
 - Finance document management
- **Insurance**
 - Managing claim history; micro-insurance (target lower income insureds; insuring process takes place online)
- **Government**
 - Smart cities initiative (monitoring traffic and air quality; road management)
 - Development sandboxes that permit research and testing in a secure and safe environment, not affected by restrictive regulatory or legal rules.
- **Other**
 - e-residency (electronic identification residence cards), optimize public records, internet-of-things data analysis, internet trust layer, spam-free email, owning one's identity, trusted authorship (rating agencies, weather outlets), intellectual property protection

However, there are many applications where Blockchain Technology is **NOT** a practical solution to a problem...for example...

- (1) When transactions need to take place within one organization, not involving any external stakeholders, a (private) decentralized Blockchain network is not a practical solution. Trust within an organization can be achieved through different means. Applying a Blockchain solution would be like using a jack hammer to install a picture hanger nail;
- (2) When transactions between two or more parties have to be highly customized and are constantly changing. In that case, creating a Blockchain 'smart contract' ('auto-pilot' computer automatically electronically executed contracts in contrast to a normal contract requiring non-electronic human intervention performance execution by the contracting parties) for every possible transaction becomes too much of a hassle. As a result, a Blockchain solution would not be advisable. For example, use of such in a research laboratory environment is too much of a volatile business model to apply Blockchain;
- (3) If a business needs to process millions of transactions per second, Blockchain does not, yet, work, because it is too slow. Existing relational database solutions have proven themselves to be capable of running trillions of queries in a very short timeframe (online use of credit cards for example).

It is important to have a clear understanding of how Blockchain technology could help solve business or personal transaction objectives. Especially, since the technology is still so new, it is best to also know when to use Blockchain and equally, when not.

BLOCKCHAIN BOOTCAMP BASICS

Blockchain technology's first use was in the separatist's cryptocurrency (crypto's) industry – using digital coins (convertible virtual currency) as a form of currency for use in financing projects or investment speculation (the ole' buy low, sale high wealth creation formula).

At crypto's base are two founding father trustless Public Blockchain networks, Bitcoin and Ethereum, which are:

- Decentralized and trust no one networks,
- running on the internet (no middleman manager such as a bank, is involved),
- which permanently records cryptocurrency trade transactions,
- executed directly between participants (a peer-to-peer structure),
- transactions authenticity is verified by competing independent Blockchain participants (referred to as Full Node validators or miner's or staker's) and a consensus vote taken to confirm that authenticity,
- who are incentivized for offering the validation services,
- by being the first to provide the Blockchain transaction and authentication verification services and
- paid for such service with the issuance of new issued ('minted') cryptocurrency 'digital coins' for doing so.

Verification involves a consensus or approval vote (a **Consensus Protocol or operating agreement**) by other verifiers that a proposed Block of transaction verification is authentic and when approved, the verified Block of transactions permanently recorded in the Blockchain. (The Bitcoin network is reported to have over 50,000 verification participants, and as later discussed, this large number of validators is fundamentally necessary to ensure security in the network and solve what is known as the Byzantine General's Problem – later discussed in detail). Individual crypto buy-sell transactions are grouped together in recorded Blocks that makes it efficient to validate and record such data in the Blockchain digital ledger.

The Blockchain network is protected from hacker attacks by cryptography (secret computer code writing) using special digital cryptographic keys required to lock (encrypt – change plaintext to unintelligible cyphertext) and unlock (decrypt – convert cyphertext back to understood plaintext) private / confidential information and data. Cryptography and network security are **BIG DEALS** in Blockchain technology, and like the internet, non-functional without it.

The Ethereum Public Blockchain platform (unlike the Bitcoin dedicated system cryptocurrency platform) validates lines of code (which is an open source code, accessible by anyone, the more common code language of which is called 'Solidity', a Turing-complete language – meaning the code is flexible and has many applications), which has made it possible for others to develop their own separate Public Blockchain networks on top of the Ethereum platform and (1) issue their own separate cryptocurrency (native tokens) or (2) build database management applications that do not use cryptocurrency – thus Blockchain technology can be much richer than just trading cryptocurrency.

To successfully attack a decentralized Public Blockchain trading cryptocurrency and gain total control, even the ability to shut it down, in addition to creating false records or falsely move cryptocurrency to an

unauthorized account, requires gaining 51% control of the computational power (Proof-of-Work) or value of tokens staked (Proof-of-Stake) to verify transactions. The so called **51% control risk**. As discussed later, the likelihood of someone gaining such 51% control is highly problematic, because of...

- (i) the way **Consensus Protocols** (the operating agreement and process to approve and add Blocks of data to the Blockchain digital ledger) are drafted, and
- (ii) the more valuable cryptocurrency becomes,
- (iii) the more electrical energy it takes to attack a Proof-of-Work chain, like Bitcoin, and
- (iv) the more money to attack a Proof-of-Stake chain, like Ethereum.

The security of these chains – as measured by the amount of energy or money someone would have to spend to attack them and gain 51% control – is in the multi-billions of dollars. It would require in effect either a sovereign nation or extraordinary rich individual to mount a successful (? , maybe) attack. As in most logical networks, if it costs you more to do something than the gains to be had, then you won't do it...unless there are mere sport or ideological reasons for performing a task disconnected from sane commercial drivers

Largest practical issue: scalability. How to make it work with large transactions and data bases.

COMMON CENTS...

There is no law that prohibits a person to attend a gambling casino and bet their life savings...and if not successful, become a burden on society. There are laws regulating the honesty of the casino and the gambling truly based on independent mathematical probabilistic episodes of winning or losing, but the gaming folks can't artificially manipulate the gaming machines to their advantage. Equally, the player is legally obligated to report their winnings – and if not, criminal punishment can follow. Rule 101 in tax law: always report the income, cause its criminal if you don't, but always take the deduction, because that is an arguable fact issue not subject to criminal rules (absent out-right fraud).

The 'right' to 'invest' in junk bonds, or stocks, or CDs, or bonds, or Cabbage Patch Dolls, or wall hanging picture plates, or gold, or chinchilla pelts, or palm oil, or red wiggler earth worms, or whatever else the crafty marketers of those 'products' can hypnotize and persuade the investing public into accepting as valuable, is readily recognized in society. Since there are many legions of investing followers—many easily hypnotized (as to the illusion of a get rich quick scheme – don't we all), others gaming the system (and often being abit more crafty and have resources to dishonestly out maneuver the less informed – not a level playing field – ok to demand a level football field...but when it comes to personal investing one has strayed from preaching to meddling) have pre-planned to be an early entrant and quick profit exit at the loss of the hypnotized legions – either as an organizer or mere benefactor of pump and dump schemes. Just like the tabloids that report on alien intruders or other miraculous events (praying on the illusionary hope emotion of humans), humanoids have a fantastic mental ability of converting illusion into desired reality. Some call that HOPE. Others call it DECEPTIVE TRADE PRACTICE.

Society has moralistically decided its not a good thing to take one's own life. Prohibiting such if one really wants to participate in that event, is impossible to prevent. (After all, life is the most precious asset we own and when a person is ok giving that up...there is no pragmatic alternate of resolve). Folks that aide and abet that event is another matter, as they still covet their life asset and will be a survivor subject to criminal penalties, and any time someone values their own life, penalties (like going to jail) restricting that desire, is always a good motivator to cause certain behaviour to follow – don't aide and abet.

Why is 'investing', or 'gambling', or 'self-inflicted' wounding of one's financial status by participating in the cryptocurrency craze any different than one's right to gamble in a casino or take their life in the confines of their own home office or invest in junk bonds? Especially when any 'winnings' are reported as taxable income? Some call this a victimless environment. But is it? Consumer and investor protection laws are written to protect not only the innocent, but the misinformed or ill-informed.

It is interesting to note that Crypto is a noted investment option that has popped up on leading financial investment houses. Why? Only one reason... The traffic (aka daily trading – buying and selling crypto) in that environment is so heavy, there are investment fee monies to be made in offering that option. Ah Greed! one of humanoids innate drivers, just like compulsory eating when hungry. A not so casual analysis of that circumstance, which BTY, the crypto folks do not complain about since churning the buying and selling crypto in any medium is an ok thing to do, the separatist Blockchain cryptocurrency folks, shout the joys of decentralization and revolution against Big Brother Banks and their centralized control, yet covet the fact that centralized regulated investment houses promote the crypto craze. This decentralized – centralized symbiotic⁹ relationship is not unlike the symbiotic relationship between unregulated decentralized crypto being valued in centralized regulated fiat¹⁰ money. One can't exist without the other. Sorta sounds like what parasites do.

At the end of the show (investing in whatever)...did we enjoy it or not? Did we feel like the ticket price (investment) was worth the entertainment? How much caveat emptor – buyer beware and absorb the consequences of one's gain or loss – is the right societal remedy? When do honorable regulators step in to protect the innocent...or even not so innocent...from those that take unfair advantage of a circumstance and not just use stuff...but misuse or abuse it to the unfair (often times deceptive) detriment of those that were bilked by the events? There is a sucker born every minute so claim circus barkers. Honesty in some regimes is looked upon as how one survives to the detriment of others and that's an ok thing to do. That's the way it works in the jungle, so why not in the conference rooms or advertising billboards?

All those in favor of not wanting to be cheated¹¹ by another, raise your hand? I figure those without raising their hand either are deaf, have selected hearing or something else in mind.

Bottom line? The crypto craze is not going away. Folks will always depend on greed and seek a quick rich path to success...problem is once there (if one is lucky enough to have scaled that peak)...there is a new greed mountain peak to overcome...some call that infinity. The challenge is to find a happy medium to protect the climbing innocent from misuse or abuse circumstances...and that's what our elected regulators and legislators are commissioned to do on our behalf, since the individual investor and consumer are but a force of one against many, and it takes a centralized regulated government environment to compete against decentralized unregulated separatists programmes

⁹ Characterized by, living in, or being a close physical association (as in mutualism or commensalism) between two or more dissimilar organisms.

¹⁰ Regulated money issued by a government.

¹¹ Taking unfair advantage to the personal gain of the one who cheated and the detrimental loss to the one that was cheated. Loss in personal rights or property.

PART 1

THE BLOCKCHAIN

The Good Ole' Days

In the good old days, life was simple and most tasks and transactions were accomplished using DIY¹².

- An abacus used for counting,
- A sextant to look at the stars and navigate the seas,
- A shovel to dig a hole,
- A hand-shake sealed a deal,
- An animal pelt traded for a basket of berries was the currency of the day,
- Transportation provided by bridled horses (or at least manually driven cars),
- Messages sent by pony express,
- Tasks accomplished with human power or horsepower.

Folks dealt with one another face-to-face.

For example, selling land was accomplished by the seller physically handing a handful of dirt scooped from the soil and placed into the hand of the buyer and the deal was done. (This worked so long as land was plentiful and sellers/buyers few in number). Honesty and trust in those processes were self-evident – you could immediately see, touch and hear it. Security that a task would happen as intended was provided by experiencing and witnessing first-hand things were going the way they were intended, else personal involvement in the correction made to get things back on track. Privacy was established between the parties by silent gentleman's agreement and neither publicizing their actions. It was easy to spot a misfit, someone up to no good (they always wore a black hat – named '*Black Bart*') - and the common sense instinct to stay clear of them. If something broke, you fixed it. Few mysteries involved (except for the Alchemist dream of converting lead into gold).

The Modern Era

Then came the modern age (accompanied with many mysteries) when electrons were captured, controlled and instructed to accomplish tasks on societies behalf (especially through the internet), making life supposedly easier...

- Computers do calculations,
- Global Positioning Systems guide ships,
- Robots drill holes,
- A contract/lawyer/court/ deed recording system seals deals,
- Online internet electronic banking used to exchange value or pay bills,

¹² DIY = Do It Yourself

- Driverless cars that free up the riders to do more important things,
- Messages sent at the click of a button and instantaneously received around the world by email, text or social media, and
- Tasks accomplished by tireless electrons, the Internet of Things (IoT) revolution, where home thermostats and lights are operated by the touch of one's smartphone handheld devices, even if on vacation in Antarctica, those autonomous performing smart contracts putting iRobot to shame.

The BIG transition or change in the fundamental way of doing things, from the Good Old Days to the Modern Era, referred to as a **paradigm** change, disrupted (upset) the good old days processes, but for the good.

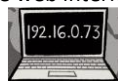
The Modern Era brought forth large centralized organizations who mastered the skill to control electrons – the so-called *middleman* (such as banks, stock exchanges, credit card companies, cities and trash collection, health care networks, online e-commerce networks – think Amazon, or centralized internet servers – think Google) which flourished and **disrupted** the Good Old Days era simple life whereby electrons are commanded to provide online internet/web services, order and pay for goods and accomplish tasks for society's benefit (no more DIY or face-to-face encounters). If something breaks, it is not repaired, but replaced with an updated duplicate.

This centralized modern era network is illustrated by the following simple internet online bank transaction example.

- Client A has a bank account at Bank A. Client B has a bank account at Bank B.
- Client A sits at home in her study and signs into the internet from her personal computer which is connected wirelessly to her modem/router¹³.

¹³ The router/modem ...

- Wirelessly (or by cable connection) communicates between Client A's computer (which is identified with a unique one-of-a-kind Internet Protocol – IP – address number (like a fingerprint) – not unlike one's home address, a number address used to identify the location of the home or the computer) and her modem, and the modem communicates between Client A's computer and the Internet Service Provider - ISP (such as Comcast or AT&T – a company providing a service of connecting computers to the world wide web internet).



(IP address example)

- The modem converts digital 'fixed or discrete or lumpy' electronic signals from the computer (a mixture of "0s" or "1s" in computer binary code, sort of like an on and off switch, the language the computer understands) and



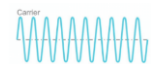
(example discrete digital electronic signal



- converts that 'lumpy' digital language to a continuously 'variable' analog signal language understood by the internet ('non-fixed' or variable sine wave like oscillating signals that can vary continuously, sort of like gradually turning up the volume on a speaker instead of clicking between fixed discrete individual volume levels),

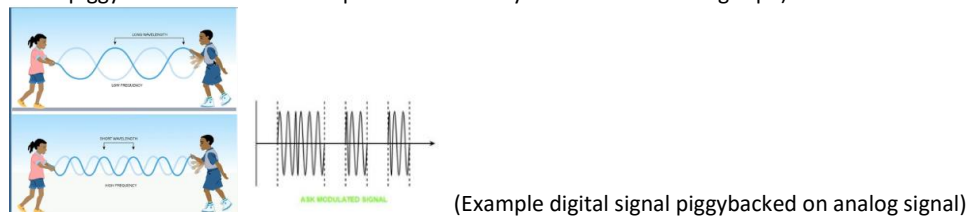


(example analog sine wave signal

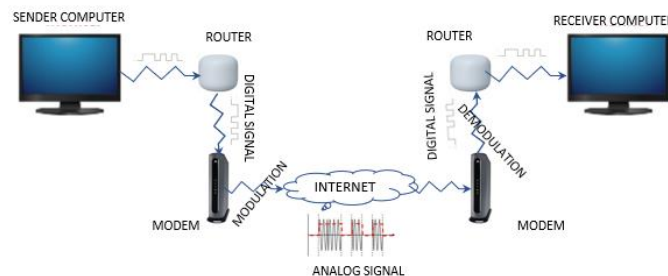


- Client A enters her ID and password (or logs on with her fingerprint or facial recognition, so called biometric passwords) through her keyboard and follows the instructions on her computer monitor, which unlocks the computer. Client A then searches the web using an internet search engine such as Google and locates the website of Bank A.
- Client A signs on to her online Bank A bank account by entering her bank account ID and password into Bank A 's website login page, and because Client A has setup two factor authentication, Bank A website login process causes a text message to be sent from Bank A to Client A's mobile phone citing a secret logon code number. Client A enters the secret code into her Bank A website sign on page which then gives Client A access to the details of Client's A bank digital ledger (bank account).
- The online bank account (ledger) detail, among other useful information, shows Client A's deposits and withdrawals, by name, date, amount and current balance in the checking account. Bank A's online digital (non-paper, electronic) account ledger, viewed from Client A's computer monitor, displays Client A's banking activities and is Client A's bank account 'ledger', but instead of bank transactions recorded by hand in a real live paper ledger book, the electronic ledger book is a digital (electronic) substitute for a hard paper copy accounting book ledger.
- Client A sets up in her online Bank A account ledger, Client B's name as a recognized party Client A knows and will do business with and inputs into Client A's bank account ledger, Client B's bank account details (account number and routing number, name, address, phone).

-
- and the converted or modulated digital signal piggy-backed onto the internet service provider electronic analog normally rhythmic sine wave signals and such conversion/piggyback process referred to as **modulation** (sort of like two people holding a jump rope, wobbling it up and down in a normal rhythmic sine wave pattern and one of the wobblers adding an extra piggyback wobble that disrupts the normal rhythm of the oscillating rope).



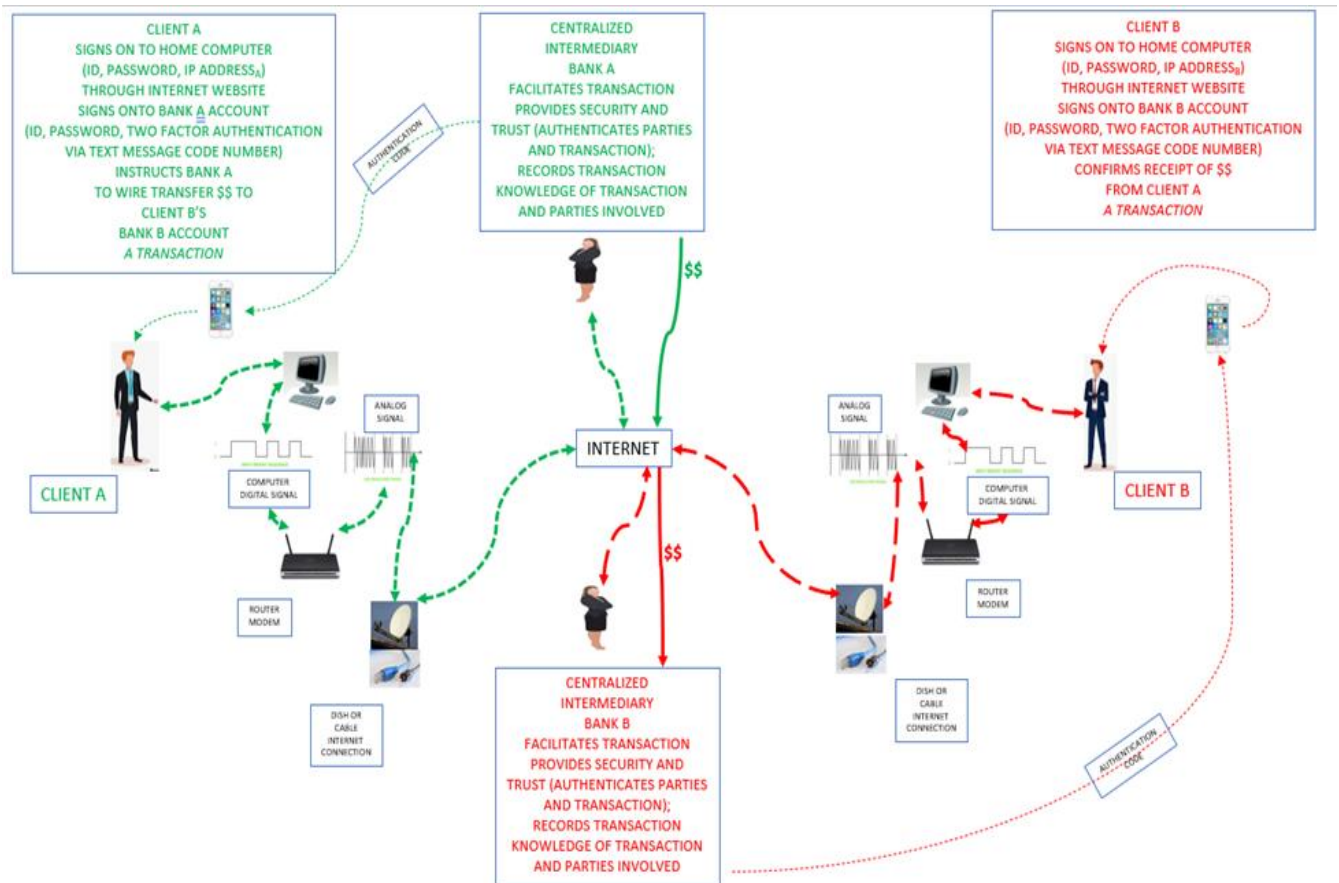
- The disrupted piggyback burdened analog signals are transported through cables or dish networks over the internet and routed to a destination computer IP (Internet Protocol) address (a one of a kind unique identification number or address that identifies the location and owner of sent and received internet communications), and the disrupted piggyback analog signal converted by the receiver's modem back to the original digital signal and such deconversion process is defined as demodulation (the language of the receiving computer).



- The sender and receiver Internet Protocol (IP) addresses are used to link the sender with the receiver...sort of like mailing a letter from your home address (and the envelope containing the senders return address) and the envelope addressed to the receiving party. The two addresses are needed to link the communication and determine the destination objectives of the electronic signals. Some of the mystery taken out of modems and routers...

- Client A sends an online message through Client’s A bank account website, to Bank A and instructs Bank A to wire transfer a certain amount of funds on a certain date from Client A’s ledger account (a debit) to Client B’s Bank B account ledger (a credit).
- Bank A, confirms the authenticity of Client A , the wire transfer amount, that Client A has sufficient funds in its checking account and also confirms the authenticity of Client B, that it has an account with Bank B and is set up as a recognized party in Client A’s account ledger. After that confirming service, Bank A electronically transfers over the internet (linking Client A’s IP address through Bank A’s IP address and then linked to Client B IP address through Bank B’s IP address) the funds from Client A’s account (the debit) to Client B’s account (a credit), and does so in a secure manner, designed to prevent third parties from interfering with the transfer or worse yet, such third parties diverting the transferred funds to an unauthorized address.
- Client A and Client B can log onto their respective online internet Bank web accounts, and confirm the transaction, Client A’s account will show a withdrawal (debit) of the sent funds, and Client B’s account will show an addition (credit) of the transferred funds.

This centralized banking transaction is illustrated in the following diagram...



Modern era centralized bank internet web communication *efficiencies* come at a cost. The price paid for these centralized 'luxuries' includes:

- Systems are much more complex and mysterious (at least to the non-expert),
- Communication accomplished by illuminated/speakered TV like screens (computer monitor), disbursed on, under or above spaceship earth – making development of human interactive social skills a challenge,
- Systems built around black boxes that users have not a clue what is inside or how it works (no more DIY fixes for faults),
- Trust is displaced into the hands of a centralized authority (and their consulting experts) to confirm authenticity of a transaction and parties involved, ensure transactions are accomplished on time (though at times slowly) and the black box functioning for user's benefit, along with compliance with relevant regulatory obligations and an administration fee charged to users for compensation of having provided the centralized services,
- Security is entrusted to experts who craft cryptographic¹⁴ processes to create secret, secure messaging so that unauthorized prying eyes are kept in the dark,
- Face-to-face ('*peer-to-peer*') independence is given up and exchanged for interfacing with computer terminals and centralized middlemen authorities;
- Privacy and anonymity is compromised (recall how much personal information is disclosed when filling out online applications to receive a centralized service, or banks having knowledge of persons identification and details of transactions between the parties),
- Complex centralized computerized systems at risk of being attacked (*hacked*) by unauthorized electrons sneaking into backdoors built by misfits¹⁵ (attackers or hackers or cyberterrorists – hiding in the dark web using pseudonyms, who may or may not be named Black Bart or wearing a black hat – maybe a black hoodie) to impose malicious acts on the internet network causing harm to honest users;
 - The devil is no longer easily recognized wearing a red cape, carrying a trident and brandishing horns or wearing a black hat). Instead, devils are hiding in the dark web, behind computer terminals and conducting electronic theft, no longer requiring a mask, get away car and driver, exchanged for stealth by night, as 24/7 anytime electronic hacking theft can happen in plain sight.
- Centralized systems, in the catastrophe scenario, are exposed to a single point of failure risk. Since a centralized system, a single successful attack could knock out the entire single system.

¹⁴ Cryptography, or cryptology (from Ancient Greek: "hidden, secret"), is the practice and study of techniques for secure communication in the presence of adversarial (misfit) behavior. More generally, cryptography is about constructing and analyzing system operating plans that prevent third parties or the public from reading private messages. Secret code messaging...think Ralphie and his secret decoding disc.

¹⁵ Misfit = someone, without permission, who illegally takes (steals) something (they see as a benefit to themselves) from someone else who legally owns it, and the means used by the misfit involves malicious intent. Means of taking could involve use of a club, torture, trickery, fraud, computer virus, ransomware, espionage, bribery, cyber attack (disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information). I've been told it is not a good thing to have a nation's military defense computer system disabled by a misfit.

The Blockchain Era

The disruptive¹⁶ paradigm's caused from changing from a DIY Good Old Days simple life to Modern Era electron controlled decentralization economy, continues with the Blockchain era ...

- First the change from the simple Good Ole Days Era decentralized DIY life -
- To the centralized Modern Era complex internet electronic age where a user pays expensive dealer prices to replace faulty widgets instead of inexpensive DIY repairs –
- And now *reverting* to the Good Old Days decentralized structure, prompted by a fundamental paradigm change caused by Blockchain technology, where the Blockchain network provides the DIY and trust effort.

Blockchain technology started out as a disruptive separatist's idea regarding creation of a standalone unregulated currency platform, for financing or investing with convertible virtual currency or cryptocurrency¹⁷ or digital paperless symbolic 'coins', through the internet— ***in a decentralized (by cutting out middlemen such as banks), trustless peer-to-peer less costly more efficient decentralized ledger environment, using the internet, in contrast to reliance on recognizing currency value only in government regulated 'fiat'¹⁸ paper money, such as the US Dollar or British Pound. [THESE MYSERIOUS TERMS ARE EXPLAINED IN PLAINTEXT ENGLISH BELOW].***

Digital coin (paperless convertible virtual currency or cryptocurrency) is a form of trading value¹⁹ without going through a centralized middleman such as a bank, over the internet, protected by cryptography (secret coding designed to keep prying eyes in the dark, think Spectre attempting to decode secret messages developed by Her Majesty's MI6 and protected by James Bond). Digital currency allows

¹⁶ Disruption is not necessarily a bad thing; Doing things fundamentally different or a paradigm change which means a BIG fundamental change – more than just tweaking, but bold and new events, can result in improved efficiency and cost saving. ***The challenge is to recognize if a big change, a paradigm shift - is beneficially disruptive or revolutionarily destructive?***

¹⁷ The administration of cryptocurrency is not managed by a central authority such as a bank, but managed in a decentralized manner where each user computer connected to the Blockchain network (where each connected computer is called a 'node'), acts like a self-contained mini-bank, and has access to a copy of all [bank] recorded transactions or 'ledger' (accounting documents that records all transactions).

¹⁸ Fiat money means money printed and controlled exclusively by a government and a central bank charged with administering such currency (regulating how much is in circulation, etc.) and the value of such fiat money established by the full faith and credit recognized by society in the government standing behind the fiat money. An example: the public recognizes more faith and value in the US Dollar as compared to the North Korean Won. What do you want in your wallet? Fiat money is the currency of a sovereign nation that enforces strict rules and regulations how such currency is controlled and severe penalties (criminal and civil) if a person prints, uses or distributes counterfeit (fake) fiat money.

¹⁹ ***Value is in the eye of the beholder.*** Us human's use money and now cryptocurrency as a means to recognize value... as its easier to carry in our wallets (real leather hip-pocket kind or online internet electronic kind) and exchange for goods and services or other things viewed as having value. Carrying around sea shells or animal pelts, as recognized currency, and traded for desired things, can get messy and cumbersome...so by recognizing value in an easily carried or electronically recorded currency has its advantages. Its interesting that us human's are the only species that place value in coins or paper or electronic currency to acquire our (i) necessary life sustaining needs or (ii) optional luxury desires, as the rest of the animal kingdom does not use wallets and is only interested in securing necessary life sustaining shelter, food, pro-creation or safety, by stealth and instinct, and don't obtain (acquire) those needs by exchanging with paper or electronic currency. Non-human animals don't go on vacations as they are constantly in a state of instinctive survival.

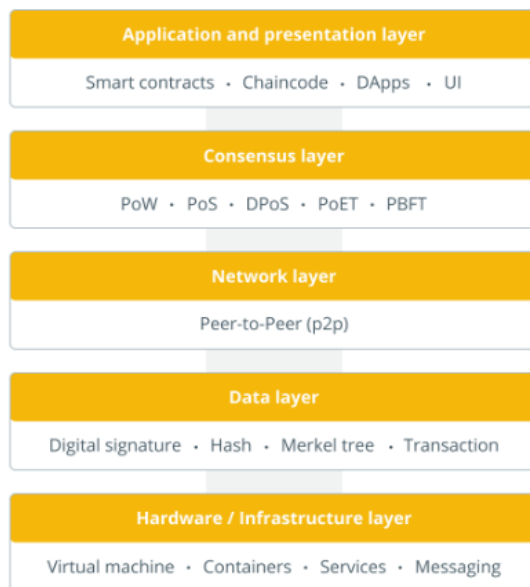
us human's to recognize value in a virtual (not tangible, but computer generated) electronic wallet phantom coin (aka electronic cryptocurrency), in contrast to centralized online internet electronic traded fiat money or even stock certificates, which if desired, can be converted into tangible paper certificates, or storing real paper money in one's hip-pocket leather wallet. There is much debate in regard to what extent cryptocurrency should be treated similarly as a security (stocks) and subject to security and banking regulations and anti-fraud/terrorist laws. More on that in Part 3.

A technical cumbersome definition of Blockchain technology is the **execution (running a programme through the internet)** of computer programmes containing algorithm **applications (bite size standalone computer programmes that calculate or determine a specific objective)**, and after determining that **objective, linked to other computer programmes or algorithm's to collective achieve an overall result.** (*Whew!*)

Basic Blockchain Architecture Layers

Translating this cumbersome Blockchain definition into plaintext English, starts with viewing the overall architecture (design) of what Blockchain technology is all about and is illustrated by the below layer diagram:

Layered structure of the blockchain architecture



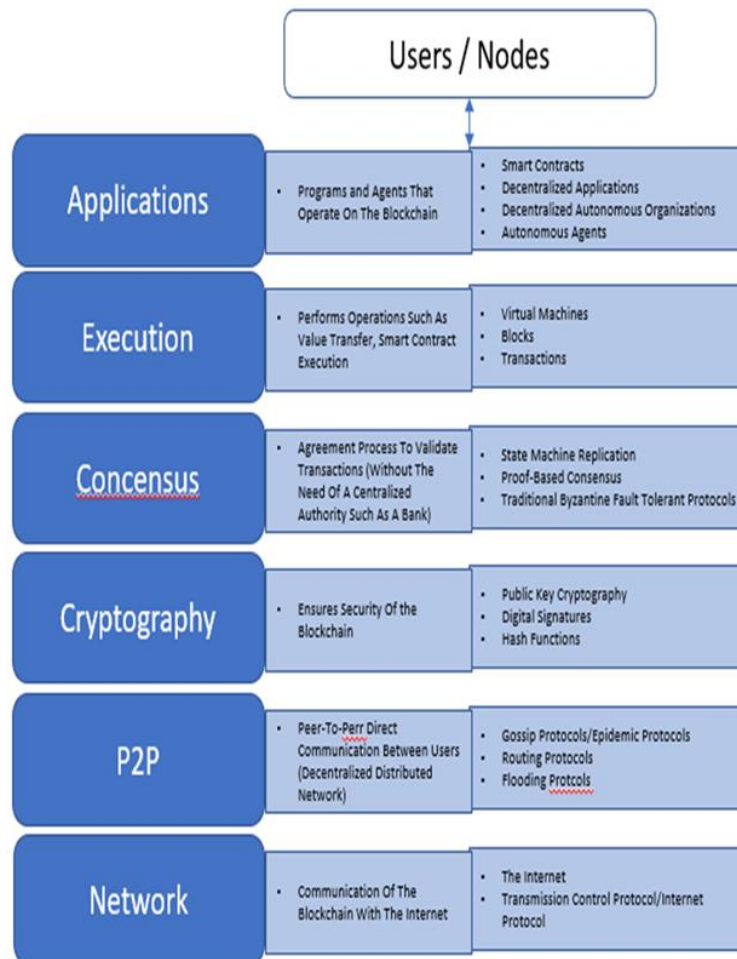
Starting from the bottom layer:

1. **Hardware/Infrastructure Layer:** The Blockchain architecture structure's foundation is first built upon the hardware (computer hardware) used to execute Blockchain computer programmes.
2. **Data Layer:** The data layer next involves the inclusion of transactions data (Alice sales cryptocurrency to Bob) as input into the computer programmes, and transaction data and participating parties identity verified and authenticated and protected by cryptographic techniques (such as use of digital signature keys, hash and Merkel trees, all explained later).

3. **Network layer:** This layer involves the direct communication structure between Blockchain participants or nodes, the peer-to-peer, or direct communication between transacting parties, decentralized contact, without going through a centralized control authority.
4. **Consensus layer:** This layer implements the Consensus Protocols or the rules, procedures and processes how transaction data and Blocks of transaction data are authenticated and securely recorded into the Blockchain data base, and in the case of Public Blockchain, how new cryptocurrency is 'mined' and added to the total quantity of cryptocurrency available for crypto trading. The layer involves application of Block validation schemes such as Proof-of-Work, Proof-of-Stake, Delegated-Proof-of-Stake, or Proof-of-Elapsed-Time (PoET). Proof-of-Elapsed-Time (PoET) is a consensus mechanism often used on Permissioned Blockchain networks (semi-private or private networks not like Public Blockchains which is accessible by any one) to decide the mining rights or the Block winners on the network, in regard to Permissioned Blockchains that choose to pay transaction validators or miners a cryptocurrency validation service fee.
5. **Application and presentation layer:** This layer is the actual implementation of applications such as smart contracts, distributed apps, chain code, user friendly interfaces (UI) between users and Blockchain technology (user friendly computer monitor visual logon and use tools).

Next Layer of Blockchain Architecture

Drilling down a bit deeper into Blockchain architecture:



- **Users/'nodes'** of a Blockchain **network** are the individual computer systems and participants connected to a Blockchain website network through the internet network connection and such individual computer systems running communication software that is compatible with the Blockchain network.
- Each user or node can communicate directly with one another (node-to-node or peer-to-peer or **P2P**) connection (no centralized authority to go through).
- Where Blockchain security (which includes objectives of securing the identity of the user, confidentiality and authentication of transaction information sent through the Blockchain network) is protected by **cryptography** (secret code messaging).
- The process and administrative procedure rules for recording permanent, time stamped dated and authenticated information in the Blockchain network (ledger) is provided by a **Consensus Protocol** (voting rules and procedure that governs how and when information is authenticated and permanently recorded in the Blockchain ledger).
- Inclusion of various Blockchain network computer software programmes, algorithms and **applications** (Apps) operating through the internet and designed to cause desired calculations and transactions to be timely and safely executed.

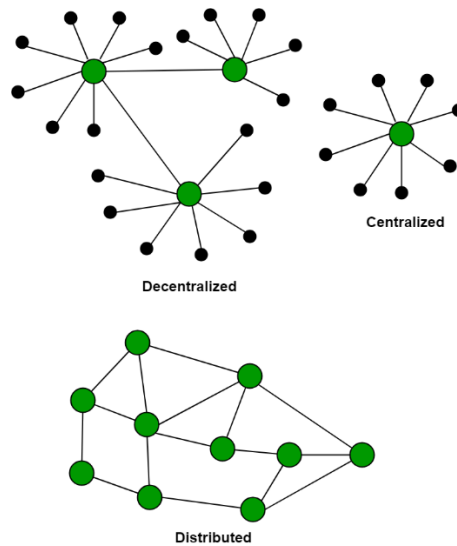
In plaintext English...

Centralized (ledger) means a single central authority (middleman, such as a bank) through which all transactions between users in a network must be routed, and the central authority managing the details of the transaction, its execution and its security.

Decentralized (ledger) – which is not the same as a (**distributive**) ledger, means not using any central authority such as a bank, to process a transaction (such as trading cryptocurrency or other transaction). These decentralized ledgers are automatically, simultaneously and collectively updated as new authenticated and approved transactions take place. Cryptography is used in the structure of decentralized ledger systems, designed to prevent or at least minimize unauthorized third party attacks (from hackers or cyberterrorists), thereby such third party attacks having little if any adverse impacts on the Blockchain network continuing to function, since if any one ledger is successfully attacked and taken out of the Blockchain, the other surviving ledgers continue to operate and business as usual. Hence the *chain of blocks* referred to as the **Blockchain**. In a decentralized network, users communicate and deal with one another directly and not through a central middleman authority.

Distributive (ledger) means the recording of transactions into an electronic ledger (in contrast to a bound paper ledger book where multiple authenticated duplicate copies of the electronic ledger are distributed, saved, stored and sprinkled about throughout the Blockchain network in many node locations, throughout the world and linked together through a central network. These distributed ledgers are automatically, simultaneously and collectively updated as new authenticated and approved transactions take place. Cryptography is used in the structure of distributed ledger systems, designed to prevent or at least minimize unauthorized third party attacks (from hackers or cyberterrorists).

Centralized, decentralized and distributed networks are illustrated in the below diagram...



Trustless means the parties in a Blockchain transaction can be (and usually are) complete strangers and no need to trust each other (unlike a centralized system where trust is devoted to the central authority or bank to authenticate transactions). Trustless trust is provided by electrons and not humans, whereby the Blockchain execution processes, Consensus Protocol and cryptographic software/algorithms, provide the ‘auto-pilot’ necessary electronic processing of trust that transactions are authentic and secure.

Peer-To-Peer means individual parties (nodes) involved in a Blockchain transaction (example: transferring cryptocurrency) communicate and deal **directly** with one another through the Blockchain network connected through the internet without going through a middleman (a central authority such as a bank or stock exchange). This one-to-one or peer-to-peer (P2P) interface adds a dimension of privacy, confidentiality and protection of the identity of the parties and their transaction, since the prying eyes of the absent central authority does not have access to transaction and party information.

Anonymity (privacy) means in a Public²⁰ Blockchain (such as Bitcoin or Ethereum), users (nodes) communicate with each other in the Blockchain network with a designated unique one of a kind public (and private) key²¹ or address (a personal identification ID or tag sort of like a fingerprint). This address or key is generated in advance of communicating in the Blockchain network and used as pseudonymity (substitute name or secret or code name) to protect the user’s birth certificate name identity.

Immutable means once an authenticated and approved time-stamped transaction is recorded in the Blockchain network, it cannot be later deleted or changed²² – a permanent and auditable record.

Non-repudiated means recorded information once placed in the Blockchain, cannot be denied as being authentic by the party who initiated a transaction when that transaction had previously been authenticated, approved and recorded in the Blockchain. (An example of non-repudiation is a ‘no-reserve’ auction. A **no-reserve auction (NR)**, also known as an absolute auction, is an auction in which the item for sale will be sold regardless of price. Once an auction begins and a bidder calls out a bid for an item

²⁰ As later discussed, Blockchains can be three flavors: public, private or permissioned.

²¹ Explained in the cryptography section of this report.

²² As later discussed, there are rare and special circumstances in which recorded Blockchain information can be changed and thus permanent immutability modified, which is referred to as ‘**forking**’.

and it is the highest, the auctioneer is then obligated to sale the item to the high bidder, regardless of the items original owner once hearing the price (and deeming it to be too low) desire to withdraw the item from the auction – cannot deny the transaction, which they are prohibited from doing and thus a ‘forced’ sale, or in the case of the Blockchain, once a transaction is recorded, the author cannot deny its validity).

Nodes means those individual participants or users (having computers and processors) connected to the Blockchain network through the internet. There are different kinds of nodes.

- Some nodes are user nodes only interested in using the Blockchain network as a tool to trade cryptocurrency,
- Some special nodes provide a service of validating, authenticating, approving, recording and storing transaction information and associated parties in the Blockchain electronic ledger, where recorded transactions stored in fixed bite(byte?)-size chunks of data and information, called a ‘Block’ of data, and each Block chained or linked together in a secure communication link (hence the word Blockchain), thereby preserving the historical, chronological time stamped, permanent (immutable) record of transactions. These records are auditable and establish credibility and authenticity of transactions. The cryptography section of this Guide will explain how Blocks of data are created, stored and protected and we shall journey through the exciting world of ‘Hash functions’ – cryptographic mathematical processes that provide security when Blocks are created and stored. This journey will hopefully be one of a peaceful river cruise (plaintext English) and not one associated with life threatening rapids (technically complicated and little understood Blockchain vocabulary rhetoric). These special service nodes, some providing expensive computer equipment required to provide the verification service, are also referred to as a ‘miner’, ‘validator’, ‘authenticator’, or ‘full node validator’, and receive a service fee for their validation services, paid in cryptocurrency.
- Some nodes (referred to as MasterNode) provide the development of Blockchain technology and associated software and algorithms and overall management of the network, that causes the Blockchain to function.

Consensus Protocol means those procedures and rules (an operating procedure or plan, a consensus voting procedure where the vote is taken among Full Node validators or miners) Blockchain node miners must honor in order to authenticate and approve addition of a new Block of data permanently recorded into the Blockchain network. Because there is no central authority authenticating and approving transactions which would otherwise have one-stop autonomous authority to approve and authenticate transactions, the decentralized structure of Blockchain causes the authentication and approval process by many distributed parties to be much more complex. Consequently, much effort is put into computer cryptography (security coding), software and algorithms to establish a secure consensus process (or protocol) structured to ensure objective and efficient means to establish authenticity, security and Block recording approval authority.

The made ingredients for a successful Consensus mechanism includes:

- **Agreement:** All honest nodes decide on the same value.
- **Integrity:** No node can make the decision more than once in a single consensus cycle.
- **Validity:** The value (‘transaction’) agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node.

- **Fault tolerance:** The consensus algorithm should be able to run correctly in the presence of faulty or malicious nodes (Byzantine Faults).
- **Termination:** All honest nodes terminate the execution of the consensus process and eventually reach a decision.

Obtaining Consensus in a decentralized distributed system such as Blockchain is so challenging to design that a theory known as **CAP theorem (aka Brewer's theorem)** has been proven, which states that a distributed system cannot have simultaneously all three of the much-desired properties:

- **Consistency-** all nodes in a system have a single, current and identical copy of the data.
- **Availability-** Nodes in the system are up, accessible for use, and are accepting incoming requests and responding with data without any failures as and when required; data is available at each node and the nodes are responding to requests;
- **Partition Tolerance-** If a group of nodes is unable to communicate with other nodes due to network failures, the distributed system continues to operate correctly.

CAP are challenging to obtain, but CA, CP and AP are achievable.

In Blockchain the **CAP theorem** is tolerated as follows: **Consistency** is (temporarily) sacrificed and not simultaneously achieved with **Partition Tolerance** and **Availability** but is achieved over time. This is called eventual consistency, where Consistency is achieved as a result of validation from multiple Full Validation Nodes over time. It means a temporary disagreement among nodes on the final state of approving a Block for recording, but it is eventually agreed upon. Competitive Consensus Protocol 'mining' and consensus vote on a final state eventually achieves a CAP state.

Miner (aka as Full Node or Validator) means that special node (participant) in a public Blockchain network, that provides a Proof-of-Work or Proof-of-Stake (and others) service of validating, authenticating, approving, recording and storing ledger transaction information and associated parties, in the Blockchain network, where recorded transactions are linked together in fixed bite (byte?) -size chunks of data and information, called a 'Block' of data (hence the word Blockchain), and Blocks chained or linked together in a communication link, thereby preserving the historical, chronological time stamped, permanent (immutable) record of transactions. These special service nodes are also referred to as a 'validator', 'authenticator', or 'full node validator', and receives as an incentive to perform the validation service, a service fee for their services, paid in cryptocurrency. Miners may also provide expensive computer equipment required to provide the service (if Proof-of-Work Consensus Protocol is used). The process a miner uses to provide the Proof-of-Work or Proof-of-Stake (and other similar services) service of authenticating Public Blockchain cryptocurrency transactions and being compensated with a fee for providing such service, is referred to as 'mining' (like, mining for cryptocurrency).

Fiat money means money printed and controlled exclusively by a government and a central bank charged with administering such currency (regulating how much is in circulation, controlling inflation, etc.) and the value of such fiat money established by the full faith and credit recognized by society in the government standing behind the fiat money. An example: the public generally recognizes more faith and value in the US Dollar as compared to the North Korean Won. *What do you want in your wallet?* Fiat money is the currency of a sovereign nation that enforces strict rules and regulations how such currency is controlled and severe penalties (criminal and civil) if a person prints, uses or distributes counterfeit (fake) fiat money. [Fake news is not the only fake thing worthy of punishment]. Cryptocurrency is not fiat money nor is it a

form of counterfeit fiat money but comparable to other tradeable valued assets (such as trading in stocks, bonds or bartering and exchanging valued goods at a flea market – “I’ll trade you two blenders for that micro-wave oven plus 1/100th of a Bitcoin”, possibly a taxable transaction but tough to enforce).

In a nutshell...

Blockchain technology originated from the separatist’s revolutionary (disruptive) desire to:

- cut out the financial industry centralized middleman when parties to a digital transaction (where a transaction can be any digital or electronic activity, such as dealing with an online financial bank account, or investing funds online in an investment such as Bitcoin or other digital currency, which can be verified online²³) wish to gain more control,
- be more independent (not depend on a middleman and independent of big banking business centralized authoritarian *potentially(?)* corrupt structure),
- gain better privacy and confidentiality (for both the parties involved and the content of information and data) and
- do so by concluding a transaction directly between themselves – so called Peer-To-Peer (‘P2P’) transactions, and
- do so in a secure, efficient, transparent (the parties to the transaction see the details of the flow of the transaction unlike going through a centralized authority who limits what a party can see within the borders of the centralized authority),
- trustless (don’t need to trust a centralized authority like a bank or the parties in the transaction, just trust the ‘auto-pilot’ mathematical based process to electronically and objectively establish the authentication of a transaction and its parties and its truth),
- immutable (once an approved timestamped transaction is recorded in the Blockchain network, it cannot be deleted – a permanent record),
- non-repudiated (recorded information cannot be changed, only amendments added to and recorded as an addendum to prior approved transactions) and
- lower cost (no fees are paid to a middleman).

This new separatists decentralized, Peer-To-Peer way of communicating and dealing direct between parties regarding the exchange of digital assets, was viewed as a **‘paradigm’** shift, meaning a new and better thinking-out-of-the-box way of conducting affairs compared to the older traditional centralized way. Even so, the old centralized way is not necessarily broke, it works. (Just like an abacus still works, but there are more efficient options to add up numbers. But if one’s electric power is interrupted – either by the grid being down or a battery depleted and solar cell not working- , the manually operated abacus just might become the calculator of choice over non-operating electronics). As in *Skyfall*, sometimes the ole’ ways are the best.

This new Peer-To-Peer technology gets much more complicated when there are many more parties in an internet network connection and many transactions are to take place, unlike being in a closed room, which requires the need to establish new rules and processes (so called protocols) how transactions are

²³ Blockchain and its authenticity checks and balances works well for digital assets – assets that can be viewed on a computer screen like one’s bank account balance or cryptocurrency wallet account balance.

to take place and authenticated as being the truth²⁴ as well as establishment of effective security systems (using cryptography – secret coding - computer algorithms) to prevent, or at least minimize, double spending of cryptocurrency or malicious parties from disrupting the online internet Blockchain process, or worse, illegally divert wealth (normally digital cryptocurrency) from legitimate Blockchain participant owners to illegitimate parties (that's code for hackers breaking the decentralized security cryptographic codes and stealing Bitcoin or other cryptocurrency).

In Geek Speak, ...

Blockchain is

- A trustless²⁵, decentralized²⁶, organized in ('bite-size') blocks of multiple transactions (containing recorded data and information – a ledger) linked²⁷ to each other, of all (past and present) blocks or digital events that have been executed and shared²⁸ among all participating parties. As new transactions take place, new blocks of data/information, if approved²⁹ by the participants as valid transactions, are sequentially linked (the making of a chain) with prior associated block transactions³⁰.

How Do You Build trust with Blockchain

Blockchain enhances trust across a business network. It's not that you can't trust those who you conduct business with its that you don't need to when operating on a Blockchain network.

Blockchain builds trust through the following five attributes:

- **Distributed:** The distributed ledger is shared and updated with every incoming transaction among the nodes connected to the Blockchain. All this is done in real-time as there is no central server controlling the data.
- **Secure:** There is no unauthorized access to a Blockchain, made possible through **Consensus Protocols** (rigid operating agreements and processes) and cryptography.

²⁴ A truth authentication example is where a party who holds US dollars in their bank account, desires to purchase and invest in Bitcoin, a brand of digital cryptocurrency, owned by another party. The truthing process involves confirming that the party who owns the US dollars is an authentic party and has in fact US dollars in an account that can be used to purchase Bitcoin, that the seller of the Bitcoin is an authentic party who in facts owns the Bitcoin, that the exchange of dollars for Bitcoin can be concluded in a safe and secure manner and recorded in a ledger that is authenticated that the seller now has US dollars and the buyer owns Bitcoin.

²⁵ Trust is not required between the parties nor provided by a centralized authority such as a bank who would otherwise confirm the identity of parties in a transaction and confirm they have funds to conduct a transaction, since the operation mechanics of the Blockchain will determine what is truth and what is not.

²⁶ Noncentralized – no middleman or intermediary – peer-to-peer transactions; database of records or ledgers of transactions in which each participant in the Blockchain has a historical, updated time stamped copy

²⁷ The chain of blocks – hence the coined word 'Blockchain'.

²⁸ Sharing means each participant's computer in the system, has a copy of the time stamped historical and updated block ledgers (when one copy is approved for updating, all distributed copies get updated).

²⁹ The participants/nodes agreement or consensus where Full or Validator node is a special participant in the Blockchain network, entitled to earn a fee in exchange for providing confirmation services that a transaction is legitimate and should be added to the Blockchain as an approved transaction.

³⁰ Thus there is a permanent sequential time stamped history and record – the ledger - of all transactions – a transparent process to establish the truth of transactions. Each transaction is verified (voted on by the Full or Validation nodes as being valid or invalid) by the majority of participants of the system (what is referred to as gaining consensus or agreement that the transaction is valid).

- **Transparent:** Because every node or participant in Blockchain has a copy of the Blockchain data, they have access to all transaction data. They themselves can verify the identities without the need for mediators.
- **Consensus-based:** All relevant network Full or Validator node (miner) participants must vote and obtain consensus that a transaction is valid. This is achieved through the use of Consensus Protocol algorithms.
- **Flexible:** Smart Contracts which are electronically self-executing based on certain conditions (without the need for human performance intervention) can be written into the Blockchain network. Blockchain Network can evolve in pace with business processes.

What Are The Benefits of Blockchain Technology

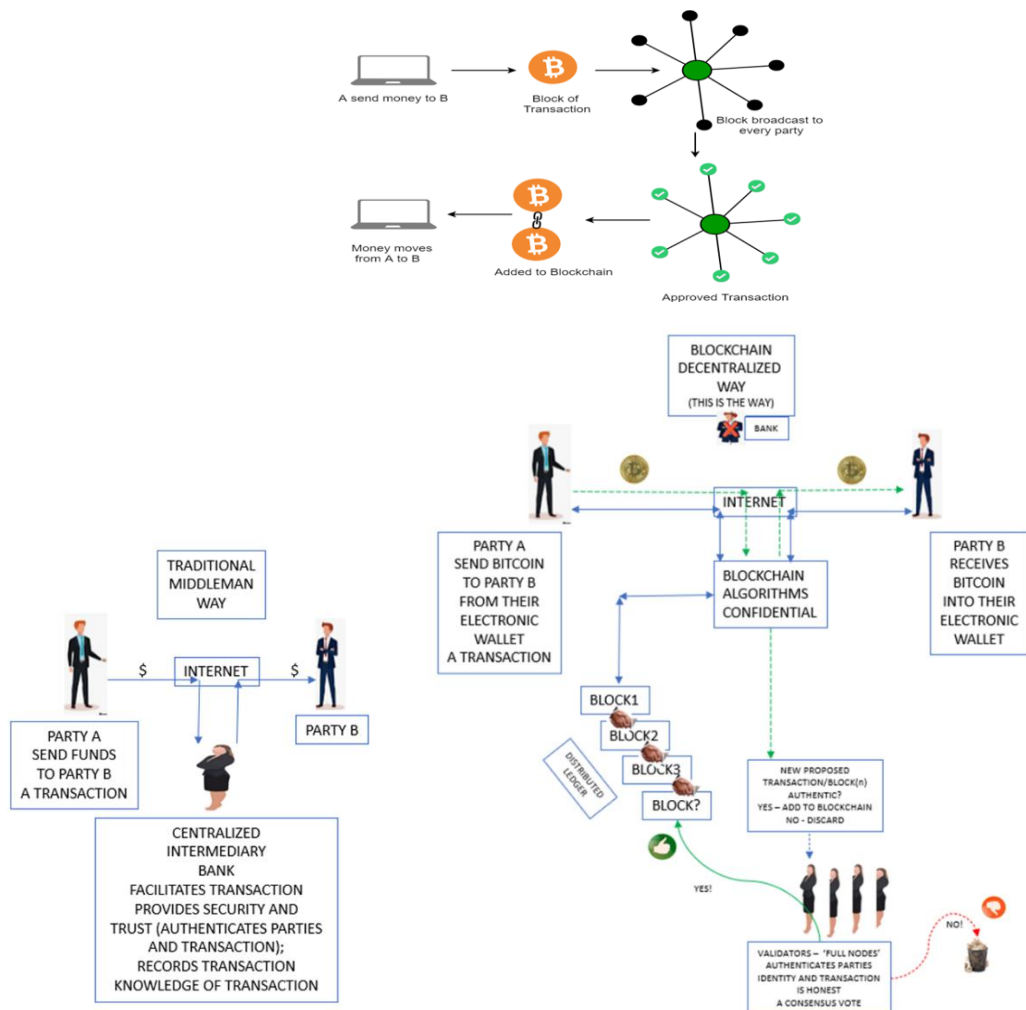
- **Time-saving:** No central Authority verification needed for settlements making the process faster and cheaper.
- **Cost-saving:** A Blockchain network reduces expenses in several ways. No need for third-party verification. Participants can share assets directly. Intermediaries are reduced. Transaction efforts are minimized as every participant has a copy of shared ledger.
- **Tighter security:** No one can temper with Blockchain data as it shared among millions of participants. The system secures itself against cybercrimes and fraud by use of cryptography and Consensus Protocols.

A decentralized trustless Public Blockchain era network transaction is illustrated by the following internet online cryptocurrency transaction example and accompanying diagram...

- Party A owns a Public Blockchain network electronic wallet (a virtual electronic ledger in which ownership in digital cryptocurrency is recorded). Party B owns a similar Blockchain network separate electronic wallet.
- Party A sits at home in her study and signs into the internet from her personal computer which is connected wirelessly to her modem/router³¹.
- Party A enters her ID and password (or logs on with her fingerprint or facial recognition, biometric security) which unlocks the computer. Party A then searches the web using an internet search engine such as Google and locates the website address of the Public Blockchain network of interest.
- Party A signs on to her online digital wallet account by entering her ID and password, and because Client A has setup two factor authentication, the digital wallet login process causes a text message to be sent to Client A's mobile phone citing a secret logon code number. Client A then enters the secret code into her electronic website sign on page which then gives Client A access to the details of Client's A electronic ledger (electronic virtual wallet account).
- The online virtual wallet detail, among other useful information, shows Client A's deposits and debits in the virtual wallet account. This internet online digital (non-paper, electronic) electronic virtual wallet account ledger, viewed from Client A's computer monitor, displays Client A's wallet activities and is Client A's electronic virtual wallet 'ledger', but instead of wallet transactions recorded by hand in a real live paper ledger book, the electronic ledger book is a digital (electronic) substitute for a hard cover accounting book ledger.

³¹ See footnote 10, modems and routers

- Client A sends an online message through her Public Blockchain virtual electronic wallet website, and instructs that website to transfer a certain amount of cryptocurrency funds on a certain date from Client A's ledger account (a debit) to Client B's virtual electronic wallet account ledger (a credit). Client B had previously given Client A, Client B's virtual electronic wallet contact details, which Client A disclosed to her virtual electronic wallet website account for its use in transferring the funds from Client's A account to Client B's account.
- The Public Blockchain, confirms the authenticity of Client A, the transferred cryptocurrency amount, that Client A has sufficient cryptocurrency in its wallet account and also confirms the authenticity of Client B, that it has a virtual electronic wallet account, and their authenticity. After that confirming service, cryptocurrency funds are electronically transferred over the from Client A's virtual electronic wallet account (the debit) to Client B's virtual electronic wallet account (a credit), and does so in a secure manner, designed to prevent third parties from interfering with the transfer or worse yet, such third parties diverting the transferred funds to an unauthorized address.
- Client A and Client B can log onto their respective online internet web virtual electronic wallet accounts, and confirm the transaction, Client A's account will show a reduction (debit) of the sent funds, and Client B's account will show an addition (credit) of the transferred funds.



This new 'fintech' – digital cryptocurrency 'banking' - decentralized trustless Public Blockchain technology...

- Cuts out the controlling centralized middleman – no more bank ,
- Electronically distributes duplicate, constantly updated, authenticated and secure cryptocurrency transaction ledger copies (recording what has been sold, purchased, by whom, when) throughout the Blockchain network (where complete copies of such ledgers are stored and managed by certain computers with processors connected to the Blockchain network through the internet - known as 'Full Node validators') thereby eliminating a single point of failure risk that exists if only one centralized authority (bank) is in control of the recorded ledger. Many distributed copies of the Blockchain ledger, provides protection against Blockchain failure since if any one ledger is down, the rest can function and keep the Blockchain alive and well,
- Reinstate independence of parties to a transaction by being able to deal directly with each other (at least the illusion of direct dealing because the parties still communicate through the 'centralized' internet and Blockchain network - but no actual in person meetings) – so called peer-to-peer contact – one on one connection - (the good ole' days),
- A way to claw back privacy and anonymity between the parties in a transaction (no middleman prying eyes),
- Reduce costs (bank overhead charges are high, Blockchain fees are less),
- Improve efficiency (bank transaction settlement time can take days, Blockchain a matter of minutes)

Because Blockchain technology depends on use of the internet, both Blockchain and the internet, controlled by computer scientists and engineers, massage electrons in a complex computing environment/infrastructure, much of which is managed by black boxes that only the expert creators of such systems understand and operate. Consequently, there are many mysteries lingering in the Blockchain technology and internet black box technologies not readily or easily understood by those who use the systems... and certainly aren't DIY repairable, if a fault, by the users .

The distributed nature (many places to advance transactions as well as to hide) and electronic complexity of the Blockchain decentralized system can be a breeding ground for malicious cyber attackers or hackers (the bad folk) ***to find or develop 'electronic' backdoors and attempt to disrupt, disable, destroy, or maliciously control the Blockchain computing environment/infrastructure, or destroy the integrity of the data or steal controlled information.***

Because of the potential risks of ...

- (1) harmful electronic maliciousness from cyber attackers or hackers,
- (2) as well as the absence of trust among parties, total strangers, to a Blockchain transaction

Blockchain is fundamentally a cryptographic³² project structured to ‘automatically’ through objective emotionless mathematical algorithm³³ processes, to establish trust by electronic means and not by face-to-face authentication, and a structure that establishes the authentication and permanent immutable (can’t be changed) history of all transactions.

This means, Blockchain computer software and associated algorithms, operate in ‘auto-pilot’ mode and is structured such that Blockchain computations automatically establish authenticity and integrity in the system – a self-serving and self-executing trust environment, thereby trust is not independently established in face-to-face encounters or required between parties in the system (a ‘trustless’³⁴ environment) – electron system processing trust governed by mathematical processes - displaces human trust. This ‘auto-pilot’ trust operation includes defense security tactics against trustless cyber attackers and hackers (the bad folk).

The Curious Case of Cryptography....Because cryptography is so central to Blockchain, a separate section has been devoted in this Guide to cryptography, to explain what it is all about and how it applies to Blockchain.

Satoshi Nakamoto

Any study of Blockchain technology will result in the early identification of the mysterious **Satoshi Nakamoto**. Building on previous work of others, Blockchain was originally discussed/created to serve as a separatist’s public decentralized ledger for digital (bitcoin) cryptocurrency finance and investment transactions, using cryptography to advance the technology. The birth of the Blockchain idea was authored by and credited to a person (or group of people?) using the pseudonym Satoshi Nakamoto, in a 2008 publicly published discussion whitepaper. Since that publication, Blockchain technology has grown exponentially because of Blockchain Developers.

Publications about Mr. Nakamoto focus on his/her/their Blockchain genius and little has been written why Mr. /Ms. Nakamoto elected to remain mysterious and anonymous.

Question?...Was the real secretive Satoshi Nakamoto concerned he/she/them, might be subject to civil penalties or criminal charges as a result of accused violation of certain laws, because of his/her/them publication of the 2008 whitepaper that initiated Bitcoin cryptocurrency and the concept of Blockchain?

DMCA. Perhaps just another phantom conspiracy theory...*but* ...

In 1998, the United States passed into law the **Digital Millennium Copyright Act (DMCA)**. That law, among other things, issued very broad prohibitions against violating copyright rights, including prohibitions against cryptanalytics (the analysis of cryptography or security code writing used to keep

³² See footnote 11

³³ Algorithm is a well-ordered collection of unambiguous and effectively computable mathematical equations and operations that, when executed (or implemented), produces a desired result and halts the operation in a finite or fixed amount of time. In effect a computer program recipe for executing a mathematical calculation.

³⁴ A trustless system is one in which users equally trust and mistrust each other within the network. Trustless in the sense that the Blockchain transaction authentication electronic process provides the necessary trust – trust that users and transactions are honest – thereby eliminating the need for users to depend on personally trusting each other in a transaction or need for a centralized trusted authority; this is in contrast to a centralized system, such as a bank, where the intermediary bank involved in a transaction between users, provides a third party middleman trust service of authenticating user identities and the truthfulness of a transaction as well as a secure network.

private or confidential one's identity or transmission of messages) that could be maliciously used directly or indirectly to circumvent copyright cryptographic security processes (such as a security breaking computer code or algorithm that allowed a hacker, without authorization, to unlock, copy, use and distribute, protected copyright material – such as music, electronic documents, etc). Penalties under the law if violated can include civil damages (injunctions, attorneys fees) and/or criminal charges (up to 3-5 years in prison, fines, injunctions).

Mr. Nakamoto's 2008 paper could have been viewed as containing cryptanalytic technology advice that could be interpreted to violate the DMCA. There were various legal and criminal claims under the DMCA before the 2008 publication, which would give rise to concern to those involved directly or indirectly in cryptanalytics.

Cryptography Export. Perhaps there were concerns regarding the uncertainty of penalties (civil and or criminal) that could be incurred because of the ever changing and uncertain landscape regarding restrictions associated with the export of cryptographic technology from the United States. Such restrictions tied to protection of National Security (spy stuff), thwarting terrorism, etc. The risk of such penalties if accused of illegally exporting sensitive cryptographic technology, has been somewhat reduced in recent times because of (1) court challenges against export controls, associated with ever changes in technology that converts an otherwise export controlled event into public domain knowledge (not unlike the spread of the internet – now very much a public asset - whose beginnings started as a secret U.S. Department of Defense classified research project) and (2) challenges based on the preservation of constitutional protected free speech rights (there is a fine line between illegal cryptographic computer software export control rules and constitutionally protected – at least in the United States - freedom of speech publication rules). The ubiquitous (its everywhere) use of the internet has resulted in wide-spread publication of information that makes cryptographic export controls difficult to rationalize and enforce, if not problematic to do so.

“Plain Sight Security”...Because the techniques and processes involved with security cryptography (secret coding stuff) can be readily understood and disclosed – by hackers and non-hackers alike - security now has to be implemented by processes that are in ‘plain sight’ (the security process is published in the public domain, but the mechanics of the process – complex random based mathematical calculations the solution of which are not easily determined even knowing the equation) - is what provides the secrecy and security), and cryptography is a tool to facilitate such “in plain sight security”.

Side stepping the risk...Publish or Perish... Perhaps the real Satoshi Nakamoto, recognizing the merits of Blockchain and Bitcoin (cryptocurrency) technology, envisioned as a way to cause their development without him/her/them being accused of DMCA or cryptography export violations, was to plant the seed of the Blockchain concepts in the public domain through the anonymous published whitepaper under the Satoshi Nakamoto pseudonym (an awkward **person of interest** to identify and charge with a violation of law) and let society advance the technology at a pace and world-wide distribution, much more rigorous than one developer, potentially personally exposed to law violation charges, could advance. Therefore no real author(s) of the whitepaper could be easy target(s) of DMCA or cryptographic export violation claims and as defendant(s) in a complaint, and potentially suffer civil and/or criminal penalties. The real Satoshi could then ride the coat-tails of world-wide Blockchain and Bitcoin developments (and most likely

one of the now numerous world-wide Developers) and enjoy the benefits of the effort, with little risk of DMCA or cryptographic export violation sanctions.

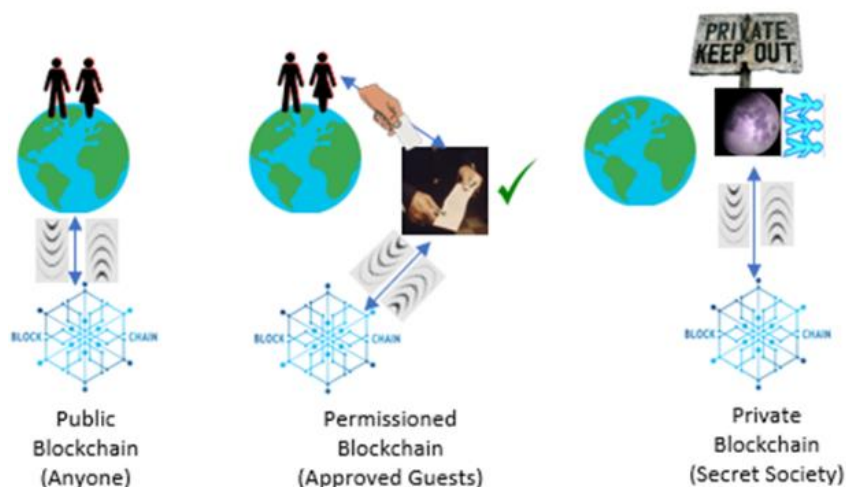
There have been various court cases challenging the DMCA (constitutional freedom of speech arguments) as well as legislative efforts to improve the law and make it less onerous particularly in regard to ‘honest’ cryptanalysts. Thus, because of the dampening of both the DMCA and cryptographic export control legal restrictions, perhaps it is time for the real Satoshi Nakamoto to come forward and reveal their true identity and bask in the rays of acclaimed technology success – however, given the passage of time and world-wide involvement by many Developers in the technology, it might be difficult to determine who is the real Satoshi Nakamoto (*No!, I am Spartacus*), regardless how many step forward to claim the title.



Who is the real mysterious Satoshi Nakamoto? Only the Shadow knows...

Three Flavors of Blockchain

Blockchains come in three flavors: **Public_{Trustless}** (also called ‘permissionless’ Blockchain), **Permissioned_{Trusted}** and **Private_{Trusted}** (Permissioned and Private Blockchains are also collectively called ‘permissioned’ Blockchains)



- **Trustless Public Blockchains (also referred to as Public Ledgers or Permissionless Blockchain – an electronic distributed digital record of transactions maintained in a computer network, accessed through the internet)** are large decentralized trustless networks (like the Bitcoin or Ethereum Blockchains) not owned by anyone, that operate with their own separate native cryptocurrency (digital currency), such as Bitcoin (native to the Bitcoin Blockchain) or Ether (native to the Ethereum Blockchain), have open-source code³⁵ that their development community maintains and the network is open to anyone (participating parties can be complete strangers) to participate at any level (a **user node** – a participant of Blockchain primarily interested in its application services such as a trader in cryptocurrency, **full node validator also known as a miner** – a special participant who is authorized to authenticate Blockchain transactions and permanently

³⁵ Open source code means computer software that is in the public domain and anyone can inspect, modify, and enhance (with voting rules and procedures to control how such code is modified or updated).

record the results in the Blockchain or **Masternode** – a special participant primarily responsible for writing and implementing the computer programs that make Blockchain work).

Convertible virtual currency, or cryptocurrency³⁶ use is necessary with a Public Blockchain, as both (1) a speculative finance scheme, and (2) a way to issue ('mint') new digital cryptocurrency (put more in circulation) and compensate and incentivize full node validators, to be active in providing the network validation and transaction authentication services. Full nodes validators or miners, provide a service by participating in the authentication process associated with transaction validation services, and when any one full node validator first publishes their validation results, such validation is then approved as being truthful by a vote from other full node validators (an approval 'voting' process, known as the **Consensus Protocol**, a set of approving rules and procedures) and when approved, the authenticated transactions permanently recorded in the Blockchain ledger. This full node validation service is offered in exchange for a service fee paid in cryptocurrency – thereby resulting in new cryptocurrency added into circulation (where the cryptocurrency compensated validation service, for the Bitcoin Blockchain process is called 'mining'. Details of the mining service will be later discussed). Trust is not needed in a trustless Public Blockchain between parties in a transaction because the Blockchain software/algorithm transaction automatic validation self-executing process execution mechanism provides the necessary independent and reliable 'electronic' trust to authenticate parties and the truthfulness of a transaction, regardless of the honesty characteristics of a party. An example of a Public Blockchain is anyone can invest in Bitcoin by logging on through the internet, connect to the Blockchain Bitcoin network, setup electronic wallet accounts and conclude a purchase or sale of Bitcoin by following the instructions on such network. The transaction is secure but not 'private'.

- **Pros:** Public Blockchain is a valuable solution to financial and other industries, from the point of view of a truly separatists, decentralized, democratized, trustless and cheaper central authority-free operation.
 - **Cons:** Public Blockchains can incur heavy energy consumption required to maintain them; lack of complete privacy and anonymity; and can attract participants who may not be honest in their intentions. Most Public Blockchains are designed for trading in cryptocurrencies, which by nature of their desired value are a prime target for hackers and misfits. Because of human greed, misuse or abuse schemes, can infiltrate the purity of Blockchain ideological goals, necessitating an assessment of how innocent harmed parties can be protected or restitution made to make them whole again.
- **(Trusted) Permissioned Blockchains (also referred to as Permissioned Ledgers)** is a mix between Public and Private Blockchains and support many options for customization (in effect a Public

³⁶ Cryptocurrency is like any currency 'token' used for investments or exchanged to purchase things. Such tokens only have 'value' because of the parties using such tokens 'recognize' and accept the token as having value. (No different than an island villager trading sea shells for a bucket of berries – the parties recognize value in the trade that 2 sea shells are 'worth' a bucket of berries). Consequently, cryptocurrency is recognized to have value only because society gives it that recognition. That value recognition is typically measured in the value of 'independent' decentralized cryptocurrency compared against dependent centralized controlled fiat money (ideally stable and globally recognized tradeable 'hard' currency – governed by an issuing sovereign nation) such as so many cryptocurrency coins equal so many U.S. Dollars, British Pounds, Euros, Canadian Dollars, Australia Dollars, etc. Value, like beauty, is in the eye of the beholder. Because digital currency 'value' can fluctuate at the whim of the beholder, its value history, say measured against the U.S. Dollar, has been quite volatile and dramatic. It could be used for get rich schemes are an express ticket to bankruptcy - not quite as uncertain a risk as a trip to the gambling casinos – but still a risk, at least compared against the more stable hard currency such as the U.S. Dollar. What's in your wallet? (possibly both – JIC).

Blockchain but with regulated access). It is a Blockchain where participants of the network are already known to each other (not strangers, such as different government departments interfacing with one another in their day to day business and a need to have a platform to share secure data and information) and trusted and each participant has an assigned role to play in the network. Anyone can access a Permissioned Blockchain, as long as they have permission to do so. Permissioned Ledgers do not need to use a cryptocurrency incentivized distributed Consensus Protocol voting mechanism to provide trust (that is, such network does not need to pay participants in cryptocurrency for providing transaction authentication services, though such tokens can be used if desired); instead, an agreed set of rules of conduct, an operating agreement (called an agreement protocol, similar to a shareholder or partnership management agreement), is honored by the trusting participants to maintain a shared vision of the truth about the state of the records on the Permissioned Blockchain (an internal authentication process for transactions and the parties involved). As provided by the terms of the agreement protocol, verification of transactions on the chain are confirmed by ‘verifiers’, where verifiers, selected members in the Permissioned Blockchain, are pre-selected by a committee made up of members from the Permissioned Blockchain (a form of a central authority). There is no need for a cryptocurrency compensation verification service mining mechanism (typically used in Public Blockchains – using Proof of Work or Proof of Stake compensated authentication services), since the Permissioned Blockchain verification process is not intense or complex, especially since trust between the members is already established. Permissioned Blockchain core code may or may not be open source.

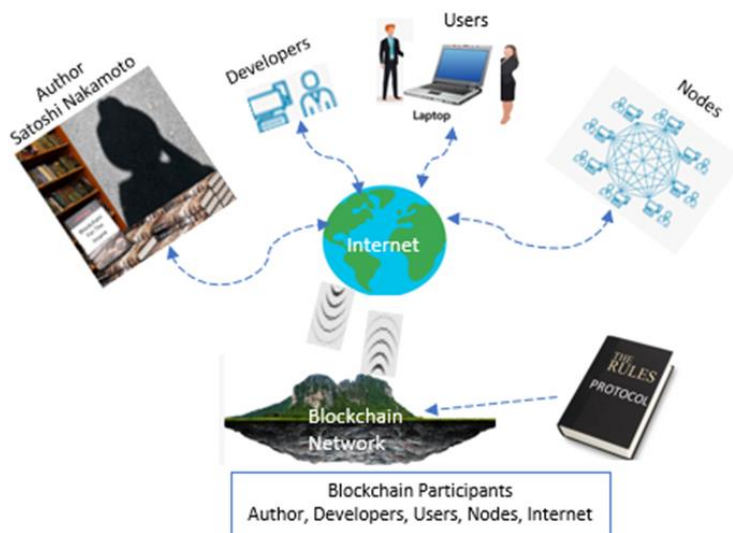
- **Pros:** Permissioned Blockchains allow anyone to join the network after an identity verification process; establish special and designated permissions for members to perform only specific activities on a network; can generate revenue by ‘renting’ a ***Blockchain-as-a-Service (BaaS) structure***— such as renting out a Permissioned Blockchain for the purpose of (1) improving transparency and accuracy in a third party business accounting process and financial reporting or (2) provide an interface service where entries are made by third party end users in the third party’s business and then the rented Blockchain automates the rest of the accounting processes.
 - **Cons:** Permissioned Blockchains are vulnerable to hacking (user information stolen or accounts hacked – data held hostage until ransomware is paid or sensitive information stolen and misused); networks that connect the users to the service depend on security measures that could be bypassed. As with Public Blockchains, human greed, misuse or abuse schemes, can infiltrate the purity of Blockchain ideological goals, necessitating an assessment of how innocent harmed parties can be protected or restitution made to make them whole again.
- **(Trusted) Private Blockchains (also referred to as Proprietary Blockchains or Distributed Ledger Technology (DLT) or Permissioned Blockchain)** do not utilize tokens or cryptocurrencies, tend to be small, establishes a mechanism to share confidential databases and provide some level of guarantee of the authenticity of data, and closely controls its membership, where members already trust each other and readily trade confidential information. Participants can join a Private Blockchain network only through an invitation where their identity is authenticated and verified. The validation is accomplished by the network operator(s) or by a clearly defined protocol (set

of rules) implemented by the network through automated approval methods (so called smart contracts - a computer program or a transaction protocol or transaction execution procedure – that electronically and **automatically** executes – no human contract performance intervention is required -, control or document events and actions according to the terms of an underlying ‘written’ contract or agreement, in contrast to a conventional contract performance where the contracting parties provide the physical manual, not automatic, performance duties. Smart contracts are, absent a special law to the contrary (some States have passed laws deeming smart contracts a being legally binding ‘written’ contracts), not legal agreements in and of themselves and can’t be enforced in a court of law as a legal contract, but is viewed rather as merely a *means* of performing obligations and not a legally binding arrangement. Technically, Private Blockchains are not Blockchains because they deviate from the core concept of decentralization. The private ‘club’ architecture nature of a Private Blockchain is comparable to a de facto centralized structure. An example use of a Private Blockchain might be to allow for collaboration and the secure sharing of sensitive continuously updated data among various government departments.

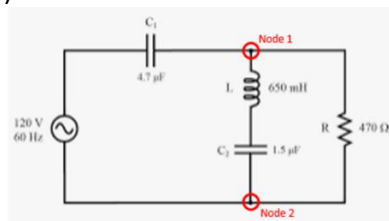
- **Pros:** It is a distributed ledger (sort of like a central member approving bank, connected to many branch banks) that operates as a closed database (among the branches) secured with cryptographic techniques. Only those with permission can make transactions, or validate/authenticate Blockchain data base changes. Private Blockchains prioritize efficiency and immutability—the state of not being able to be changed (a permanent record).
- **Cons:** A Private Blockchain is not decentralized. Private Blockchains are not widely applicable; built to accomplish specific tasks and functions; susceptible to data breaches and other security threats (because of a limited number of validators used to reach a consensus about transactions and data. The more validators the more secure a system. Private Blockchain by their very nature have few in number participants when compared to Public Blockchains).

Five Flavors of Blockchain Participants

There are five flavors of Blockchain participants which includes: the **author, developers, users, nodes** and the **‘internet’**.

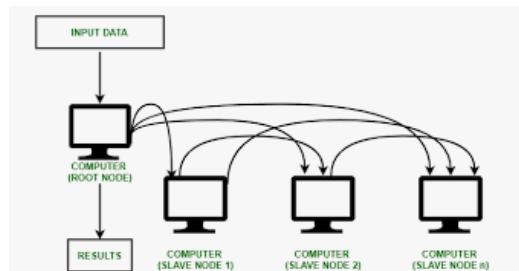


- **Author:** The master author of Blockchain is the mysterious Satoshi Nakamoto.
- **Developers:** Development of Blockchain technology is conducted by **Developers**. **Developers** seek to use the unique features of Blockchain technology to solve data and information exchange problems and create other opportunities. **Developers** program, develop, and test software and systems. There are two types of **Developers**:
 - **Core Blockchain Developer.** Core blockchain developers develop and maintain the architecture of Blockchain systems. They design protocols (operating plans and agreements – the rules of operation), develop security patterns, and supervise the network as a whole.
 - **Blockchain Software Developer.** Blockchain software developers build applications (computer programmes and algorithms) onto existing Blockchain platforms. They handle front-end and back-end development, design, and maintenance. They are programmers.
- **Users:** Users are the participants in a Blockchain that are the senders or receivers of transaction information. An example User includes a User named Alice (the sender of data) sends cryptocurrency (an example of a transaction) from Alice's cryptocurrency digital virtual electronic wallet account (her electronic wallet) to a User named Bob (the receiver of the transaction) and his cryptocurrency account (his electronic wallet). User's depend on the integrity and security of the Blockchain network and the internet, to ensure the intended transaction takes place and recorded in a secure manner, and shielded from unauthorized prying eyes (Eve, the eavesdropper).
- **Nodes:** A node is *technically* identified as an intersection point or connection in a telecommunication network or electric circuit (such as three wires connected together at a common point through which electrical signals can pass, the point or connection being a 'node', which we can call a little node, see diagram below).



(Illustration: 2 little nodes in this circuit).

- A node can also refer to much larger bundled components (group of computers and systems connected together to provide a common function, which we can call a large node, see diagram below) of any system or physical equipment connected to a network capable of performing specific duties such as creating, receiving, or sending data across a communication channel. Size matters as nodes can be small or large, simple or complex.



(Illustration: 4 big nodal system)

- In regard to a Blockchain network, the term “**node**” is commonly used and is specific to an individually owned physical (1) computer and (2) processor (see below diagram) possessing a digital Internet Protocol (IP) address (its fingerprint, used to identify and locate a specific computer through the internet), linked to the internet which is then linked to and part of a Blockchain network (such as a cryptocurrency finance Public Blockchain) and each **node** acting as communication endpoints performing certain tasks. Tasks can include
 - producing or sender node. requesting that a transaction take place on the Blockchain network (a User **node**), and a receiving node, the destination node to which a transaction is sent for the useful benefit of the sender and receiver nodes (another User **node**),
 - a storing node, a node at which all Blockchain transactions (actually fixed chunk size Blocks of transactions) are recorded and stored (and can be a lightweight **node** or **full or client node**, later described in detail),
 - a validating (miner or full node validator, later described in detail) node or
 - a moving data node (lightweight **node** or full **node**), causing the transfer of cryptocurrency from one User account to another, or creating new cryptocurrency, or validating and recording a transaction in the Blockchain ledger.
- A **node** for all practical purposes is an electronic device that is connected (through the internet) to and a part of the Blockchain network and possesses a digital Internet Protocol identification address (a fingerprint or unique serial number).

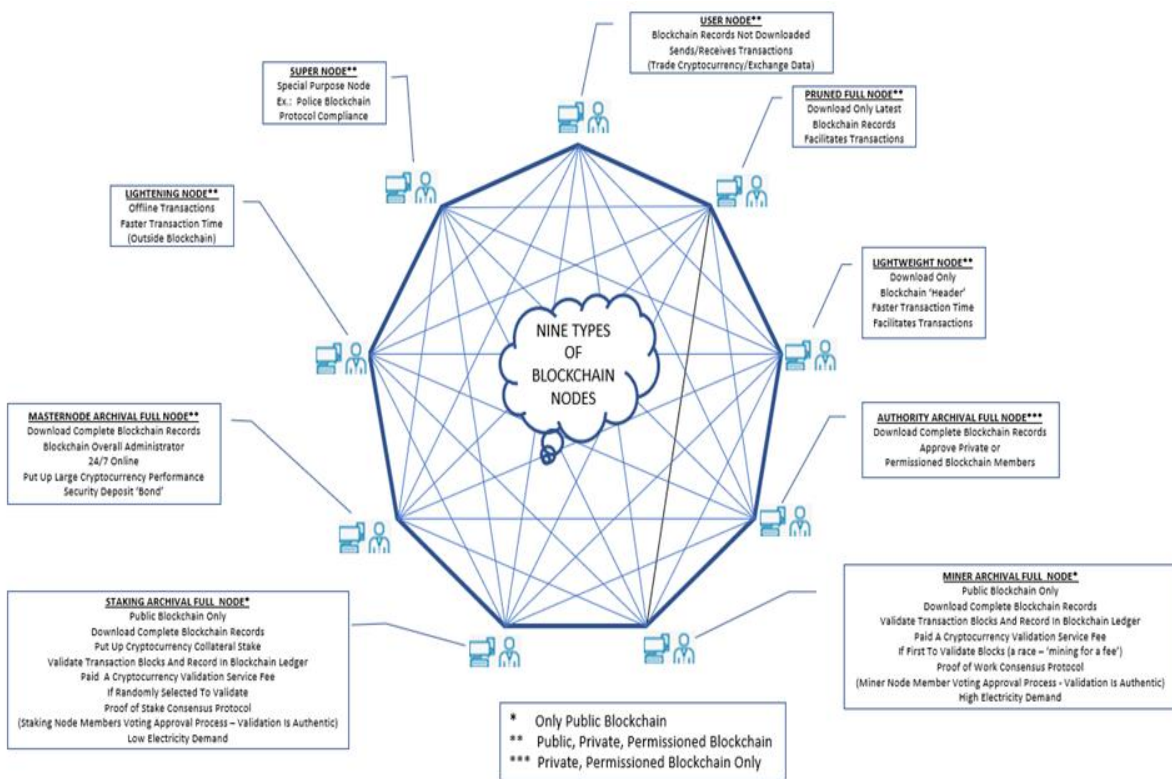


(Illustration: Blockchain with 8 ‘nodes’ attached)

- All **nodes** are capable of sending and receiving messages to and from each other. **Nodes** can be:
 - **Honest** (send and receive truthful information),
 - **Faulty** (a **node** computer system is turned off or has some non-malicious unintentional fault (as later described as a **Byzantine Fault**) such as broken hardware, faulty software, loss of power) or
 - **Malicious** (another **Byzantine Fault**), is where a **node** intentionally inflicts malicious conduct upon the Blockchain network such as by attempting to steal cryptocurrency from others, send untruthful information into the Blockchain system and done so contrary to the rules (protocol) of the Blockchain and gain an unintended benefit or advantage to the detriment of another node or honest Blockchain participants). **Malicious behaviour is behaviour intended to disrupt, disable, destroy, or maliciously control a computing environment/infrastructure or destroy the integrity of the data or stealing controlled information.**
- **Nodes** are assigned to perform certain functions in the Blockchain (some may act as an electronic wallet where they only store [cryptocurrency] ledger accounts; some are just Users, sending and receiving transactions; some may be **full clients** or **full node validator or miner** nodes, providing services to the entire Blockchain and also running algorithms that are securing the Blockchain network. **Full node validators (or miners)** store a complete historical record or ledger of all the transactions that were ever recorded in the Blockchain. Some may be ‘mining’

nodes, and that node known as a ‘miner’, responsible for validating transactions (for a fee – payment in cryptocurrency) and participating in consensus voting to approve the addition of new Blocks of transaction data into the Blockchain. ***In the special cryptographic section in this Guide, is an explanation how mining and the recording of Block information in the Blockchain works.*** Nodes are located all over the world and can be operated by anyone though in the special case of **Full node validators**, it’s may be difficult, particularly in a Proof-of-Work environment, to provide expensive computer hardware equipment– and those that act as **Full node validators** in a Public Blockchain do so because they are compensated with cryptocurrency for performing that service.

- Depending on the operating rules, its protocol, of the Blockchain, the definition of a node may differ somewhat. For example, a Blockchain node might be defined to include a fax machine, three laptops, and a file server. That system in this scenario could have five defined ‘nodes’, each with its own identification address (their MAC³⁷ address) used to connect to and communicate with such devices over the internet and the Blockchain network. (Probably more than we needed to know...)
- Nodes (illustrated and described in the below diagram) are characterized as **User, Full (Pruned, Archival, Authority, Miner, Staking, Master), Lightweight, Lightning, and Super, plus the Internet (also viewed as a node).**



³⁷ A MAC (Media Access Control) address, is a unique (no duplications allowed), 12-character alphanumeric (numbers and letters) digital number (similar to a serial number) that is used to identify individual electronic devices on a network and is the electronic device’s physical address. An example of a MAC address is: 00-B0-D0-63-C2-26 which could refer to a specific electronic device such as a modem physically located at a physical location. Access to such MAC address by an internet network, requires contact with such address in order to locate, connect and communicate with the electronic device at its physical address. In comparison, the internet protocol digital address -- or IP address -- identifies the device globally, sort of like arriving at someone’s home address (the IP address), then going inside the home and locating the fax machine (the MAC address).

- **Full Node Validators**
- A Blockchain **Full Node Validator's** primary job is to confirm the legality or authenticity of each batch (or Block) of Blockchain network transactions (such as multiple requests from multiple parties to transfer multiple cryptocurrency transactions), where such batch of transactions are lumped together in the Blockchain into what is known as Blocks (each Block is limited to a certain bit³⁸ size of data – where bit is the fundamental binary language computer's understand). A unique identifier or address (similar to the Media Access Control -MAC, or Internet Protocol – IP, addresses) for each node/User in the Blockchain network, distinguishes one from another. There are two categories of Full Node Validators (**Prune and Archival**) and a special **Block Signer Node**:
 - **Block Signer Node** validate and digitally sign the transactions and then either record Blocks in the Blockchain (in non-Public Blockchains) or in Public Blockchain transfer validated transactions from the Mempool to Full Node Validators for recording of Block data.
 - **Pruned Full Nodes** store Blocks of data on a hard disk by pruning older blocks. They have limited data storage space. Pruned Full Nodes first have to download the entire Blockchain on their hard disk and then delete the older data Block by Block, starting from the beginning. They keep deleting the older blocks until the storage contains only the recent most transaction records up to the limit of their data storage space. The pruned remaining most recent transaction data is synced with the Blockchain network to keep adding the newer records to its storage and simultaneously deleting the older ones.
 - **Archival Full Nodes** store and maintain the entire Blockchain database. They have no defined storage limit and are the most common type of Blockchain Full Node Validator node. Archival Full Nodes are further categorized as **Authority** nodes, **Miner** nodes, **Staking** nodes, and **Master nodes**.
 - **Authority Nodes**
 - In a Public Blockchain, anyone becomes a node by downloading and synchronizing the Blockchain data with the Blockchain network. However, this access is limited only to a few Authority Nodes (*Traffic Light Nodes*) when it comes to Private and Permissioned Blockchains which is not Public and limits participants by invitation or vote. Authority Nodes control and restrict the access of who can participate and be a member in a Private or Permissioned Blockchain.
 - **Miner Nodes (create new Blocks and mint cryptocurrency)**
 - Miner (same as Full Node Validator) nodes are the integral components of the Public Blockchains using a **Proof-of-Work** (later described in detail) Consensus Protocol (operating rules) model, such

³⁸ Bits – the most basic 'language' a computer understands; Bit is the combination of the words **binary digit** (derived from the binary base 2 number system, where numbers, letters, symbols, etc. can be represented by a combination of zeros and ones); a bit is either the digits zero '0' or one '1' (in computers, 0s and 1s are determined whether or not a electric current is flowing through a circuit controlled by an on/off switch – like a water faucet valve, when the water is on, a 1 is produced, when off, a 0 - called a logic junction, a 1 means current is flowing, a 0, no current is flowing). Eight bits are equal to one byte in computer speak. A computer's compiler (a hardware device that translates computer binary 0 and 1 code into everyday plaintext language and vice versa, such as English or pictures) converts the unique combination of bits (0s and 2s) into a letter, number, symbol, word, picture or graphs displayed on a computer monitor, easier understood by the human viewer. For example, capital letter 'A' binary code is 01000001; the binary code of lower case 'a' is 01100001. The computer uses the binary number understood language in its operation, us human's view the binary number as a readily understandable plaintext character, in this case an A or a, on the computer's monitor screen as our understood plaintext English language.

as Bitcoin. These nodes have to solve complex mathematical problems (described later) to authenticate and approve transactions on the Blockchain. These problems require highly complex computing devices and a lot of electricity. Once a miner node is the first to finish solving the problem and approved (by the other miners) to add a validated Block of transaction records to the Blockchain, it is rewarded with some newly-issued ('minted') cryptocurrency tokens as an incentive for providing the Proof-of-Work validation service.

- When a miner is the first to verify the authenticity of Block data in a Public Blockchain (the verification consensus process will be discussed in the cryptography section of this report which involves using cryptographic mathematical techniques to confirm the truthfulness of Block data) –such first place verification success will result in a payment to the miner of a mining or cryptocurrency service fee if the Block is approved in accordance with the Consensus Protocol (approval voting procedure), and then added to the Blockchain. The successful node or miner will broadcast (notify and communicate with) to other nodes as required by the Consensus Protocol, advising such nodes of the miners request to add the new verified Block of transactions to the Blockchain. Based on the legitimacy and authenticity of authorized signatures and transactions associated with a Block, and confirmation that the requesting recording node was the first to successfully solve the complex mathematical puzzle, as assessed by the other mining nodes, mining nodes then vote (and the pass mark vote requirements specified in the Consensus Protocol) and either accept the addition of the new Block (the normal expected case) or reject it. (Nodes are expected to treat other nodes with respect and not act maliciously or precariously when approving a Block addition request and always vote in good faith since miners will want to be similarly treated if they are a successful miner – do unto others as you would want them to do to you...sort of philosophy). When the successful full node receives sufficient vote approval to record a Block, the successful full node validator or miner, then permanently records and stores (links) the new approved Block on top of the existing Blocks in a chain of Blocks whereby all nodes in the Blockchain network will then have access to the updated Blockchain, thereby growing the chain of Blocks.
- **Staking Nodes (create new Blocks and mint cryptocurrency)**
 - The nodes which verify the validity of transactions in the Blockchains using the consensus model Proof-of-Stake – later explained in detail, (in contrast to the high energy consuming Proof-of-Work consensus model originally used by the Bitcoin Blockchain network) are called Staking Nodes. To set up a staking node, full nodes have to dedicate as collateral their proposed amount of native tokens (a performance security deposit of cryptocurrency that is used by the relevant Blockchain network) on that Blockchain. Then the system selects as validators those Staking Nodes who have deposited the highest stake, and authorized to process a Block of transactions (confirm

their authenticity) and record them in the Blockchain ledger. The selection is made according to some predefined protocol rules. For example, some Blockchains consider the amount of dedicated cryptocurrency collateral funds, while some consider the amount of time spent on validating transactions on the Blockchain. Staking nodes consume significantly less energy than Proof-of-Work miner nodes to validate transactions because Proof-of-Stake does not require the special equipment investment nor computational power and electricity consumption required to solve complex mathematical puzzles.

- **Masternodes**
 - Masternodes are more powerful than regular Full and Lightweight nodes. In addition to validating, preserving, and broadcasting transactions, Masternodes may also assist other events on the Blockchain, depending on their nature, such as managing voting events, providing protocol execution, and enforcing the rules of the Blockchain. Masternodes are usually available all the time (24/7), and they have more Random Access Memory (RAM is working memory and disappears when the computer power is turned off) than regular full and lightweight nodes. A Masternode may be compared to running a very big server on the Blockchain network.
 - Not just anyone can run a Masternode host. The host must deposit a minimum (sometimes fairly large) quantity of cryptocurrency as collateral because the power and control of operating a Masternode might be exploited. When the Masternode host violates the Blockchain's regulations (its protocol), the potential loss of collateral posted by the host acts as a deterrent and incentive for the Masternode host to act in accordance with the protocol.
- **In summary, Full Nodes** do the following:
 - Secure the network by determining whether or not Block transactions are legitimate/authentic and vote to accept or reject them.
 - Save and store approved transaction Blocks (storing Blockchain transaction history).
 - The transaction history is broadcast and disseminated by full nodes to other nodes that may need to synchronize with the Blockchain (updates on transaction history are important, among other objectives as an audit trail, confirm the existence or absence of a transaction).
 - Cause new ('minted') cryptocurrency to be issued.
- **Light or Lightweight Node**
 - After Full Node, the second-most common type of Blockchain node is the Light or Lightweight Node. The purpose of these nodes is to accommodate faster transactions and daily activities. They are also known as Simplified Payment Verification (SPV) nodes. Instead of downloading and storing the entire Blockchain, they are designed to download and store just the necessary information, i.e., Block Headers, to facilitate a Blockchain transaction. Hence, they save users a massive amount of time and storage.
- **Lightning Nodes (off-chain transactions)**

- When a Blockchain network is too busy, users can often face delayed transactions. Lightning nodes are used to reduce the latency and time delay in completing transactions to the minimum. These nodes enable off-chain transactions by establishing a connection between the network and users outside the Blockchain. They reduce the load on the Blockchain network; hence the transactions are nearly instantaneous and cost very minimal fees.
 - **Super Nodes**
 - Super nodes are a less commonly found type in the Blockchain world. They are designed to perform some specialized task. Examples include, (1) police and maintain compliance with Blockchain operating regulations of the network or (2) implement an upgrade.
 - **Miscellaneous**
 - Thousands of nodes can be active at the same time on Blockchains. Anyone in a Public Blockchain may act as a node by downloading a Blockchain's transaction history. In Permissioned or Private Blockchains, only authorized nodes are permitted to download Blockchain information. Many cryptocurrency and Blockchain enthusiasts volunteer to run nodes. They do it to contribute to the Blockchain community's development, security, and integrity, but it's also a fun pastime that makes them feel like they're a part of the project.
 - Some Blockchains have so much transaction data that running a full node demands a large capacity of RAM on a device. As a result, electronic wallet programs are used by many Blockchain cryptocurrency Users who only wish to be a User on a Blockchain and not be a full node. For example, users may broadcast transactions from their cryptocurrency electronic wallet into the Blockchain without downloading the complete Blockchain history to their smartphone.
 - Another important characteristic of a node is its **availability** (*can you see me now*). For example, an "online node" is a node that is assigned to send (be on 'autopilot' all the time) Blockchain and Block updates all across the Blockchain network consistently and always be online. In contrast, offline node, such as lightweight nodes, only needs to download the most recent copy of the decentralized Blockchain ledger every time they log onto the Blockchain network to stay in sync with the rest. This process is termed *synchronizing with the Blockchain*.
 - A single full node can potentially operate a complete Blockchain, but because it is kept on a single device (computer), it is particularly vulnerable to power outages, hackers, and systemic malfunctions. Plus this single node structure is contrary to the decentralization aspirations of Blockchain technology. The more nodes a Blockchain has, the better it can withstand disasters and resist hacking. It will be difficult for a corrupt party to wipe out all of the Blockchain data at once if the data is dispersed (decentralized and distributed) over many machines. Even if many nodes fail, it only takes one node with the whole Blockchain history to back up and restore access to all the data and keep the Blockchain functioning as open for business as usual.
- **Internet:** The worldwide web internet is the platform through which Blockchain Technology exists. While it might be odd to consider the internet as a Blockchain node, for all practical purposes it is one of the fundamental 'participants' required to facilitate Blockchain technology, hence this Guides naming the Internet as a 'node'.

Disruption vs Destruction

We all are exposed every day to 'Blackbox' technology, where a Blackbox is a device or system that processes certain input information and publishes an output that us human's understand and have a beneficial use, but we haven't a clue what is going on inside the Blackbox, though we don't need to necessarily understand. Examples of backboxes include:

- A computer modem which communicates between our computers and the internet and reports its activities in a human language we understand displayed on our computer monitor yet we have not a clue how the modem Blackbox does this;
- Pushing the start button on our cars and the car starts and takes us from point A to point B but we generally don't have a clue how the black box engine, transmission and onboard computers does this;
- Opening up our mobile phones and resetting our home thermostat with just a few touches on the phone's screen, but we do not have a clue how this Internet-Of-Things black box does this;
- Blockchain black box technology running on the internet that is connected to our computers and we haven't a clue how all this takes place.

A characteristic of a black box is a device if it fails, is merely replaced – no DIY repair fix.

The reason we don't need to understand how the black box works is because we have trusted third party experts to provide the Blackbox complex system with the right ingredients that serves a useful honest service or output to the Blackbox user, does not harm us or take advantage of us. These trusted experts that build, manage and program backboxes, are in control of the internal workings of backboxes. That is power and as I recall absolute power can corrupt.

Unfortunately, because Blackbox internals are a complex mystery to most of us, there can be malicious intentions of certain Blackbox experts who are capable of manipulating the black box mystery to their unfair advantage, harming the innocent user. Consequently, there are Blackbox checks and balance rules, regulations or standards, that must be followed (and consequences if not) and some central controlling certifying authority enforcing such rules for the protection of the innocent consumer (think Underwriter Laboratory standards for electrical safety). Equally, malicious external (and in some cases internal) third parties who have access to and understand the black box innards could also manipulate the operation of a Blackbox to the detriment of innocent users.

Blockchain is based on the concept that trust is not needed between the parties, and that the Blockchain Blackbox operates in such a way that it self-proves itself – in effect a built in electronic auto-pilot trust process without the oversight by a central control authority. This trust is created in part by cryptography (security locks) and secure automatically executed mathematical programmes without human intervention and that is why Blockchain is fundamentally a cryptographic project.

Even so, locks are made to keep honest people honest, and a dishonest person can eventually find a way to open or break a lock – though it might be challenging and messy to do so. There are no absolute locks. Blockchain Blackbox is powered by electrons who are mindless³⁹ subatomic particles that do what they are told to do. Good soldiers. Those soldiers are manipulated by complex software programmes and

³⁹ I say mindless...but are they? Given us human's are made up of molecules, which are made up of atoms and the atoms ingredients includes mindless electrons and other particles built out of quarks...it is interesting that us human's, technically made up of mindless ingredients, somehow have a mind and a consciousness. So maybe electron's do have some sort of mindedness qualities we don't know about. I suspect by now some of you may have had your minds numbed by this diatribe of wanderment against common sense.

computer algorithms, and that complexity may hide malicious manipulations of those soldiers by malicious commanding participants in the Blockchain network.

The realities of backboxes reminds us that no system, including the Blockchain network, is **absolutely** trustworthy and not without some form of risk (albeit it may be rather remote) from malicious influencers.

There comes a point where centralized authority, such as a government, may have some regulatory oversight responsibility to protect the innocent...and in some cases provide that protection even if the innocent does not want it (think mandatory vehicle seat belt enforcement - its for your own good as mom used to say). It's ok for Blockchain technology to be disruptive, its quite another if it becomes destructive, especially if the destruction is not because of the internal workings of the technology but because the peripheral use and application of the technology is misused or abused – whether by accident or malicious intent. The cross-roads between disruptive and destructive is sometimes a thin and precariously moving line, hence this Guide's focus on Blockchain for Legislators.

As much as Blockchain Developers cherish network totally void of any central oversight...the real answer as in all things meaningful in life, is that the bubble is somewhere in the middle and not at the extremes. All things in moderation I figure.

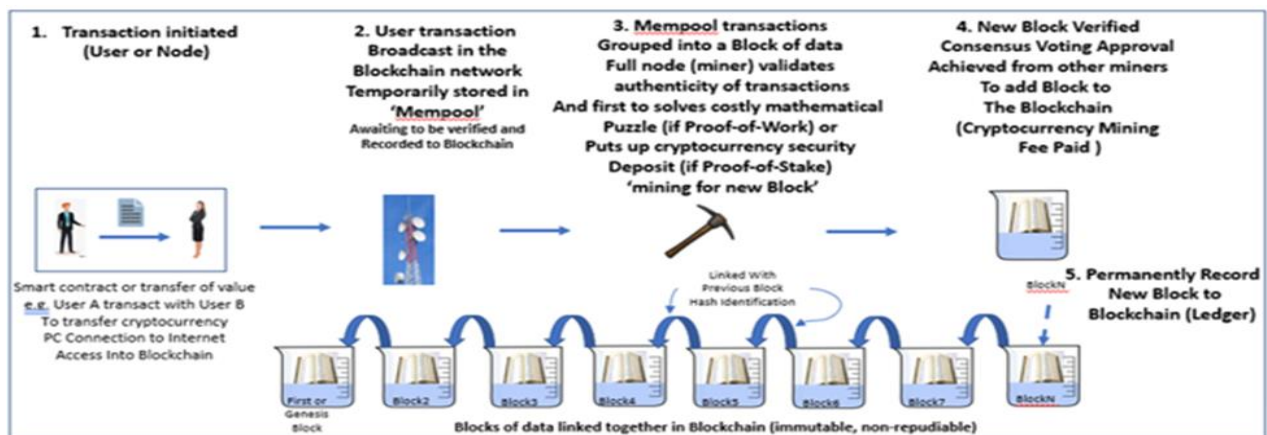
The Miracles of Blockchain Technology

Although the birth of Blockchain Technology was motivated on finding a separatist's way to transact digital currency (cryptocurrency) investment and currency business without the need for a centralized authority (such as a bank), it resulted in solving some fundamental problems regarding its decentralized trustless network structure:

- How to reach a truthful decision (consensus) among parties who are trustless strangers and equally trust and distrust each other, who are distributed throughout the world and the absence of any centralized decision making authority? (Solution to the Byzantine Generals Problem);
- How to securely, chronologically and permanently record, authenticated transaction data and information in an electronic ledger that is immutable (can't be changed) and not subject to being repudiated (parties to a transaction cannot deny the truth of a transaction once it is recorded in the Blockchain)? The chain of blocks – or Blockchain technology solution.

The details of how each of these solutions are achieved are discussed in this Guide, along with examples.

A high level description of adding new transactions to a Blockchain is illustrated in the below diagram.



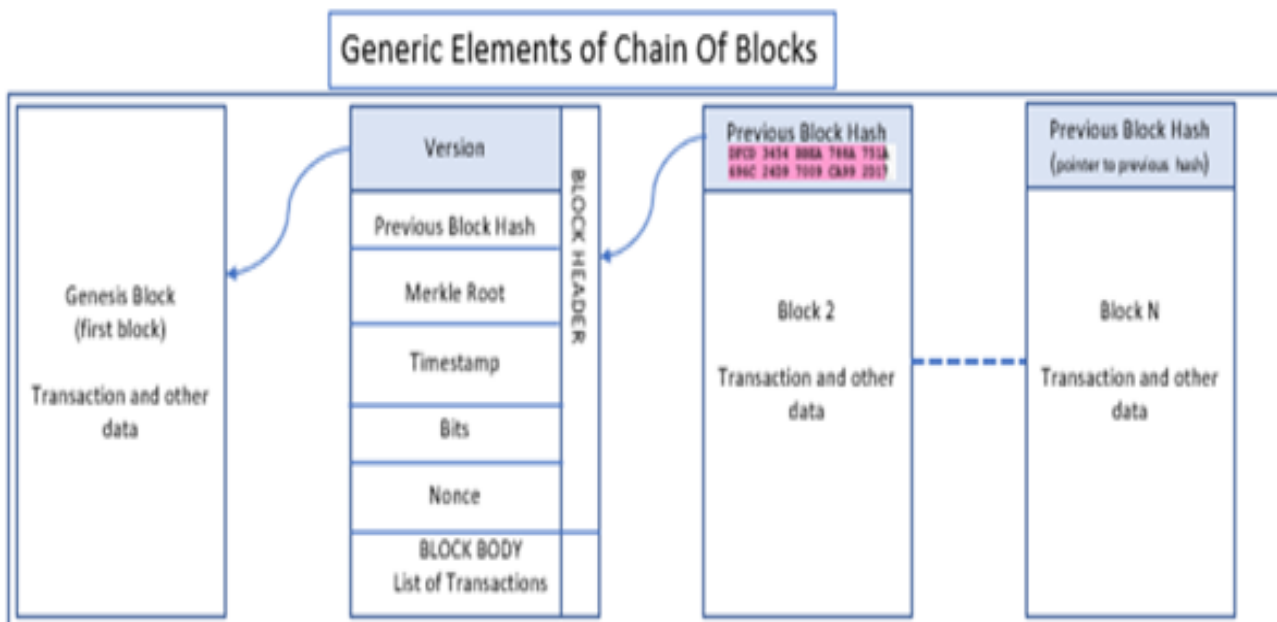
1. **Transaction initiated:** User/Node starts transaction (send cryptocurrency from Party A to Party B using respective virtual electronic digital wallets) by first initiating the transaction using PC and internet Blockchain website; digitally signs transaction with private cryptographic 'key' (key used to authenticate identity of sender and receiver)
2. **Transaction Broadcast In Blockchain:** Transaction temporarily filed in **Mempool** of Blockchain network.
3. **Find New Block:** Transaction pulled from **Mempool**, verified by miner as authentic and grouped with other verified transactions into a Block of data, miner solves costly mathematical puzzle (if Proof-of-Work scheme – a mining process) or puts up cryptocurrency deposit (if Proof-of-Stake scheme). Full node validators compete with one another to be the first to find a new Block which entitles the winner to cryptocurrency service fee. Proof-of-Work and Proof-of-Stake cryptocurrency compensated schemes incentivizes many many parties to act as validators and the Consensus Protocol voting approval process ensures truthfulness of verified transactions (the proofing schemes and the Protocol are both required to achieve objective truthful decision making in a decentralized distributed trustless network – solutions to the Byzantine Generals Problem).
4. **New Block Found:** Once the miner (Full Node) fulfills the Consensus obligations – solves a mathematical puzzle/puts up a stake, the Block is 'found', successfully 'mined', and other validators vote the Consensus obligations have been fulfilled, and the winning validator permitted to permanently record the Block in the Blockchain and receive a fee.
5. **Add New Block To Blockchain:** New created Block is permanently recorded in the Blockchain (Ledger) and the last added Block links itself cryptographically with the hash (unique identification tag or fingerprint) of the immediate prior Block.

The Blockchain of Blocks of data are distributed throughout the Blockchain network. Blockchain is generally characterized as an immutable (uneditable) trust-layer ensuring recorded transaction data is secure and preserved. Immutability is achieved by protecting the integrity of the data by applying various cryptographic techniques (hash functions, digital signatures). For Public Blockchains, 'remote' history of transactions is immutable since it is both technically and economically infeasible to reverse such history. For recent history of transactions (next 5 or 6 Blocks in a Blockchain), some minor administrative edits (called a soft fork) may occur (in an otherwise immutable or non-editable network) and are permitted (such as delays in adding new Blocks to the Blockchain, two validators (miners) validate a Block at the same time), but such soft fork 'edits' eventually self-correct themselves, such as a conflicting event being discarded once the majority of miners follow a longer chain of Blocks of data and the longer chain becomes remote and permanently immutable.

Block data are secured with a 'hash' identity (a numerical like fingerprint or serial identification number), digital signature, time stamps and other secure techniques (explained later in this Guide) resulting in each node in the Blockchain network maintaining a consistent version of Blockchain data and avoid a single point of failure and achieve traceability and transparency.

The structure of Blocks of data is important. In addition to the Blocks recording the details of each individual transaction stored in the Block, each Block is associated with a Block Header, a sort of table of contents containing certain administration information about the Block (version, previous Block hash – used to link Blocks together, Merkle root – summary identification tag of all transactions IDs rolled into a single ID stamp role, timestamp, bits – used in the Block verification process, nonce – used in the Block verification process, and Block Body) – details of all transactions in the Block. In computer speak the Block Header contains 'meta data' (data that provides information about other data. Besides useful administration information, a search for a particular transaction can be more efficiently conducted by searching the Block Header data first, then focusing on just that one Block, search for the desired transaction. This is important since any one Block can store over 2000 transactions and to search an entire Blockchain for a transaction could require enormous time and computing expense).

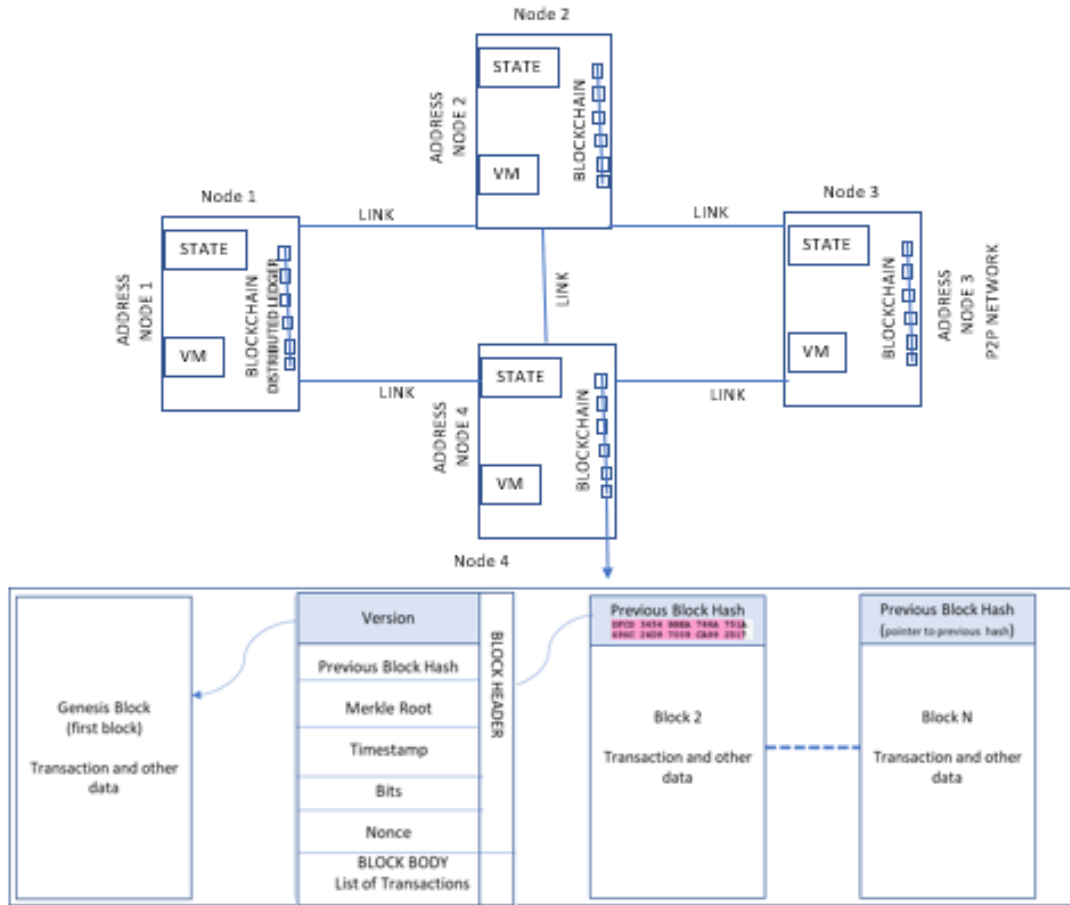
The structure of Blockchain Block Headers is illustrated in the following diagram:



- **Address:** Unique random number generated identifiers, owned by senders /receivers of transactions (a Public Key/Private Security Key or cryptographic alphanumeric number - number/symbol characters that encrypts/decrypts messages, Bob sends message to Alice, Bob sends Transaction request to a Blockchain)
- **Transaction:** transfer of value (data/information) from one Address to another (A transfers N Bitcoins to B)
- **Block:** Fixed length ('MegaByte' – 1000 characters) Block of transaction data storing multiple transactions (ex. 2000) plus Block Header information; (Authenticity validated by validators (miners) – full nodes – applying a Consensus 'voting' process)
 - **Version (1-4):** Consensus Protocol version, all Blocks must use same version else a new Blockchain
 - **Previous Block Hash:** Unique one-way (can't decrypt) fixed length randomly generated alphanumeric number hash or digest - a 'fingerprint' identity - of the previous Block; used to link Blocks into a chain and preserves authenticity and order
 - **MERKLE ROOT:** A single hash that accumulates all the individual hashes of each current Block transactions hashes and used as a one stop checkpoint to verify all the transaction data in a Block as being authentic without verifying each individual transaction
 - **TIMESTAMP:** creation of a time of a Block (in chronological order)
 - **NONCE:** Unique random generated one time use number used to in Proof-of-Work scheme to solve a complex mathematical puzzle. (A complex procedure explained in plain English in the cryptographic section of this manuscript).
 - **Block Body:** Record of transactions associated with the Block

The nonce value is critical as it is an essential ingredient in the Proof-of-Work Public Blockchain Block validation Consensus Protocol process. Its use is explained in plain English detail in the cryptographic section of this guide.

The storage of Blockchain data and associated Blocks in each Full Node validator participant computers, is illustrated in the below diagram.



Some Troubling Blockchain Questions - Odd fellows...

Some odd features of the Blockchain technology...

Can unregulated decentralized Blockchain cryptocurrency exist without regulated centralized fiat money?

- Blockchain technology is a separatists independent decentralized network. However, in regard to convertible virtual currency or cryptocurrency, such currency is valued in fiat money, so called hard currency (generally hard currency is money that is accepted world-wide as having value such as U.S. dollars). Fiat money exists through the consequence of being exclusively minted, regulated and published by centralized governments and managed by a centralized banking system. Owners of cryptocurrency hope the exchange rate of crypto to US dollars (or other fiat money hard currency) increases so one day they can exchange their crypto for higher valued centralized fiat money. Decentralized unregulated crypto depends on centralized regulated fiat money. Go fish!

Is centralized authority really any more risky than decentralized Blockchain structure?

- Blockchain decentralized structure is cited as being more secure than centralized authority because in a centralized structure, if there is a successful hacker attack, the entire centralized system is at jeopardy, whereas in the decentralized structure, a successful hacker attack will

only affect a small portion of the decentralized network allowing the rest of the network to function business as usual. Reading the newsprint suggests that there have been few successful hacker attacks on centralized systems, and this absence of security failure, most likely occurs because the centralized structure is focused and easier to manage and implement security measures. In contrast, the news media has reported some successful hacker attacks (theft of cryptocurrency) by successful hackers presumptively because, in part, of the much more complex and diversified structure of decentralized networks and hence more complex cryptographic security processes to implement. Most publications reporting on Blockchain technology include comments that more 'research' is needed to make Blockchain safer and more reliable. And research normally alludes to a circumstance still in the testing phase.

How easy is it for Blockchain participants to 'game' the system and defeat the ideal decentralized structure and revert it back to a centralized structure?

- Blockchain cryptocurrency transaction authentication (where the authentication service is provided by competing Blockchain participants known as 'miners' or 'full node validators') are required for Proof-of-Work Consensus Protocols, to commit a certain amount of resources or work (time, electricity cost, computer power), to solve complex security math problems – technically described as finding a security hash, and 'mine' for a cryptocurrency (Bitcoin) fee, as if mining for gold, which is payment if for the first miner solving the math puzzle (finding a peculiar hash value).
- Competing Proof-of-Work miners can improve their chances of earning Bitcoin for being the first to find hashes by either spending funds and upgrading their computer systems or joining other miners (so called mining pools) in a consortium who collectively work together in finding a hash (solution to a complex math puzzle) and if successful the consortium or pooled members, sharing in the Bitcoin mined fee. This pooling concept (the bigger it is, the more powerful and likelihood it is to earn the mining fee), if the pool gets too big, has been viewed as a form of centralization, contrary to the spirit of Blockchain decentralized structure, as well as mimicking an authoritarian centralized structure that limits non-pooled miners competition to mine for authentication service Bitcoin payments. Further, the dilution of participating mining parties also affects the success of solving the Byzantine Generals Problem needed to ensure authentication processes are valid and truthful, which depends on many honest Generals or individual validating nodes being active in the network.
- If a mining pool gets too large or successful, it might end up generating more than 51% of the security authentication hash rate (puzzle resolution) of the Bitcoin network, which in effect would give 51% control of the network to such miners...a form of centralization, and such 51% control results in the risk that the controlling miners could cause Bitcoins to be spent more than once (the double spending risk) since such miners control authentication or what is the truth in the network. This double spend risk is something Blockchain participants are particularly sensitive too and want (need) to avoid. There are various technical solutions and research in play to try and avoid the 51% mining pool control problem.
- Pooling of stakers in the Delegated Proof-of-Stake Consensus Protocol system is also a concern with tending toward centralization and diluting the Byzantine Generals Problem solution.

What is preferred: Elegant Simplicity or Intellectual Complexity? (Recall U.S. space gurus spent millions on developing the perfect but complex writing ink pen to work in any position in weightless outer space...the Russians used a pencil).

- Transitioning from a centralized to decentralized network system exponentially increases the complexity of the network. Consequently, elegant simplicity of problem solving (problems associated with security, consensus - agreement of what is the truth - voting by participants or nodes, protocol rule making, source code language, etc.) is difficult to achieve leaving intellectual complexity in its place. Probably one of many reasons why the vocabulary of Blockchain speak is painful to understand and explain and why the Blackbox of Blockchain is so opaque.

SPECIAL SECTION FUNDAMENTALS OF ENCRYPTION

Since Blockchain is so dependent on cryptography, this Fundamentals of Encryption special section was prepared as a user friendly guide intended to assist with the ready understanding of important cryptographic terms and processes, which will lead to a deeper understanding of what Blockchain is all about.

A big thanks to **Cloudflare** (<https://www.cloudflare.com/>) which was a useful resource used for describing and defining many of the important topics).

This Special Section is prepared in bite size Articles 1-15 covering various topics listed below.

Article 1	What is Encryption?
Article 2	Why Is Randomness and Unpredictability Important In Cryptography?
Article 3	What Is Secure Socket Layer (SSL) and Transport Layer Security (TLS)?
Article 4	What Is Hypertext Transfer Protocol (Secure, HTTPS)?
Article 5	What Are Cryptographic Keys?
Article 6	What Is Public Key Infrastructure?
Article 7	What Is a Hash?
Article 8	Bizarre Byzantine General's Problem (and The Practical Byzantine Fault Tolerance – Proof Of Work – Proof Of Stake – Delegated Proof Of Stake Solutions)
Article 9	What Is A Digital Signature?
Article 10	Generating Public And Private Keys Using The RSA And Elliptic Curve Encryption Algorithms (The Number Modulo P [N Mod P] Factoring and Elliptic Curve Discrete Logarithm Problem, Methods)
Article 11	What Is The Internet?
Article 12	What Is A Router And Modem?
Article 13	How Will Blockchain Promote the New Web3?
Article 14	What Is Quantum Computers?
Article 15	What Is Hacking All About? Or Understanding: Common Hacking Techniques; Protecting Authentication And Tampering; What Is A Brute Force Attack – Why Do It – Bruce Force Techniques; How To Protect Passwords And Cryptographic Keys

Terminology used in the Articles and helpful in understanding cryptography and encryption includes:

- **Encrypt:** means scrambling or coding Plaintext data and information into unrecognizable (babble) characters
 - (For example, the phrase: *“meet you at six”* in plaintext English, could be encrypted into unrecognizable babble as *“@89bZr 7*(%cBg XX3&^% ghj56(*&\$nz”*).
- **Decrypt:** means unscrambling unrecognizable (babble) characters back into its original plaintext data and information format.
 - (And in the above example, *“@89bZr 7*(%cBg XX3&^% ghj56(*&\$nz”* converted or decrypted back to *“meet you at six”*).
- **Algorithm:** means a precise set of computer programme instructions (coded into computer language), a step-by-step detail procedure – a recipe, that communicates with other computer software programmes, how to scramble (encrypt or cypher/cipher) and unscramble (decrypt or

decipher/decipher) data and information. Most cryptographic algorithms use mathematical functions to perform encryption processes.

- **Cypher/Cipher:** means the same thing as algorithm; or plaintext data and information that has been scrambled or encrypted into unrecognizable babble (the opposite of plaintext). Data and information is not limited to just text (letters) only, it can also be numbers, special characters such as “#”, or even graphical information.
- **Cryptographic primitives:** means the building blocks of a security protocol or system, a set of steps taken to achieve the required security goals by utilizing appropriate security mechanisms. Primitives can include: Hash primitive, symmetric key primitives, asymmetric primitives, keyless primitives, public key interface primitives, random number generator primitives, cipher primitives, digital key primitives
- **Key (digital):** means a complex sequence or string of alpha-numeric⁴⁰ characters, produced by an algorithm, that allows data and information to be encrypted (scrambled babble) or decrypted (unscrambled babble). Like a conventional metal key that locks and unlocks a lock, a cryptographic electronic key encrypts (locks) plaintext messages (converts it to scrambled unrecognizable babble to keep it secret) and decrypts (unlocks) the locked (encrypted) message back to understandable plaintext. The characters in the key are often time used in an algorithm equation to calculate certain information...hence why the digital electronic key locks and unlocks information. It is essential that the characters making up a key are chaotic, random and unpredictable – in order to avoid hackers from cracking and discovering the key. [Discussed in the Articles includes what keys look like and how they are created and used – the math of keys].
 - (An example of a alpha-numeric key might be the following string of characters “@89bZr7*(%cBg”).
- **Plaintext:** means decrypted or unencrypted original readily understandable data and information, and in this manuscript means plaintext in the English language, the opposite of Cyphertext.
- **Protocol:** a set of rules or operating procedures, the details of which vary from its application.
 - **Internet Protocol (IP)** is a protocol, or set of rules, for routing and addressing packets of data so that it can travel across Internet networks and arrive at the correct destination.
 - **Blockchain Protocols** are a set of rules and procedures used to govern the Blockchain network. The rules define the interface of the network, interaction between connected computers, incentives, kind of data, etc. The Protocols aim to address four principles: Security (keep things safe and secure), Decentralization (honor the separatists objective of no Big Brother), Consistency (objective of reliability and confidence), and Scalability (ability to grow in size with efficiency).
- **Authentication:** (from Greek, "real, genuine", "author") is the act of proving an assertion, such as the truthful identity of a computer system user; provides some degree of certainty that a given message has come from a legitimate source and the message itself is truthful and not altered.
- **Trap-door functions:** algorithm mathematical functions that are easy to calculate in one direction but almost impossible to calculate in the other (like a trap-door, easy to fall through, but tough to

⁴⁰ Alpha-numeric *numbers* (the term *number* does not exclusively refer to an arithmetic numerical value, such as 12345, but is just the naming convention given to a set or string of characters defined as a being alpha-numeric) are a string (a sequential connected set of characters, similar to a single word) of characters made up of letters and numbers and sometimes symbols such as for example “Ab34cD569z” (English language illustration). However, special non-number/letter characters are also used to create alpha-numeric *numbers*, especially when an alpha-numeric number is used as a security device, such as a password or encryption key. Examples of special characters include &, \$, @, -, %, *, and *empty space*. Thus, for example, an alpha-numeric password ‘number’ could be “Ab45%C&_d”.

get back out). The determination of digital cryptographic keys relies on this concept, such that the process of determining keys is easy to understand but nearly impossible to actually determine the key itself, which is a concept to guard against hacker attacks.

- **Prime number:** a number that is only divided by 1 or by itself, such as the number '7'. Prime numbers are regularly used as cryptographic digital keys since the key number will be a unique whole number integer and when used in cryptographic calculations, will result in unique and one of a kind results.

ARTICLE 1

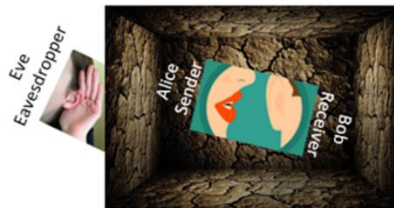
WHAT IS ENCRYPTION?

KEY TAKE AWAY: NO SYSTEM IS HACK PROOF AND NO SYSTEM IS IDIOT-PROOF.

What is encryption?

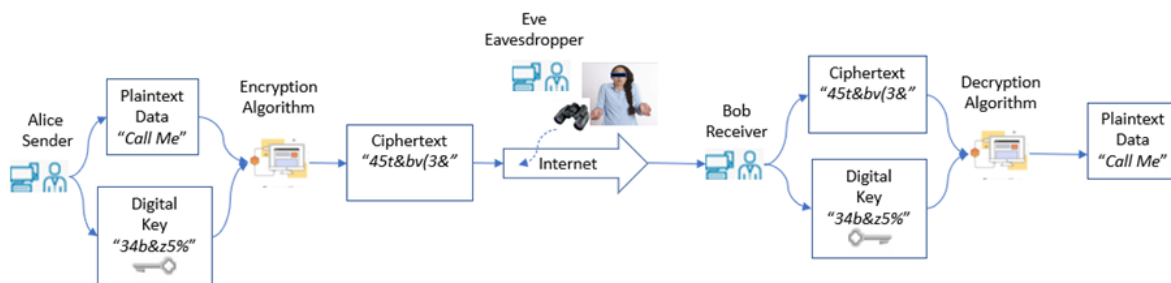
Encryption is a way of securely communicating between a sender and receiver, by scrambling (**encrypting**) **plaintext** data and information into unintelligible random looking data and information (**ciphertext**). In technical terms, it is the process of converting human-readable plaintext to incomprehensible text, also known as ciphertext. In simpler terms, encryption takes readable data and information and alters it so that it appears random.

There are many methods for encrypting data and information and keep it secure from prying eyes and ears from Eve the eavesdropper, during its transmission between Alice the sender and Bob the receiver, who has the secure ability of reversing the process or decrypting the ciphertext gibberish back to plaintext.



Encryption requires the use of a computer programme, called an **algorithm**, which is a complex mathematical function written in computer language, that is executed when Alice inputs into the algorithm programme two items (1) plaintext data AND (2) a special unique **alpha-numeric**⁴¹ 'number', similar to a password, a **cryptographic digital key** (a set of mathematical alpha-numeric values that both the sender and the recipient of an encrypted message agree on), after which the algorithm generates the random unintelligible ciphertext that is transmitted over public transmission lines, such as the internet, to Bob.

$$\text{plaintext} + \text{key} + \text{algorithm} = \text{ciphertext}$$



Although encrypted data appears random, encryption proceeds in a logical, predictable way, allowing a party that receives the encrypted data and possesses the right key to decrypt the data, turning it back into plaintext. Effective secure encryption (and a measure of that effectiveness referred to as the encryption '**hardness**') will use keys complex enough (complexity being measured how random or unpredictable they

⁴¹ See footnote 1.

are and how hard they are to crack) that a third party is highly unlikely to decrypt or break the ciphertext by brute force — in other words, by guessing the key. (Computers are good at guessing things.)

Data can be encrypted "at rest," when it is stored, or "in transit," while it is being transmitted somewhere else.

What is a (digital) key in cryptography?

A cryptographic digital key is a string of characters (letters, numbers and symbols) of specified length used within an encryption algorithm for altering plaintext data and information so that it appears random and unintelligible and converts it to ciphertext. This plaintext to cipher text process can be accomplished by many methods. Articles 5 and 6 describe some of the fundamental ways of converting plaintext to ciphertext. Like a physical key, it locks (encrypts) plaintext data and changes it to ciphertext data so that only someone with the right key can unlock (decrypt) the ciphertext data back into plaintext data. Keys are generally determined by using a random long prime numbers (a prime number is a pure integer (whole) number, such as '7', that can only be divided by 1 or itself). Once a key is determined based on the prime number, it can then be encrypted into an alpha-numeric cipher (string of numbers, letters and/or symbols and the conversion process used by an encryption algorithm such as a hash, described in Article 7) and the key cipher used with encryption algorithms as the locking (encrypting) and unlocking (decrypting) key.

What are the different types of encryption?

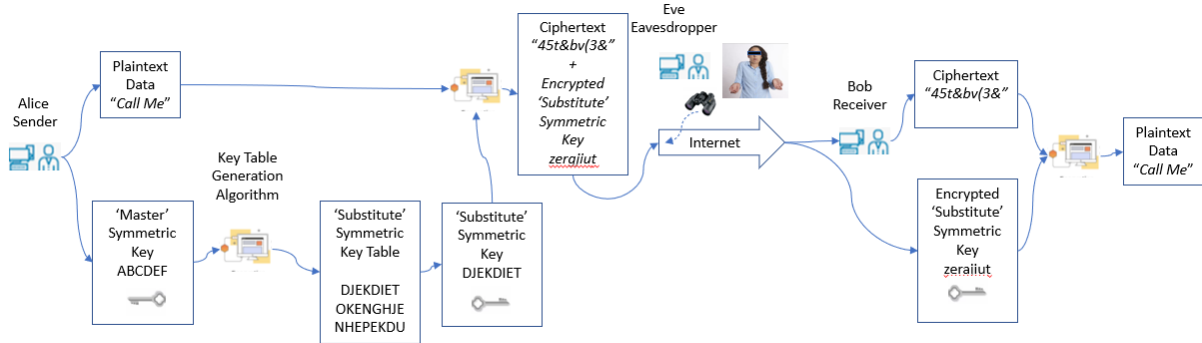
The two main kinds of encryption are **symmetric** encryption and **asymmetric** encryption. Asymmetric encryption is also known as **public key encryption**.

Symmetric Keys

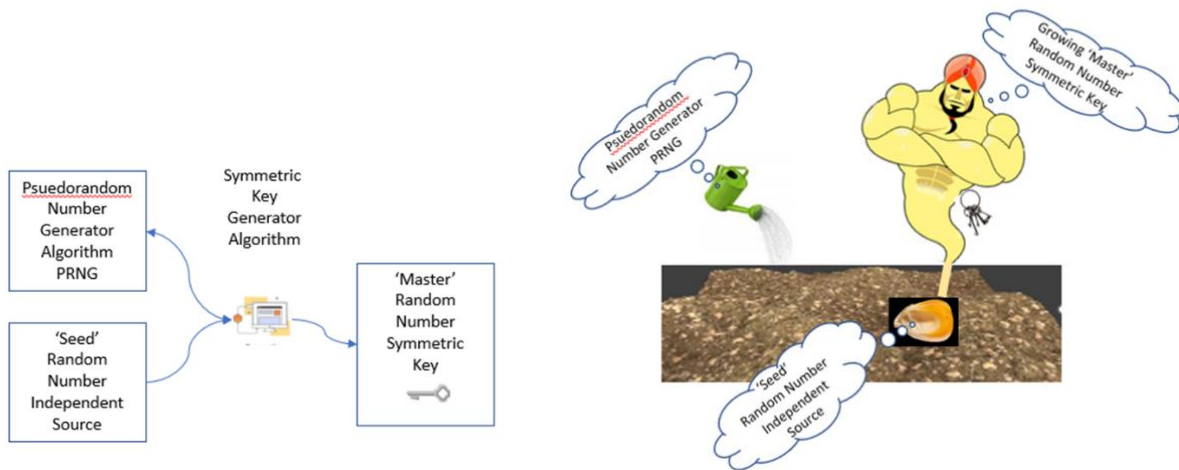
In symmetric encryption, there is only one secret private key used to encrypt plaintext to ciphertext, and all communicating parties (but not the public) share and use the same (secret) private key for both encryption and decryption, locking and unlocking the data, similar to one's home front door key. Symmetric-key algorithms have their own applications, such as encrypting data for personal use, or for when there are secure channels that the private key can be shared over.

While the symbolic use of a single symmetric key appears as if only a single secret key is used over and over again to encrypt and decrypt data, in practice it is not that simple. A symmetric key (sort of like a 'master' symmetric key) is often used first to select a different secret symmetric key from a key table (which often has more characters than the master key), and that substitute symmetric key (sometimes referred to as a **session key** or an **ephemeral temporary or static key**) used to encrypt and decrypt data. Each time the master symmetric key is used, the substitute symmetric key table is created (by an algorithm using the master symmetric key to create the table and the key table generation process known as *key set up or initialization*) and a substitute symmetric key selected from the table is used to encrypt and decrypt. (The encryption process varies with the algorithm but often involves substituting characters in the substitute symmetric key for characters in the plaintext to create an unintelligible ciphertext message, then reversing the process when decrypted. The data in a **'block' cipher** is segmented into certain size blocks, then block ciphers take exact chunks of data, encrypt them with the key table, and then take the next bite size chunk, and repeat the process until all the data is encrypted. This chunking encryption process makes the encryption process more efficient and less time consuming. An alternate encryption process is a **stream cipher**. A stream cipher creates a key that is as long as the plaintext message to be encrypted, and each byte of plaintext data is encrypted with a byte from the long key. Quite complex but easy for computers. Stream ciphers, while quick and simple, are not used if there is a lot of data to

encrypt. The sender and receiver will each need to have the symmetric key algorithm. The encryption key set up process is illustrated in the below diagram.



The master symmetric key is often determined from two random keys by adding together: (1) a random number generated by a pseudorandom number generator (PRNG) algorithm and (2) another random number, called a 'seed', obtained from another source (such as the time in milliseconds on a user's computer clock). The 'seed' is 'planted' with the PRNG and used to 'grow' a random number that eventually becomes the master symmetric key.



When substitute symmetric key is transmitted to the receiver, it is also encrypted.

The biggest risk of symmetric keys is that it needs to be shared while simultaneously keeping it secret between the using parties and protect it; the main advantage is that it can encrypt very large data files.

Asymmetric keys solve this sharing and protection risk, but cannot efficiently encrypt large data files (symmetric can be 1000 times faster than asymmetric).

Asymmetric Keys

In, **asymmetric**, or **public key**, encryption, there are two keys: one key is used for encryption, and a different key is used for decryption. The decryption key is kept private (hence the "private key" name), while the encryption key is shared publicly, for anyone to use (hence the "public key" name). Asymmetric encryption is a foundational technology for Transport Layer Security or TLS (often called by its

predecessor, Secure Socket Layer or SSL). TLS and SSL are discussed in Article 3. Asymmetric is mainly used on small files such as emails and attachments to messages.

Fundamentally, asymmetric keys use whole prime numbers (a number only divided by 1 or itself) as their starting point. A long random number is first generated from which a private key is determined, and the private key used to derive a public key. The elegance of this process is that an intruder cannot reverse engineer the public key to obtain the private key (what is called the discrete logarithmic problem, explained in Article 10 – a problem easy to calculate one way, but the reverse is essentially impossible - a trapdoor process, easy to fall in, tough to get out). Thus asymmetric ‘hidden in plain sight’ keys avoid the single key disclosure risk associated with symmetric keys. Article 10 of this section advances a deep dive, in plain English, how asymmetric public and private keys are generated.



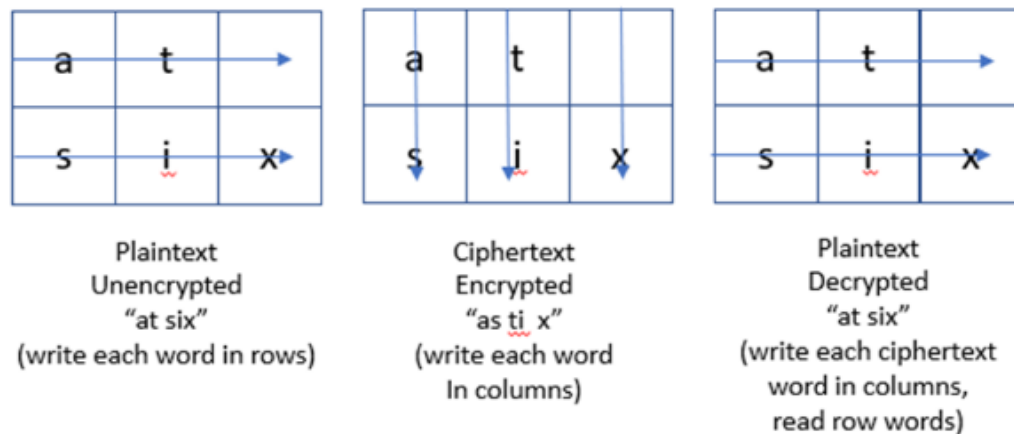
What are the different methods for encryption and creating ciphers?

Encryption Methods

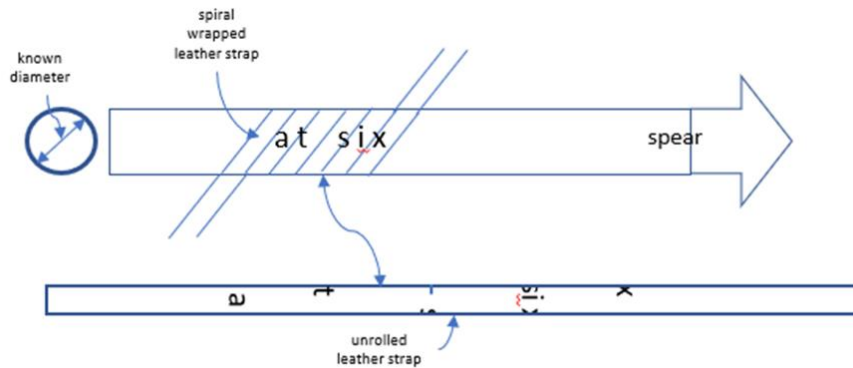
There are many techniques for encrypting (scrambling) data and information. This paper does not describe all the various techniques, but a summary of the principles of the more common simple techniques follows:

- **Concealment Ciphers (a two way cipher, encrypts and decrypts):** hiding a secret message in plain sight.
 - For example, assume an encrypted in plain sight nonsense message is written on a piece of paper for anyone to read and reads in English “*Beatles alter bizarre misty smiles mixed with joy*”. The message is hand delivered to a General awaiting instructions at what time his army is to attack a certain village. (If the nonsense message is intercepted by the enemy, it will not reveal the secret information). The General has in his possession the **key** or instructions how to decrypt (unlock/unscramble) the message from its nonsense plain sight status into understandable plaintext. The key instructions requires the General to take the third letter of the first six words of the message and create a plaintext message with an exception if the third letter is a “z” it can either be the letter “z” or a “space”, which ever makes more sense. Thus the encrypted nonsense message when unlocked/unscrambled becomes “*at six*” (try it...).
 - “*Beatles alter bizarre misty smiles mixed with joy*” (ciphertext) + *decryption key* = “*at six*” (plaintext)
- **Substitution Ciphers (a two way cipher, encrypts and decrypts):** substituting one character in a real message for another to create a fake message and reversing the process to decrypt the fake message back into the authentic plaintext message. This technique preserves the order of the characters.

- For example, the substitution cipher used, is to encrypt (scramble) the real plaintext English message by substituting the immediate next letter in the English alphabet for the real letter character in each word. The unencrypted real message “*at six*” then becomes the encrypted fake message “*bu t jy*” (abcdefghijklmnopqrstuvwxyz). The order of the characters is preserved since “*a*” will always precede “*b*”, “*t*” will always precede “*u*”, etc..
- “*at six*” + encryption key = “*bu t jy*”
- **Transposition Ciphers (a two way cipher, encrypts and decrypts):** substituting one character in a real message for another to create a fake message and reversing the process to decrypt the fake message back into the authentic plaintext message. The substitution changes the order of the letters (characters) in the message.
 - Examples include:
 - Backward writing a message (“*at six*” becomes “*ta xis*” or “*xis ta*”, depending on what decryption key instructions are used);
 - “*at six*” + encryption key = “*ta xis*”
 - Writing the message in a grid by rewriting the message based on the grid and then give the solution key (how to interpret the message) to the reader. Assume a 3 column 2 row grid, as illustrated below; write each word of the plaintext message in the rows of the grid, then use the column ‘word’ to convert the message to scrambled babble; thus “*at six*” (each word written in consecutive order in each row of the grid) becomes “*as ti x*”. The solution key to the grid is given to the reader and instructed to take each scrambled word, place it in the columns of the grid and read the real message across the rows.



- A clever transposition cipher used by the Spartan’s a long time ago to conceal secret messages in plain sight was to use a *scytale*. A scytale is a transposition cipher technique of wrapping in spiral fashion a strip of leather around a spear handle, write the plaintext message on the leather spiral in the direction of the spear shaft (the encryption key), unwrap the leather strap and the markings on the strap appear as abstract art (ciphertext). To decipher the ciphertext message, the reader who the leather strap was delivered to, needs to know two things about the decryption key: (1) the leather strap needs to be wrapped in a spiral fashion (2) around a spear with a certain size diameter.



Hash (Function) Algorithm (such as Secure Hash algorithm or SHA) “Cipher” (a one-way cipher, only encrypts) are very special and unique ciphers and explained in Article 7.

Why is data encryption necessary?

Privacy: Encryption ensures that no one can read communications or data at rest or stored except the intended recipient or the rightful data owner. This prevents attackers (hackers and cyber terrorists), advertising networks, **Internet Service Providers - ISPs** (such as Comcast or AT&T), and in some cases governments, from intercepting and reading sensitive confidential data, protecting data owner privacy.

Security: Encryption helps prevent data breaches, whether the data is in transit or at rest (stored). If a corporate device (such as a laptop computer) is lost or stolen and its hard drive is properly encrypted, the data on that device will remain secure and secret. Similarly, encrypted communications enable the communicating parties to exchange sensitive data without leaking the data.

Data integrity: Encryption also helps prevent malicious behavior such as “*on-path attacks*”. When data is transmitted across the Internet, encryption ensures that what the recipient receives, has not been viewed or tampered with on the way.

Regulations: For protecting privacy, security and data integrity reasons, many industry standards and government regulations require anyone that handles user protected data, to keep that data encrypted. Examples of regulatory and compliance standards that require encryption include HIPAA (US Health Insurance Portability and Accountability Act of 1996, protect private health information), PCI-DSS (Payment Card Industry Data Security Standard, protect private credit card data) , and the GDPR (General Data Protection Regulation (GDPR) is legislation that updated and unified data privacy laws across the European Union (EU) – protects personal private information).

What is an encryption algorithm?

Encryption algorithms are the methods (usually a complex mathematical functions executed in a computer programme) used to generate keys as well as transform plaintext data into ciphertext. An algorithm, taking plaintext input, will use the encryption key input in order to alter the plaintext data in a predictable way, so that even though the encrypted ciphertext data will appear random and unintelligible, it can be turned back into plaintext data by using the decryption key.

What are some common encryption algorithms?

Commonly used symmetric encryption cipher algorithms include:

- AES

- The Advanced Encryption Standard , also known by its original Dutch name Rijndael, is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST)
- 3-DES
 - Triple Data Encryption (Standard) Algorithm (Triple DEA), applies the DES cipher algorithm three times to each data block.
- SNOW
 - SNOW encryption cipher is primarily keeping a secret or hiding something in plain sight. The encoding scheme used by SNOW relies on the fact that spaces and tabs ‘characters’ (known as *whitespace*), when appearing at the end of lines, are invisible when displayed in pretty well all text viewing programs. This allows messages to be hidden by using such whitespace tab and space ‘characters’, without affecting the text's visual representation. And since trailing spaces and tabs occasionally occur naturally, their existence should not be sufficient to immediately alert an observer who stumbles across them. These whitespaces or snow, are hidden in plain sight.⁴²

Commonly used asymmetric cipher encryption algorithms include:

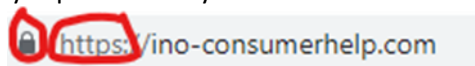
- RSA
 - RSA is one of the oldest data encryption processes that stands (**Rivest–Shamir–Adleman**) the surnames of the authors of the technique. While it is not necessary to understand the details of the technique for Blockchain, for the more adventurous, Article 10 goes into some of the detail of RSA computations.
- Elliptic curve cryptography
 - **Elliptic-curve cryptography (ECC)** is an approach to asymmetric encryption public-key cryptography based on the algebraic equation and structure of elliptic curves over finite fields or fixed regions bounded by the elliptic curve, discussed in Article 10.

What is a brute force attack in encryption?

A brute force attack is when an attacker who does not know the decryption key attempts to determine the key by making millions or billions of guesses. Brute force attacks are much faster with modern computers, which is why encryption has to be extremely strong and complex. Most modern encryption methods, coupled with high-quality passwords, are resistant to brute force attacks, although they may become vulnerable to such attacks in the future as computers become more and more powerful, especially with quantum computing. Weak passwords are still susceptible to brute force attacks.

How is encryption used to keep Internet browsing secure?

Encryption is foundational for a variety of technologies, but it is especially important for keeping HTTP (Hypertext Transfer Protocol Secure, explained in Article 4) requests and responses secure. The protocol (security standard) responsible for this is called HTTPS (A website served over HTTPS instead of HTTP will have a Universal Resource Locator or URL – the web address number - that begins with https:// instead of http://, usually represented by a secured lock in the address bar (example).



⁴² Similar to hiding in plain sight tab and space whitespace ‘characters’ SNOW encryption, is the science of *steganography*, concealing messages in other messages. Some historical techniques have involved invisible ink, subtle indentations in paper, and even tattooing messages under the hair of messengers. In this digital age, steganography provides means for hiding messages in digital audio files, in some kinds of images, and even for generating pseudo-English text which encodes the message.

HTTPS uses the encryption protocol called Transport Layer Security (TLS), explained in Article 3. In the past, an earlier encryption protocol called Secure Sockets Layer (SSL) was the standard, but TLS has replaced SSL. A website that implements HTTPS will have a TLS certificate installed on its origin server.

ARTICLE 2

WHY IS RANDOMNESS AND UNPREDICTABILITY IMPORTANT IN CRYPTOGRAPHY?

KEY TAKE AWAY: CHAOS AND RANDOMNESS ARE ESSENTIAL TO KEEP THINGS SECRET.

Randomness and unpredictability are extremely important for secure encryption. Each new digital public and private cryptographic key that a computer uses to encrypt/decrypt data must be truly random, so that an attacker won't be able to figure out the key and decrypt the data. However, computers are designed to provide predictable, logical outputs based on a given input. They aren't designed to produce the random data needed for creating unpredictable encryption keys. Thus human intervention required to obtain randomness.

A random number generator (RNG) is a computational computer or physical device designed to generate a sequence of numbers or symbols that lack any pattern, i.e. appear random. For those of you with Microsoft Excel, the Math Formula tab has a random number generator RAND. Try it. Such programs while useful are less effective than others in generating more realistic randomness.

In contrast, a more efficient pseudorandom number generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm for generating a sequence of numbers that approximates the properties of true random numbers, that appears to be statistically random (no pattern), despite having been produced by a completely deterministic (meaning a process where the input to a procedure will always result in the same output) and repeatable process.

There are many online PRNG computer programme calculators from which to determine large random prime numbers.

An example (<https://www.mobilefish.com/>):

Cryptographic Pseudorandom Number Generator input:

Number of bytes *: Max 16384 bytes.

Encoding *: Uppercase

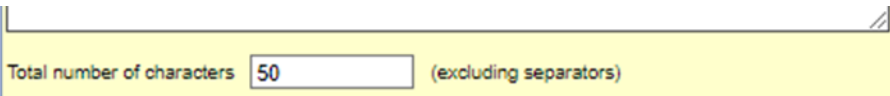
Separator *:

* = required

Cryptographic Pseudorandom Number Generator output:

Hexadecimal

```
8a11f560bdfae5df03646baa1d1ef0a2d383ed5a505ac5a468
```



Examples of physical devices used to generate a random number includes the number of times a computer user opens websites or the measured physical temperatures of a computers central processing unit or the measured sound levels in decibels of the computers cooling fan, etc..

Lava Lamp

Another physical device used to generate a random number is the lava lamp.

To produce the unpredictable, chaotic data necessary for strong encryption, a computer must have a source of random data. The "real world" turns out to be a great source for randomness, because events in the physical world are generally unpredictable.

As one might expect, lava lamps are consistently random. The "lava" in a lava lamp never takes the same shape twice, and as a result, observing a group of lava lamps is a great source for random data. Taking periodic pictures of many lava lamps and digitizing the images into numbers, is a useful pseudorandom number generator.

Lava lamps have actually been used (by Cloudflare) to determine random numbers. Surprisingly, it is not the first – a company called Silicon Graphics designed a similar system called "Lavarand" in 1996..

What does 'random' mean in the context of cryptography?

In cryptography, random does not just mean statistically random; it also means unpredictable.

Encrypted data should ideally look like totally random unpredictable data, since predictable data can be guessed. If there is any pattern – if certain values are used for encryption more than others, or if values appear in a certain order consistently – then mathematical analysis will pick up on the pattern and allow an attacker to have a much easier time guessing the key used for encryption. Essentially, if encrypted data is predictable, it might as well already be compromised.

The process of encryption itself is a predictable one: Encrypted data plus the right digital encryption key equals decrypted data, and the decrypted data is the same as it was before it was encrypted. But the encryption keys used have to be unpredictable.

To understand why unpredictability is so important, imagine two poker players: Bob always bets when he has good cards and folds (declines to match other players' bets) when he has bad cards. Alice, meanwhile, mixes up her betting strategy so that there's no discernable pattern to it: sometimes she bets when she has good cards, sometimes she contents herself with matching other players' bets, and sometimes she even bluffs by betting big when she has bad cards. When Alice and Bob enter the same poker tournament, Alice lasts much longer than Bob does, because Bob is too predictable. Opponents quickly figure out when Bob has good cards and react accordingly. Even though they can't see his cards, they can discern roughly what cards he's holding.

Similarly, even though attackers can't see the "cards" – or, the encrypted content – that's sent over a network, they can guess it if the method for concealing the content is too predictable.

Pseudorandom generators require two qualities: there must be many many data points and the data points must be random and not predictable or uniform. Randomness is critically important to protect against malicious attackers that try and guess or determine the pseudorandom number.

Why can't computers create randomness?

Computers run on logic. A computer program is based on if-then statements: If certain conditions are met, then perform this specified action. The same input into a program results in the same output every time – that called determinism. This is by design. An input should lead to an expected output – a deterministic outcome - , not an unexpected one. Imagine the chaos if a printer printed random text that was different from the text in the document that was sent to the printer, or if smartphones were to call a different phone number than the one the user entered. Computers are only useful because of their (relative) reliability and predictability. However, that predictability is a liability when it comes to generating secure random encryption keys – since randomness is not the preferred state of comfort of computers.

Some computer programs are good at simulating randomness (Excel RAND program), but not good enough at it for creating encryption keys.

How can a computer use random, real-world inputs to generate random data?

A software program called a pseudorandom number generator (PRNG) is able to take an unpredictable input and use it to generate unpredictable outputs. Theoretically, a PRNG can produce unlimited random outputs from a random input.

Such an algorithm is called "pseudorandom" and not "random" because its outputs are not actually completely random. Why is this the case? There are 2 main reasons:

1. When given the same 'seed' (starting point) to start with twice in a row, a PRNG will produce the exact same results.
2. It's difficult to prove if the results it generates will be completely random the entire time (if the PRNG runs indefinitely).

Because of reason No. 2, the algorithm continually needs new inputs of randomness. A random starting point input is known as a "cryptographic seed."

What is a cryptographically secure pseudorandom number generator?

A cryptographically secure pseudorandom number generator, or CSPRNG, is a PRNG that meets more stringent standards, making it safer to use for cryptography (since the process generates highly random and unpredictable results). A CSPRNG meets two requirements that PRNGs may not necessarily meet:

1. It has to pass certain statistical randomness tests to prove unpredictability.
2. An attacker must not be able to predict the outputs of the CSPRNG even if they have partial access to the program.

Like a PRNG, a CSPRNG needs random data (the cryptographic seed) as a starting point from which to produce more random data.

What is a cryptographic seed?

A cryptographic seed is the initial input of data that a CSPRNG starts with for generating random data. Although a CSPRNG could theoretically produce unlimited random outputs from a single cryptographic seed, it is far more secure to regularly refresh the cryptographic seed. An attacker may eventually compromise the initial cryptographic seed – and remember, a CSPRNG will produce the exact same

outputs again if it is fed the same seed, so the attacker could then duplicate the random outputs. Additionally, even the most rigorously tested CSPRNG is not guaranteed to produce unpredictable results indefinitely, all the time.

What is entropy?

In general, "entropy" means disorder or chaos (something not predictable). But entropy has a specific meaning in cryptography: it refers to unpredictability. Cryptographers can measure how much entropy a given set of data has in terms of the number of bits of entropy. The higher the entropy, the higher the unpredictability and the stronger the randomness and the stronger the encryption key.

ARTICLE 3

WHAT IS SECURE SOCKET LAYER (SSL) AND TRANSPORT LAYER SECURITY (TLS)?

KEY TAKE AWAY: SSL AND TLS ARE ALL ABOUT THE PROCESS FOR KEEPING THE INTERNET SECURE

Before discussing Secure Socket Layer and Transport Layer Security, we need some understanding of SESSION KEYS.

What is a SESSION KEY?

A session key is any symmetric cryptographic key (using a single 'substitute' key generated from a key table by a master key) used to encrypt one communication session only. In other words, it's a temporary key that is only used once, during one stretch of time, for encrypting and decrypting data sent between two parties; future conversations between the two would be encrypted with different session keys. A session key is like a password that someone resets every time they log in.

In Transport Layer Security - TLS (historically known as Secure Socket Layer "SSL"), the two communicating parties (the client and the server) generate session keys at the start of any communication session, during the TLS "handshake" (when the client and server first communicate with each other). The official Request For Comments (RFC) is a formal standards-tracking document developed in working groups within the Internet Engineering Task Force (IETF), used for managing the ongoing development of the internet, which does not actually use the words "session keys", but functionally that's exactly what they are.

What is a session?

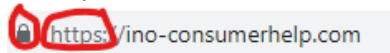
A session is essentially a single conversation between two parties over a network such as the internet. A session takes place over a network, and it begins when two devices (normally computers) acknowledge each other and open a virtual connection. It ends when the two devices have obtained the information they need from each other and send "close notify" messages, terminating the connection, much like if two people are texting each other, and they close the conversation by saying, "Talk to you later." The connection can also time out due to inactivity, like if two people are texting and simply stop responding to each other.

A session can either be a set period of time, or it can last for as long as the two parties are communicating. If the former, the session will expire after a certain amount of time; in the context of TLS encryption, the two devices would then have to exchange information and generate new session keys to reopen the connection (automatic log off and to log on again if timed out).

What is Secure Socket Layer (SSL)?

SSL, or Secure Sockets Layer, is an encryption-based Internet security protocol (a set of operating rules using cryptography regarding security of the Internet). It was first developed by Netscape in 1995 for the purpose of ensuring privacy, authentication, and data integrity in Internet communications. SSL is the predecessor to the modern Transport Layer Security (TLS) encryption used today. TLS is deemed to be more secure since SSL has been successfully attacked.

A website that implements SSL/TLS encryption security technology has "HTTPS" (HyperText Transfer Protocol Secure) in its Universal Resource Locator (URL – its www or worldwide web address) instead of "HTTP" as well as a locked lock icon, example:



What is Transport Layer Security (TLS)?

Transport Layer Security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet. A primary use case of TLS is encrypting the communication between web applications and servers, such as web browsers loading a website. TLS can also be used to encrypt other communications such as email, messaging, and voice over Internet Protocol (IP or VoIP).

TLS was proposed by the Internet Engineering Task Force (IETF), an international standards organization, and the first version of the protocol was published in 1999. The most recent version is TLS 1.3, which was published in 2018.

How does SSL/TLS work?

- In order to provide a high degree of privacy, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt.
- SSL initiates an **authentication** process called a handshake between two honest communicating devices to ensure that both devices are honest and really who they claim to be.
- SSL also digitally signs data in order to provide **data integrity**, verifying that the sent data is not tampered with before reaching its intended recipient.

There have been several iterations of SSL, each more secure than the last. In 1999 SSL was updated to become TLS.

Why is SSL/TLS important?

Originally, data on the Web was transmitted in plaintext that anyone could read if they intercepted the message. For example, if a consumer visited a shopping website, placed an order, and entered their credit card number on the website, that credit card number would travel across the Internet unconcealed.

SSL was created to correct this problem and protect user privacy. By encrypting any data that goes between a user and a web server, SSL ensures that anyone who intercepts the data can only see a scrambled mess of characters. The consumer's credit card number is now safe, only visible to the shopping website where they entered it. (Even so the process is not totally secure, if there is a malicious person working for the website vendor, that person could have access to credit card information and could wrongfully use it for themselves or the information sold to other malicious third parties for their malicious use).

SSL also stops certain kinds of cyberattacks: It authenticates web servers, which is important because attackers will often try to set up fake websites to trick innocent users and steal private data innocently sent to such fake website (a user thinking they are signed on to an authentic website when in fact it is bogus and user information entered intercepted by the bogus website author). It also prevents attackers from tampering with data in transit – by encrypting the data -, comparable to a tamper-proof seal on a medicine container.

Is SSL still up to date?

SSL has not been updated since SSL 3.0 in 1996 and is now considered to be deprecated (to withdraw official support for – not necessarily an absolute prohibition - or discourage the use of (something, such as a software product) in favor of a newer or better TLS alternative). There are several known hacker cracking vulnerabilities in the SSL protocol, and security experts recommend discontinuing its use. In fact, most modern web browsers no longer support SSL at all.

TLS is the up-to-date encryption protocol that is still being implemented online, even though many people still refer to it as "SSL encryption."

What is an SSL certificate?

SSL (TLS) can only be implemented by websites that have an SSL certificate (technically a "TLS certificate"). An SSL certificate is like an ID card or a badge that proves someone is who they say they are. SSL certificates are stored and displayed on the Web by a website's or application's server. Identification certificates are issued by trusted control authorities, attesting to the authenticity of the entity or person holding an SSL certificate.

One of the most important pieces of information in an SSL certificate is the website's public key exclusively assigned to that website. The public key makes encryption and authentication possible. A user's device views the public key and uses it to establish secure encryption keys with the web server. Meanwhile the web server also has a private key that is kept secret; the private key decrypts data encrypted with the public key. (More on this two key asymmetric encryption process in Articles 5 and 6).

As an example...assume buyer Alice enters a website owned by seller Bob. Alice purchases a product from the website and enters her credit card number as payment. In the background, what is happening is that Alice's credit card number is automatically encrypted with Bob's public key. Alice could view that public key if she wanted to, but most of us don't even know it exists. When the encrypted credit card information is received by Bob, he uses his private key (which is similarly a process happening in the background and Bob does not have to physically activate the private key) to decrypt the encrypted credit card information into plaintext so he can then process the purchase of the order. SSL/TSL protocols make all this happen quickly and seamlessly.

Certificate authorities (CA) are responsible for issuing SSL (TSL) certificates and are recognized through a Public Key Infrastructure. A Public Key Infrastructure (PKI) is explained in Article 6.

What are the types of SSL certificates?

There are several different types of SSL/TLS certificates. One certificate can apply to a single website or several websites, depending on the type:

- **Single-domain:** A single-domain SSL/TLS certificate applies to only one domain (a "domain" is the name of a website, like www.ino-consumerhelp.com).
- **Wildcard:** Like a single-domain certificate, a wildcard SSL/TLS certificate applies to only one domain. However, it also includes that domain's subdomains. For example, a wildcard certificate could cover www.abc.com, blog.abc.com, and developers.abc.com, while a single-domain certificate could only cover the first.
- **Multi-domain:** As the name indicates, multi-domain SSL/TLS certificates can apply to multiple unrelated domains.

SSL/TLS certificates also come with different validation levels. A validation level is like a background check, and the level changes depending on the thoroughness of the check.

- **Domain Validation:** This is the least-stringent level of validation, and the cheapest. All a business has to do is prove they control the domain.
- **Organization Validation:** This is a more hands-on process: The CA directly contacts the person or business requesting the certificate. These certificates are more trustworthy for users.
- **Extended Validation:** This requires a full background check of an organization before the SSL certificate can be issued.

How can a business obtain an SSL/TLS certificate? Some are free some have a fee.



What information does an SSL/TLS certificate contain?

SSL/TLS certificates include:

- The domain name that the certificate was issued for
- Which person, organization, or device it was issued to
- Which certificate authority issued it
- The certificate authority's digital signature
- Associated subdomains
- Issue date of the certificate
- Expiration date of the certificate
- The public key (the private key is kept secret)

The public and private keys used for SSL/TLS are essentially long strings of characters used for encrypting and signing data. Data encrypted with the public key can only be decrypted with the private key.

Why do websites need an SSL certificate?

A website needs an SSL/TLS certificate in order to keep user data secure, verify ownership of the website, prevent attackers from creating a fake version of the site, and gain user trust.

Encryption: SSL/TLS encryption is possible because of the public-private key pairing that SSL/TLS certificates facilitate. The generation and use of digital key pairs are discussed in Articles 5 and 6. Clients (such as web browsers) get the public key necessary to open a TLS connection from a server's SSL/TLS certificate.

Authentication: SSL/TLS certificates verify that a client/user is communicating to the correct web server that actually owns the domain. This helps prevent domain spoofing (scammers or hackers pretending to be someone else to steal data or money or to spread malware) and other kinds of attacks.

HTTPS: Most crucially for businesses, an SSL/TLS certificate is necessary for an HTTPS web address. HTTPS is the secure form of HTTP, and HTTPS websites are websites that have their traffic or flow of information and data, encrypted by SSL/TLS.

In addition to securing user data in transit, HTTPS makes sites more trustworthy from a user's perspective. Many users won't notice the difference between an http:// and an https:// web address, but most browsers tag HTTP sites as "not secure" in noticeable ways, attempting to provide incentive for switching to HTTPS and increasing security.



Once the SSL/TLS certificate is issued, it needs to be installed and activated on the website's origin server. Web hosting services can usually handle this for website operators. Once it's activated on the origin server, the website will be able to load over HTTPS and all traffic to and from the website will be encrypted and secure.

What is a self-signed SSL certificate?

Technically, anyone can create their own SSL/TLS certificate by generating a public-private key pairing and including all the information mentioned above. Such certificates are called self-signed certificates because the digital signature used, instead of being from a CA, would be the website's own private key.

But with self-signed certificates, there's no outside authority to verify that the origin server is who it claims to be. Browsers don't consider self-signed certificates trustworthy and may still mark sites with one as "not secure," despite the https:// URL. They may also terminate the connection altogether, blocking the website from loading.

Is it possible to get a free SSL/TLS certificate?

Yes, the free version of SSL/TLS shares SSL/TLS certificates among multiple customer domains.

Before an internet user and Internet Service Provider server can begin to exchange or transmit information protected by TLS, they must securely exchange or agree upon an encryption key and an encryption/decryption algorithm to use when encrypting data. Among the various methods used for generating private and public keys and encryption/decryption processes and key exchange/agreement are: public and private keys generated with **RSA** (denoted TLS_RSA in the TLS handshake protocol), **Diffie–Hellman** (TLS_DH), **ephemeral Diffie–Hellman** (TLS_DHE), **elliptic-curve Diffie–Hellman** (TLS_ECDH), **ephemeral elliptic-curve Diffie–Hellman** (TLS_ECDHE), **anonymous Diffie–Hellman** (TLS_DH_anon), [23] pre-shared key (TLS_PSK) and **Secure Remote Password** (TLS_SRP). Examples of how key generation and encryption/decryption actually works, Article10 discusses in detail RSA and elliptic curve processes.

What is the difference between TLS and HTTPS?

HTTPS is an implementation of TLS encryption on top of the HTTP protocol, which is used by all websites as well as some other web services. Any website that uses HTTPS is therefore employing TLS encryption. What does TLS do?

There are three main components to what the TLS protocol accomplishes: Encryption, Authentication, and Integrity.

- Encryption: hides the data being transferred from third parties.
- Authentication: ensures that the parties exchanging information are who they claim to be.
- Integrity: verifies that the data has not been forged or tampered with.

How does TLS work?

For a website or application to use TLS, it must have a TLS certificate installed on its origin server (the certificate is also known as an "SSL certificate" because of the naming confusion described above). A TLS certificate is issued by a certificate authority to the person or business that owns a domain. The certificate contains important information about who owns the domain, along with the server's public key, both of which are important for validating the server's identity.

A TLS connection is initiated using a sequence known as the TLS handshake. When a user navigates to a website that uses TLS, the TLS handshake begins between the user's device (also known as the client device) and the web server.

During the TLS handshake, the user's device and the web server:

- Specify which version of TLS (TLS 1.0, 1.2, 1.3, etc.) they will use
- Decide on which cipher suites (see below) they will use
- Authenticate the identity of the server using the server's TLS certificate
- Generate session keys for encrypting messages between them after the handshake is complete

The TLS handshake establishes a cipher suite for each communication session. The cipher suite is a set of algorithms that specifies details such as which shared encryption keys, or session keys, will be used for that particular session. TLS is able to set the matching session keys over an unencrypted channel thanks to a technology known as public key cryptography.

The handshake also handles authentication, which usually consists of the server proving its identity to the client. This is done using public keys. Public keys are encryption keys that use one-way encryption, meaning that anyone with the public key can unscramble the data encrypted with the server's private key to ensure its authenticity, but only the original sender can encrypt data with the private key. The server's public key is part of its TLS certificate.

Once data is encrypted and authenticated, it is then signed with a message authentication code (MAC). The recipient can then verify the MAC to ensure the integrity of the data. This is kind of like the tamper-proof foil found on a bottle of aspirin; the consumer knows no one has tampered with their medicine because the foil is intact when they purchase it.

How does TLS affect web application performance?

The latest versions of TLS hardly impact web application performance at all.

Because of the complex process involved in setting up a TLS connection, some load time and computational power must be expended. The client and server must communicate back and forth several times before any data is transmitted, and that eats up precious milliseconds of load times for web applications, as well as some memory for both the client and the server.

However, there are technologies in place that help to mitigate potential latency created by the TLS handshake. One is TLS False Start, which lets the server and client start transmitting data before the TLS handshake is complete. Another technology to speed up TLS is TLS Session Resumption, which allows clients and servers that have previously communicated to use an abbreviated handshake.

These improvements have helped to make TLS a very fast protocol that should not noticeably affect load times. As for the computational costs associated with TLS, they are mostly negligible by today's standards. TLS 1.3, released in 2018, has made TLS even faster. TLS handshakes in TLS 1.3 only require one round trip (or back-and-forth communication) instead of two, shortening the process by a few milliseconds. When the user has connected to a website before, the TLS handshake has zero round trips, speeding it up still further.

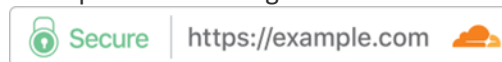
ARTICLE 4

WHAT IS HYPERTEXT TRANSFER PROTOCOL (SECURE)?

KEY TAKE AWAY: HTTP AND HTTPS ARE OPERATING RULES THAT HELP SECURE WEBSITES

What is HyperText Transfer Protocol (Secure)?

Hypertext Transfer Protocol Secure (HTTPS) is more secure and an extension of the Hypertext Transfer Protocol (HTTP). It is an Internet protocol (operating rules) that uses cryptography for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS – an encryption or secret code writing procedure) or, formerly, Secure Sockets Layer (SSL). The protocol is therefore also referred to as HTTPS over TLS, or HTTPS over SSL. The principal motivations for HTTPS are authentication of the accessed website and protection of the privacy and integrity of the exchanged data while it is in transit. A user of a website can determine if that website is secure by noting the website address which sites HTTPS and a locked lock icon (and sometimes the plaintext message “Secure”:



How is HTTPS different from HTTP?

HTTPS is not a separate protocol from HTTP. It is simply using TLS/SSL encryption over the HTTP protocol. When a user connects to a webpage, the webpage will send over its SSL/TLS certificate (which certified that the webpage is authentic which contains the public key necessary to start the secure session. The two computers, the user and the server, then go through a process called an SSL/TLS handshake, which is a series of back-and-forth communications used to establish a secure connection.

Why are Websites using HTTPS are more trustworthy for users?

A website using the more secure HTTPS is like a restaurant displaying a "Pass" from the local food safety inspector: potential customers can trust that they can patronize the business without experiencing massively negative effects. HTTPS uses the SSL/TLS protocol to encrypt communications so that attackers can't steal data. SSL/TLS also confirms that a website server is who it says it is - authentication, preventing impersonations. *Chrome and other browsers mark all HTTP websites as "not secure."*

Is HTTPS more secure, for both users and website owners?

With HTTPS, data is encrypted in transit in both directions: going to and coming from the origin server. The operating protocol keeps communications secure and encrypted so that malicious parties can't observe what data is being sent. As a result usernames and passwords can't be stolen in transit when users enter them into a form. If websites or web applications have to send sensitive or personal data to users (for instance, bank account information), encryption protects that data as well.

Does HTTPS authenticate websites?

Users of rideshare apps such as Uber and Lyft don't have to get into an unfamiliar car on faith, just because the driver says they're there to pick them up. Instead the apps tell them information about the driver, like their name and appearance, what kind of car they drive, and the license plate number. User can check these things and be certain they are getting into the right car, even though every rideshare car is different and they've never seen the driver before. Similarly, when a user navigates to a website, what they're actually doing is connecting to faraway computers that they don't know about, maintained by people

they've never seen. An SSL/TLS authenticity certificate, which enables HTTPS, is like that driver information in the rideshare app. It represents external verification by a trustworthy third party that a web server is who it claims to be. This prevents attacks in which an attacker impersonates or spoofs a website, making users think they're on the site they intended to reach when actually they're on a fake site.

Does HTTPS use symmetric or asymmetric cryptography?

HTTPS, which is HTTP in combination with the TLS protocol, uses both types of cryptography. All communications over TLS start with a TLS handshake (a communication link is opened up between a user and a server, such as logging onto a website over the internet). Asymmetric cryptography is crucial for making the TLS handshake work.

During the course of a TLS handshake, the two communicating devices will establish the session keys, and these will be used for symmetric encryption for the rest of the session (unless the devices choose to update their keys during the session). Usually, the two communicating devices are a client, or a user device like a laptop or a smartphone, and a server, which is any web server that hosts a website. (For more, see

What is the client-server model?

In the TLS handshake, the client and server also:

- Negotiate which cryptographic algorithms to use (doing so securely via asymmetric cryptography). This 'negotiation' is generally occurring automatically in the background, with the user being unaware such is going on.
- Authenticate the server's identity against its TLS certificate (using asymmetric cryptography)

ARTICLE 5

WHAT ARE CRYPTOGRAPHIC KEYS?

KEY TAKE AWAY: THE KEY TO CRYPTOGRAPHY ARE THE KEYS...

Cryptographic Keys

Cryptographic keys are a text file of random looking digital alphanumeric numbers (numbers whose characters can be made up of letters, numbers or symbols) of any length, used in cryptographic algorithms to

- encrypt plaintext data to ciphertext;
- decrypt ciphertext data to plaintext;
- create digital signatures used to authenticate the identity of an author;
- authentication (a document or identity is truthful)

The alphanumeric characters are each represented by an 8 bit size binary code (0s and 1s). The only good key is a long key.

Keys do not encrypt or decrypt anything, the algorithm does that. The information in the key is used to encrypt/decrypt, but it is not the algorithm itself.

There are enumerable methods how keys are actually used in algorithms, but functionally the characters in the key are used to determine new substituted characters or group of characters in a plaintext message and convert it to an unintelligible ciphertext. The key is need to encrypt (lock) the data and decrypt (unlock) it. The locking and unlocking description was the genesis of the use of the word key.

Two Flavors

Two types of keys are:

- Symmetric meaning only one cryptographic key is used to encrypt and decrypt data; or
- Asymmetric meaning two keys, a public key – known by the public and a private key – a secret key only known by the owner, are used to encrypt and decrypt data.

Classes of Keys (purpose)

- **Signing Keys** create digital signatures
- **Authentication Keys** authenticate computers or Users (are who they say they are)
- **Data Encryption** used in an algorithm to encrypt or decrypt data
- **Session Keys** one time use keys for any one transaction – sending encrypted data across an unsecure network such as the internet
- **Key Encrypting Keys** a key that encrypts another key for sending over an unsecure network
- **Root Key** a master key used to create other keys, often issued by a Certificate Authority

Key Retrieval

Two often misunderstood terms:

- **Key Escrow:** the actual storing of the key and the passphrase somewhere, somehow so that it can be retrieved out of storage and used when needed.

- **Key Recovery:** the process of storing a key in broken pieces (called sharding) and having the ability to recombine the pieces when needed.

Signing and Encrypting With A Key – What’s the diff?

- **Sign** a file or message when you just want to prove that the document comes from you and to ensure that it has not been changed in transit (typically, Hashes are used, see Article 7);
- **Encrypt** a file or message when you want to hide or mask it.
- **Sign and Encrypt** a file or message when you want to hide the message and prove it came from you and prove that it was not changed in transit.

Ideal Key

An ideal key is one whose characters are completely random and unpredictable (strong entropy), thereby protecting it from being discovered or ‘broken’ (cracked) by malicious parties for malicious purposes.

Keys in Modern Encryption

Historically, keys used to keep secret or hide sensitive information, were very simple, such as merely changing the order of plaintext message characters or substituting different characters for each of the plaintext characters. The key was the process or formula of how to reverse the secreting process to reveal the plaintext message. Obviously it was fundamental to protect the key itself from being discovered by unauthorized eyes. Cryptographic digital keys today are far more complex.

For instance, a website's public key might be something like:

```
04 CE D7 61 49 49 FD 4B 35 8B 1B 86 BC A3 C5 BC D8 20 6E 31 17 2D 92 8A B7 34 F4 DB 11 70 4E 49 16 61
FC AE FA 7F BA 6F 0C 05 53 74 C6 79 7F 81 12 8A F7 E2 5E 6C F5 FA 10 69 6B 67 D9 D5 96 51 B0
```

This is much more complex than say a key containing the number “4” which was interpreted by the user to substitute each character in a message with the fourth alphabet letter (example, plaintext “a” would be replaced with the letter “e”, the fourth letter in the alphabet following “a”).

Instead of simply adding or subtracting characters, modern encryption uses complex mathematical formulas known as algorithms. And instead of a simple string of random numbers for a key, modern keys are typically randomized even further.

This randomization is the case for several reasons:

1. Computers are capable of far more complicated calculations in a shorter amount of time than human cryptographers, making more complex encryption not only possible, but necessary.
2. Computers can alter information at the binary level, the 1s and 0s that make up computer data, not just at the level of individual letters and numbers.
3. If encrypted data is not randomized enough, a computer program will be able to decrypt (‘break’) it. True randomness is extremely important for truly secure encryption.

Combined with an encryption algorithm, a cryptographic key will scramble a text beyond human recognition.

In the early days a ‘computer’ was a physical person who sat at a desk and performed hand calculations. Needless to say, the term computer has evolved some since the early days.

There are two huge challenges (tension) in determining the size of digital keys (how long or how many characters they contain).

- **Keys should be as large/long as possible** since large/long keys are extremely difficult to crack because there are many more possibilities or randomness that larger character keys can be scrambled (this randomness or degree of scrambleness is referred to as its entropy – if the entropy is large, it means the key is more random and less likely to be cracked or determined; if the entropy is small, it means the key is less random – less scrambled – and higher chance it can be cracked;
- **Keys should be as small as possible** because the size of the key affects the operating characteristics of the device it is being used. Big computers can handle large keys but small devices such as mobile phone, iPad, etc., have difficulty executing large keys because of the amount of time and execution power it takes to operate large keys.

ARTICLE 6

WHAT IS PUBLIC KEY INFRASTRUCTURE?

KEY TAKE AWAY: PKI IS ALL ABOUT AUTHENTICATING THE IDENTITY OF PERSONS AND THE CONTENT OF THEIR INFORMATION SENT OVER THE INTERNET

A **public key infrastructure (PKI)** is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke (1) digital certificates (certificates issued by a recognized Certificate Authority, that the identity of a person or entity is authentic – they are who they say they are, sort of like a notary attesting to the truthful identity of a person and that data is valid and not been tampered with) and (2) manage public-key encryption. PKI is used with cryptographic asymmetric keys.

PKI is mainly used for secure transactions between companies or governmental agencies.

The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

In cryptography, a PKI is an arrangement that binds or links public keys with respective truthful identities of entities (like people and organizations). The binding or linking is established through a process of registration and issuance of certificates (sort of like being issued a passport) by a recognized certificate authority (CA). Depending on the assurance level of the identification review, this may be carried out by an automated process or under human supervision. When accomplished over the Internet, this requires using a secure certificate enrollment or certificate management protocol such as CMP (Certificate Management Protocol). [The reader of this manuscript will soon become aware that the use of the word *protocol* is about as ubiquitous as the use of the word ubiquitous – its everywhere].

The PKI role that may be delegated by a CA to a third party, to assure valid and correct registration is called a registration authority (RA). Basically, an RA performs administrative duties on behalf of the CA but does not have the signing authority of a CA and only manage the vetting and provisioning of certificates.

An entity desiring a SSL/TLS certificate (certifying their authenticity that other third parties can rely on) must be uniquely identifiable within each CA domain on the basis of information about that entity. The official Certification Authority (CA) can dedicate to a third-party validation authority (VA) that the VA, on behalf of the CA, is authorized to attest that the SSL/TLS certificate issued by the CA is legitimate and trustworthy. This administrative process reduces the work load on the CA and reduces the time to confirm verification. When a sender sends their SSL Certificate to a receiver, the receiver performs two steps to verify the sender's identity: (1) Uses sender's public key that comes with the SSL Certificate to check the sender's digital signature; (2) Verifies that the CA that issued the SSL Certificate to the sender is legitimate and trustworthy.

Capabilities

PKI provides "trust services" - in plain terms trusting the actions or outputs of entities, be they people or computers. Trust service objectives respect one or more of the following capabilities: Confidentiality, Integrity and Authenticity (**CIA**, no not the spy guys).

- **Confidentiality:** Assurance that no entity can maliciously or unwittingly view plaintext messages. Data is encrypted to ciphertext to make it secret, such that even if it was read, it appears as gibberish. Perhaps the most common use of PKI for confidentiality purposes is in the context of Transport Layer Security (TLS). TLS is a capability underpinning the security of data in transit, i.e. during transmission. A classic example of TLS for confidentiality is when using an internet browser to log on to a service hosted on an internet based web site by entering a password.
- **Integrity:** Assurance that if a third party entity changed (tampered) with transmitted data in the slightest way, it would be obvious it happened as its integrity would have been compromised. Often it is not of utmost importance to prevent the integrity being compromised (tamper proof), however, it is of utmost importance that if integrity is compromised there is clear evidence of it having done so (tamper evident).
- **Authenticity:** Assurance that every entity has certainty of what it is connecting to:
 - termed **server-side authentication** - typically used when authenticating to a web server using a password (the logging on user is authentic and proving that to the server by use of the password), or

can evidence its legitimacy when connecting to a protected service:

- termed **client-side authentication** - sometimes used when authenticating using a smart card (hosting a digital certificate and private key). The user authenticates itself by using a credit card with a chip provided by the protected service and the smart card accepted when used and the chip read or the user additionally entering a personal identification number (PIN) as additional authentication. Connecting to the real online shop and not an imitator is important...
- Good authentication is based on three requirements:
 1. Something you know (your password or passphrase – the know)
 2. Something you have (a security token – the have)
 3. Something you are (a fingerprint, facial recognition – the are)

Design

A PKI consists of:

- A certificate authority (CA) that stores, issues and signs the digital certificates;
- A registration authority (RA) which verifies the identity of entities requesting their digital certificates to be stored at and issued by the CA;
- A central directory—i.e., a secure location in which public keys are stored and indexed and available to be viewed by the public;
- A certificate management system managing things like the access to stored certificates or the delivery of the certificates to be issued;
- A certificate policy stating the PKI's requirements concerning its procedures. Its purpose is to allow outsiders to analyze the PKI's trustworthiness.

Methods of certification

There have traditionally been three approaches to getting this trust: certificate authorities (CAs), web of trust (WoT), and simple public key infrastructure (SPKI).

Certificate authorities

The primary role of the CA, sometimes referred to as Trusted Authorities (such as Verisign, CyberTrust and RSA), is to digitally sign and publish the public key bound to a given user. A user's public key is generated by using the CA's own private key, so that trust in the user public key relies on one's trust in the validity of the CA's private key. When the CA is a third party separate from the user and the system then it is called the Registration Authority (RA), which may or may not be separate from the CA. Certificates can be used to control access to computers, networks, and documents.

The term trusted third party (TTP) may also be used for certificate authority (CA). Moreover, PKI is itself often used as a synonym for a CA implementation.

Digital Certificates

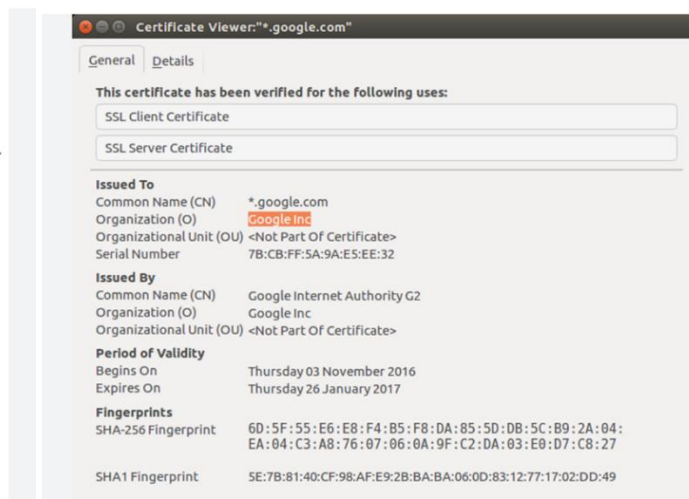
Digital Certificates content varies with the issuer but generally contains the following information:

- Certificate Version
- Serial Number
- Subject (certificate owner)
- Signature Algorithm Identifier (information about the algorithm used by the Certificate Authority)
- Issuer Name
- Not Before (start of validity date)
- Not After (expiration date)
- Key Usage (defines accepted cryptographic uses of the public key)
- Public Key
- Subject Distinguished Name (a name specifying the certificate owner)
- Subject Alternate Name Email (owner's email)
- Subject Alternate Name URI (owner's Web site URI/URL).

A Certificate Authority approved and signed Digital Certificate is called a **root signed** certificate.

Examples of certificates:

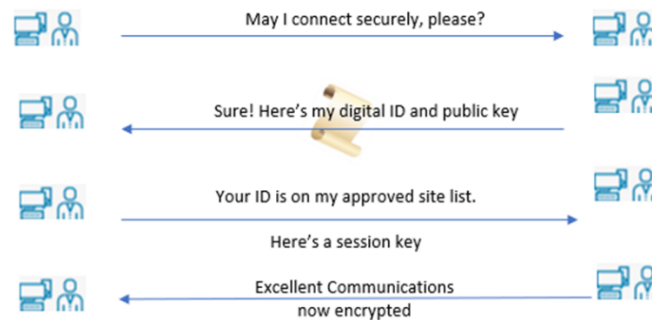
```
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 1 (0x1)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
  OU=Certification Services Division,
  CN=Thawte Server CA/Email=server-certs@thawte.com
  Validity
    Not Before: Aug 1 00:00:00 1996 GMT
    Not After : Dec 31 23:59:59 2020 GMT
  Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
  OU=Certification Services Division,
  CN=Thawte Server CA/Email=server-certs@thawte.com
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:d3:a4:30:6e:c0:ef:56:0b:e6:c1:5d:b6:ea:0c:
      68:75:47:a2:ea:c2:da:04:25:fc:a8:f4:47:51:da:
      85:b5:20:7d:9d:86:1e:0f:95:c9:e9:08:61:f5:06:
      6d:30:6e:15:19:02:e9:52:c0:62:b1:4d:99:9e:21:
      6a:0c:44:30:cd:fc:bc:c3:64:09:70:c5:fe:b1:0b:
      29:b6:2f:49:e0:3b:d4:27:04:25:10:97:2f:e7:90:
      6d:c0:25:42:99:d7:4c:43:0e:c3:f5:21:6d:94:9f:
      5d:c3:58:a1:e0:e4:d9:5b:b0:b8:de:b4:7b:df:36:
      3a:c2:b5:66:22:12:d6:87:0d
    Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Basic Constraints: critical
    CA:TRUE
  Signature Algorithm: md5WithRSAEncryption
  07:fa:4c:69:5c:fb:95:cc:46:cc:05:03:4d:21:90:0e:ca:d9:
  a0:04:49:1a:c6:da:51:e3:69:70:6c:04:01:11:a1:1a:c0:40:
  3e:59:43:7d:4c:95:3d:a1:0b:b7:0b:62:96:7a:75:0a:dd:86:
  4e:4e:9e:40:0b:08:cc:32:74:b9:6f:00:c6:e3:d3:44:0b:d9:
  8a:6f:9a:29:9b:99:18:28:3b:d1:a9:40:28:9a:5a:3c:d5:b5:
  c7:20:1b:0b:c1:cd:ab:8d:c9:f1:d9:c2:9c:2e:59:a0:da:b9:
  b2:75:1b:f6:42:f2:ef:c7:f2:10:f9:09:bc:a3:ef:0a:23:2e:
  70:47
```



How Authentication Systems Use Digital Certificates

- A Digital Certificate is generated by a Certificate Authority and stored in a file (on the hard drive of the user's computer), smart card or token. (Inserting a smart card into a client machine, activates the certificate).

- The Digital Certificate along with its public key are added to the user's computer network key server.
- An initial connection of a user to the network, will cause the Digital Certificate to first be initiated but not the public key.
- The authentication system's network server confirms the Digital Certificate is valid by checking with the network key server to see if the Certificate is in the key server's list of trusted entities.
- The connection is dropped if the Certificate check fails.
- The authentication server creates a session key, encrypts it with the public key of the Certificate and sends the encrypted information back to the user.
- The user then uses their private key to decrypt the information received from the server and extract the session key.



Certificate revocation

Authorities in the Web PKI provide revocation services to allow invalidation of previously issued SSL/TLS certificates. Most revocation statuses on the Internet disappear soon after the expiration of the SSL/TLS certificates, since such certificates have an expiration date. Revoked Digital Certificate Serial Numbers are posted on the Certificate Revocation List (CRL). This list can be searched to verify the authenticity of a Certificate.

Issuer market share

Although the global [TLS] ecosystem is competitive, it is dominated by a handful of major CAs — three certificate authorities (Symantec, Sectigo, GoDaddy) account for three-quarters of all issued [TLS] certificates on public-facing web servers.

Web of trust

An alternative approach to using a classic CA to issue SSL/TLS certificates of authenticity, to the problem of public authentication of public key information, is the **web-of-trust** scheme, which uses **self-signed certificates** and **third-party attestations** of those certificates. The singular term "web of trust" does not imply the existence of a single web of trust, or common point of trust, but rather one of any number of potentially disjoint "webs of trust". Examples of implementations of this approach are PGP (Pretty Good Privacy) and GnuPG (an implementation of OpenPGP, the standardized specification of PGP). Because PGP and implementations allow the use of e-mail digital signatures for self-publication of public key information, it is relatively easy to implement one's own web of trust.

One of the benefits of the web of trust, such as in PGP, is that it can interoperate with a PKI CA fully trusted by all parties in a domain (such as an internal CA in a company) that is willing to guarantee certificates, as

a trusted introducer. If the "web of trust" is completely trusted then, because of the nature of a web of trust, trusting one certificate is granting trust to all the certificates in that web. A PKI is only as valuable as the standards and practices that control the issuance of certificates and including PGP or a personally instituted web of trust could significantly degrade the trustworthiness of that enterprise's or domain's implementation of PKI. It's all about trust and how it was established.

Simple public key infrastructure

Another alternative, which does not deal with public authentication of public key information by a CA, is the **simple public key infrastructure** (SPKI). SPKI does not associate users with persons, since the key or important objective is what is trusted, rather than the personal trust of a person. SPKI does not use any notion of trust, as the **verifier is also the issuer**. This is called an "authorization loop" in SPKI terminology, where authorization is integral to its design. This type of PKI is specially useful for making integrations of PKI that do not rely on third parties for certificate authorization, certificate information, etc.; a good example of this is an **air-gapped network in an office**. An **air gap, air wall, air gapping or disconnected network is a network security measure employed on one or more computers in a local network to ensure that a secure computer network is physically isolated from unsecured networks**, such as the public Internet or an unsecured local area network. It means a computer or network has **no physical or wireless network interface controllers connected to other networks**, with a physical or conceptual air gap, analogous to the air gap used in plumbing to maintain water quality.

History

Developments in PKI occurred in the early 1970s at the British intelligence agency, Government Communication Headquarters - GCHQ, where important discoveries were made related to encryption algorithms and key distribution. Because developments at GCHQ are highly classified, the results of this work were kept secret and not publicly acknowledged until the mid-1990s (plus the discoveries were not readily useable due to the lack of then technology to do so).

Uses

PKIs of one type or another, and from any of several vendors, have many uses, including providing public keys and assigned ownership to user identities which they are used for:

- Encryption and/or sender authentication of e-mail messages;
- Encryption and/or authentication of documents;
- Authentication of users to applications (e.g., smart card logon, client authentication with SSL/TLS);
- Bootstrapping secure communication protocols, such as Internet key exchange (IKE) and SSL/TLS;
- Mobile signatures, electronic signatures that are created using a mobile device and rely on signature or certification services in a location independent telecommunication environment;
- Internet of things - IoT requires secure communication between mutually trusted devices (such as controlling one's thermostat from their mobile phone). A public key infrastructure enables devices to obtain and renew certificates which are used to establish trust between devices and encrypt communications using TLS.

ARTICLE 7

WHAT IS A HASH?

KEY TAKE AWAY: A HASH IS STRING OF UNINTELLIGIBLE CHARACTERS AND IS A FINGERPRINT AND DIGEST OF SPECIFIC INFORMATION, IF YOU HAVE THE FINGERPRINT YOU KNOW YOU ARE DEALING WITH TRUTHFUL UNALTERED INFORMATION

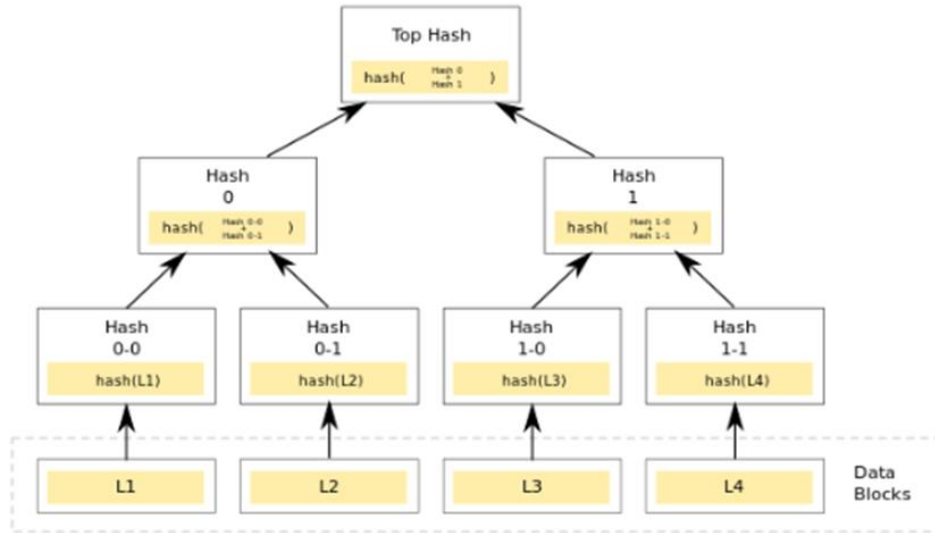
Hash (Function) Algorithm (Secure Hash algorithm or SHA) “Cipher” (a one-way cipher, only encrypts, does not decrypt to convert the Hash back to the plaintext message). Hash⁴³ functions originated from a need to compress data (sort of like zipping Microsoft excel spreadsheet files) in order to reduce the amount of computer memory required to store large files. *Hash* is in reference to a very special and unique cryptographic technique, an algorithm based on a mathematical function that garbles data and makes it unreadable. Hashing algorithms are **one-way programs**, so the hash can't be unscrambled and decoded back to the original unencrypted data by anyone else. Hashing fundamentally does two things

- (1) Protects data at rest (when saved and stored), when saved remains unreadable, and
- (2) Helps prove that data is authentic and isn't adjusted or altered after first published.

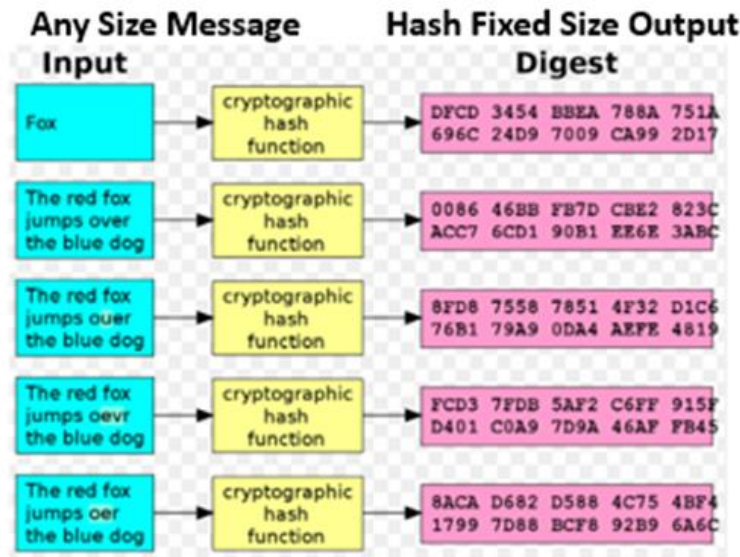
A hash is...

- a single fixed length (a fixed number of binary code bytes of information, such as 64 byte length);
- string of random looking alpha-numeric characters (numbers and letters and symbols);
- that is a one-of-a-kind representation, fingerprint (like a serial number) or digest (compact way of illustrating the data and information);
- of a message (data or information) of any arbitrary size or length.
 - If a message is **too short** (not as long as the fixed character hash) '**padding**' is used to adjust (add characters and increase the size) the length of a block to 512 bits if it is smaller than the required Block size of 512 bits.
 - If a message to be hashed is **very long**, the hashing algorithm will sub-divide the message into fixed number of character units or blocks called '**parsing**', hash each block, and if the number of those hashed blocks is still too large, the hashing of those hashed blocks of data and information will continue until only one hash digest number is developed. (This tiered multi-hashing is referred to as a **Merkel Tree**, illustrated in the below diagram).

⁴³ The origin of the word *hash*, in reference to its use in converting data into a fixed length size unintelligible scrambled digest or fingerprint substitute of the data, is from the conventional definition of the word, (a verb) 1650s, meaning "to hack, chop into small pieces," from French *hacher* "chop up" (14c.), from Old French *hache* "ax" (see hatchet). Hash browns (1926) is short for hashed (scrambled) browned potatoes (1886), with the -ed omitted, as in mash potatoes. A hash is often in reference to a food dish made from leftovers all scrambled together consisting of chopped meat (such as corn beef), potatoes, and fried onions.



- In contrast, since the hash has a fixed character length, if the hashed **message is too short** (its binary code character length is less than the fixed length of the hash) then there are arbitrary characters supplied from the hash algorithm, tacked ('padded') onto the end of the message to increase its character length to the same minimum size as the hash, then this adjusted message is converted to a hash.
- The hash is like a fingerprint or serial number (always the same for any one message). Any size message can be represented by determining its hash digest or fingerprint (by using a hash generator algorithm computer programme). The purpose of a hash is used to confirm the authenticity or prove the integrity of data and information (whether or not it has been encrypted) – is it the real thing? **How does the hash do this to authenticate a message?**
 - First, a hash of the authentic real message is generated and that original hash saved to a data base or library of hash numbers.
 - If a receiver of a message wants to confirm the message is authentic and exactly the same as the original message, the receiver will generate a hash of the message , and then compare this generated hash to the original hash, character by character, to determine if it EXACTLY matches. If the message is changed even in the slightest, such as only one letter capitalized when it should be lower case or an extra space added between words, then the recalculated hash will be totally different than the original hash (a unique feature of hash algorithms – small change in a message results in huge change in the generated hash) and if different, a red-flag to the reader than the data or information has been changed and suspect and possibly not authentic.
 - The reason why the hash is a one-way hash is because the hash cannot be decrypted or converted back to the original data and information. It is only used as a one-way fingerprint of the data and information. An example of hash characteristics and what they look like is illustrated in the following diagram.



- If the information or data is desired to be encrypted or scrambled to unintelligible gibberish cyphertext and when received, decrypted back to the original plaintext, then a suitable two-way encryption or cipher cryptographic algorithm needs to be used in addition to the fingerprint hash, since the hash cannot decrypt the underlying data and information.
- Many hashing algorithms exist, and they all work a little differently. But in each one, data is entered into the algorithm, and the program alters it to a different one-way encrypted (hash) form.
- All hashing algorithms are:
 - **Mathematical.** Strict rules underlie the work an algorithm does, and those rules can't be broken or adjusted.
 - **Uniform.** Choose one type of hashing algorithm, and data or information of any character count, put through the system will emerge at a length predetermined by the program.
 - **Consistent.** The algorithm does just one thing (compress data, create a fixed length garbled digest representation of the data) and nothing else.
 - **One way.** Once transformed by the algorithm, it's essentially impossible to revert the hash to the original plaintext data.
- **Practical uses of hash includes:**
 - **Password storage.** If plaintext username/password were stored in a database, those records are at risk of being discovered by a hacker. Instead of storing plaintext data, a hash is determined for the data and the hash stored in the database. If a hacker successfully steals the hash data, they will have no way of using it. If a honest user submits their username/password to a system for access, that data is converted to a hash and if it matches a hash in the database, entry allowed, else entry is denied.
 - **Digital signatures.** The author of a message, signs the message with a digital signature that is digitized. The alpha-numeric digital signature is converted to a hash. When the encrypted message is sent and received, the receiver can convert the digital signature of the received message to a hash and compare that hash with the original digital signature hash, and if a match, the signature is authentic, and if not, the signature is not authentic and the authenticity of the message is suspect. A tiny bit of data proves that a message wasn't modified from the time it leaves an outbox and reaches a recipient's inbox.

- **Document management.** Hashing algorithms can be used to authenticate data. The author of the document uses a hash to secure the document when it's complete as well as encrypt the data. The hash works a bit like a seal of approval. A recipient of the data can generate a hash of the received data and compare it to the original hash. If the two match, the decrypted data is considered genuine. If they don't match, the document has been changed and suspect.
- **File management.** Because of the unique characteristic of hashes being comparable to a one-of-a-kind serial number or fingerprint, hashes can be used to index large volumes of data, identify files in large data bases (look for matching hashes), and delete duplicates – assuming the duplicates are exact copies and no differences in content – characters, spaces, punctuation, etc. all the same (look for duplicate hashes). If a system has voluminous files, using hashes can save a significant amount of time in regard to the administration of such files. Searching for a hash is quicker than searching large data files. See <https://byjus.com> regarding different methods for using a hash to search for data files.
- **Common hashing algorithms include:**
 - **MD-5 (Message Digest Algorithm, 5th generation)** . This is one of the first algorithms to gain widespread approval, however, hackers have discovered how to decode the algorithm, and they can do so in seconds. Most experts feel it's not safe for widespread use since it is so easy to tear apart.
 - **RIPEMD-160.** The RACE Integrity Primitives Evaluation Message Digest (or RIPEMD-160) is considered secure. [RIPEMD stands for “RIPE Message Digest,” where “RIPE” stands for “RACE Integrity Primitives Evaluation” and where “RACE” stands for “Research and Development in Advanced Communications Technologies in Europe”—a nice example of a *recursive* abbreviation (repeated application of a rule, definition, or procedure to successive results and condense a long string of words to say just one word – and hope you don’t forget the last word needed to recreate the original long wordy message)].
 - **SHA (Secure Hash Algorithm).** Algorithms in the SHA family are considered slightly more secure. The first versions were developed by the United States government, but other programmers have built on the original frameworks and made later variations more stringent and harder to break. In general, the bigger the number after the letters "SHA," the more recent the release and the more complex and secure the program. SHA-1 is now considered unsecure. SHA-256 very secure.
 - **Whirlpool (a name assigned by the author’s of the hash).** In 2000, designers created this algorithm based on the Advanced Encryption Standard - AES. It's also considered very secure.
 - **The U.S. government** may no longer be involved in writing hashing algorithms. But the authorities do have a role to play in protecting data. The Cryptographic Module Validation Program, run in part by the National Institute of Standards and Technology, validates cryptographic modules. Companies can use this resource to ensure that they're using technologies that are both safe and effective.
- **Hash generator resources:** There are many hash calculator resources, especially through the internet, available to generate hashes. An example is the All Hash Generator (<https://www.browserling.com/tools/all-hashes>), type in the message and various hashes are generated.

The brown cow ate green grass.

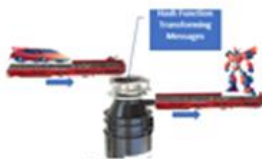
Calculate Hashes Copy to clipboard (undo)

NTLM	4B160073972118A2B0F497FEDB7905D9	MD2	f860cb7a053a5039fb22c7023d07fc7	MD4	d070983d4b5aa98cb1c6cfff14a4cd4f
MD5	9525bf472cddb5141566b199a87a75fb	MD6-128	53c1237435c5606cbad79a8f3b094e08	MD6-256	bb9271fbbf1119ea4c27b8c47895957ef462885f50
MD6-512	b06a1cc83ab63554f3888608a44861c40a921bd2	RipeMD-128	39375a85b6eaa1259f052a1afc0e7c29	RipeMD-160	04a983b679f15e5ccc871ae302f146196c7f6386
RipeMD-256	ceded4f932814994d22ece158536cb234d127dbee	RipeMD-320	576ba89b363d61293c286b37c77b708a7c92084c	SHA1	c045c30ae71b51a4fa07639b93fa796abbaad4b8
SHA3-224	215748b10745a814a2c9dfc25c74deaaa33c30e5	SHA3-256	7f246baeafde1435d32ae7d2b99fe12ba99b93ae	SHA3-384	7fbd0b6ca4bc1fc9b507c7a5cef247cf1f9e1e3ccc
SHA3-512	96623363aeabb2408959ed86c09791789b2b363f	SHA-224	71a7134b1fe7a8d24a07fa6ba2a2d3f5c3279973e	SHA-256	691a80250aaf29c4e282543bd94322c52d01176f
SHA-384	30b6b8fa5513181bdc93ced1679c34dbc60d5c0b	SHA-512	d1ef3a6e86e42ed00851a3ece74d8da973f9b9ecc	CRC16	74be
CRC32	fb319704	Adler32	a7040acc	Whirlpool	e7ab4ec1d9cbaabfa5be6f73ab4c4c1d8f7d8e665

- An efficiently drafted hash function has four characteristics:
 1. Computationally Efficient (fast/low cost);
 2. Collision Resistant (identical hash function output will not occur for different inputs; can't have the same hash collide with itself);
 3. Output Does Not Reveal Any Input Characteristics
 1. Can't reverse engineer the hash to reveal the message that was hashed
 4. Output Looks Random/Chaotic/Well Distributed/Unrelated – not predictable.

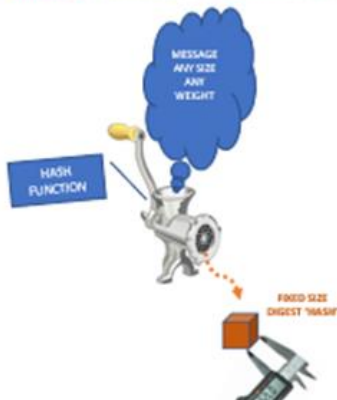
Hash Function

- What is (Secure) Hash Function (SHA)?
 - Mathematical functions/formulas (techniques) that are fundamental building blocks (a coding tool) used in cryptographic algorithms and protocols to achieve information security. A digital signature (like a fingerprint).
 - 'Transforms' information to a secure code identification number (a serial number)



- What is an algorithm?
 - A step-by-step orderly method of accomplishing some task/desired outcome (an operating / instruction manual / a recipe / a formula).
- What is a protocol?
 - Rules describing how computers communicate with each other – how information is sent/received and how it is structured.

HASHING DIAGRAM



Hash Basics

- 'Hash' – think meat grinder: input a steak, out comes random looking ground beef – that is used to identify the same steak.
- Hash Function mechanics...
 - Input arbitrary length (bits) of any message (information/data)
 - If the message size (bits) is shorter than the fixed output size (bits), the message is 'padded' to increase its size
 - Put the message through a hash function computer algorithm (a mathematical 'black box' transformer using mathematical formulas);
 - The hash function 'black box' transforms the information into a hash (think chopped up meat, potato, onion mix) and outputs a single fixed length (fixed bits) random/chaotic looking 'code' (a tag, digest, fingerprint or hash); the digest is a substitute for the input but totally different in structure and looks.
 - Always get the same hash output if the exact same input information is input;
 - Any change in the input (capitalize just one lower case letter) will result in a totally different output hash.
 - The digest/hash CANNOT be reconverted/transformed back into the original input message. A one-way (encrypt only) cryptography process (can't decrypt the hash back to the original message)

ARTICLE 8

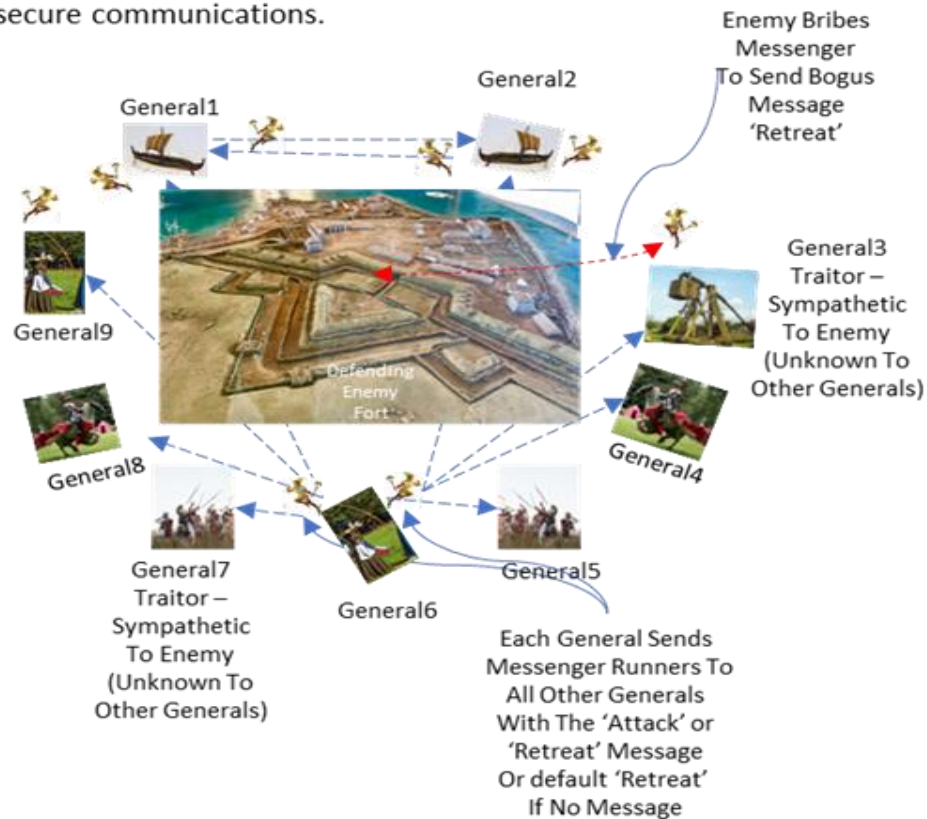
THE BIZZARE BYZANTINE GENERAL'S PROBLEM AND THE PRACTICAL BYZANTINE FAULT TOLERANCE

– PROOF OF WORK – PROOF OF STAKE – DELEGATED PROOF OF STAKE
SOLUTIONS

KEY TAKE AWAY: HOW TO OBTAIN A TRUTHFUL DECISION IN AN ENVIRONMENT OF UNCERTAINTY.

The Byzantine Generals Problem (illustrated in the below diagram) is all about an uncertain military decision making problem regarding:

The different attacking armies
only win if they all attack at
the same time, but they have no
secure communications.



- A group of Byzantine Generals (9 in total in the example diagram, 2 of which are traitors sympathetic to the enemy and one bribed messenger) surrounding an enemy fort,
- Each leading its assigned division of soldiers or sailors,
- Separated by long distances, and
- Communicating between each other only by using messengers running on foot,
- The Problem: How can the Generals collectively and truthfully assess the truthfulness of messages received from other Generals, whether to Attack an enemy fort or Retreat?

- The General's collective decision is affected by -
 1. ALL of the General's must Attack together to win, less than all means defeat, and
 2. The General's do not know if any of the messages they receive from other Generals are truthful or not, since the message could have been sent by an unknown traitor General or a messenger intercepted in route and bribed by the enemy to deliver a bogus message.
- Consequently, any of the received messages have a 50/50 chance of being truthful, Attack or Retreat. So what does a General do or believe?

2 of the 9 Generals are traitors and sympathetic to the defending Byzantine enemy and one of the messengers from one of the other 7 loyal Generals has been bribed – thus three of the messages associated with three Generals are not from a loyal source.

- 6 of the messages are truthfully sent from 6 loyal Generals, and those General's messengers will truthfully report to the other General's their loyal General's preferred action, either Attack or Retreat, and will not report conflicting actions (i.e., either the message sent by a loyal General to all the other General's will all read 'Attack' or all read 'Retreat', no mixed messages and if no action is reported, it is assumed that General's default action is 'Retreat').
- When a message is read by a receiving General, the receiving General does not know if the messenger message is truthful, there is a 50/50 chance the message is truthful, 'Retreat' or 'Attack'.
- The 2 traitor non-loyal Generals can have their messenger report any message the traitor General wants: 'Attack' or 'Retreat' – and even different messages sent to different Generals - or no message at all (no message is a default 'Retreat' message). If the 2 traitor General's as well as the one bribed messenger, all send 'Retreat' messages to the other Generals, then each of the 6 loyal General's would record 6 'Attack' messages and 3 'Retreat' messages. Since each General does not know which of the messages are truthful (other than their own), each received message by any of the Generals has a 50/50 chance of being an honest message. Each General received message can range at the extreme from all 'Attack' or all 'Retreat', or 50% 'Attack' and 50% 'Retreat' or combinations in-between, and **no clear consensus decision what to do – the Byzantine Generals Problem!**
- The uncertain consensus dilemma remains even if all Generals are loyal and no messenger has been bribed, since all messages will still be suspect as being truthful and 50/50 chance of the message being truthful to Attack or Retreat, which results in the same no clear consensus decision.

The Byzantine Generals Problem is a **game theory problem** that describes how difficult it is for dispersed parties in a decentralized network (such as parties dispersed around the world and connected to each other through the internet and each signed onto a trustless Public Blockchain website) to reach a consensus without the help of a trusted central authority (such as a bank). **Game theory is a framework for thinking about social events with competing actors, and** is about social circumstances among competing participants and produces optimal decision-making of autonomous (act independent of each other) and competing agents.

How can members of a decentralized network agree on the truthfulness of a transaction when no one single centralized authority can verify the identities of other members– **how to make an informed decision in light of uncertainty?**

Some Game Theory examples:

- A card game of poker (competing participants – the players; acting independent of each other; and the game theory being to take into account the probabilistic arrangement of the playing cards among the players and how each player tries to optimally determine if they have a winning hand – including the occasional event of pure bluffing, with a player’s intent of confusing the other players and derail any interpretation of a player displaying characteristics that tips the other players of a player’s hand – such as a player always folding if a bad hand – but in some cases, staying in with a bluff and play the bad hand), or
- In the case of the Byzantine Generals Problem, how independent competing participants – (the Generals or Full Node validators or miners in a decentralized trustless Public Blockchain) – optimally find a decision mechanism of solving for the truthfulness of -
 1. the Generals determining to collectively Attack or Retreat, when the decision is based on receiving messages from the other Generals to attack or retreat, that may or may not be truthful or
 2. Full Node validators or miners voting and reaching consensus to approve the permanent recording of information onto the trustless Public Blockchain, when the approval is based on receiving trustless voting messages among all the Full Node validators and accepting the majority vote as being truthful?

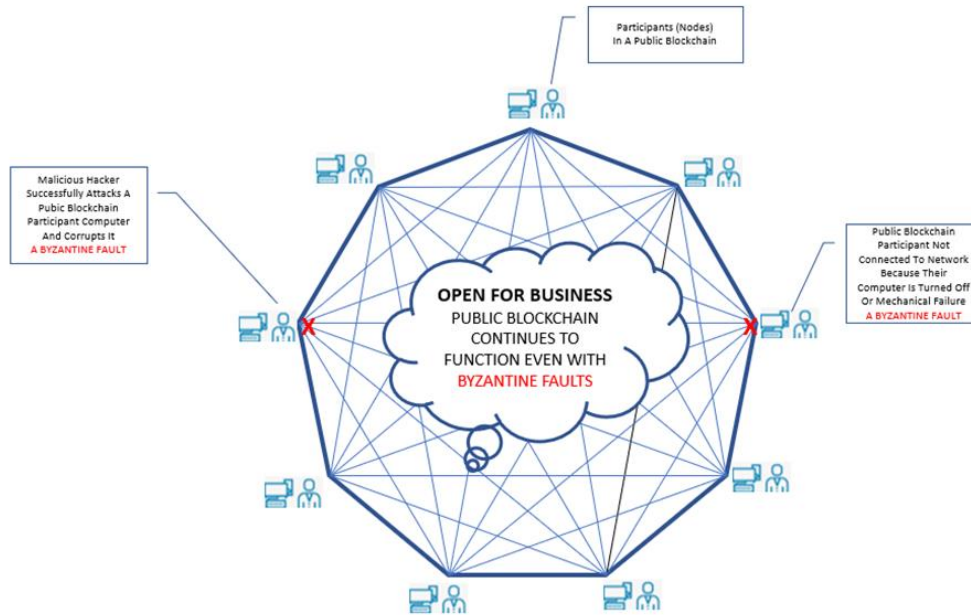
This decision making uncertainty problem of seeking the truth, applies to any trustless decentralized network, whether a group of Generals or participants in a trustless Public Blockchain network.

When studying Blockchain technology, especially ‘trustless’ Public Blockchains, two fundamental building blocks naturally pop up:

1. The mysterious Satoshi Nakamoto (accredited unknown author of Public Blockchain) and
2. Solution to the **Byzantine Generals Problem**.

The bizarre problem with the Byzantine Generals Problem is understanding the problem and its solution in regard to Public Blockchain. Here goes my plain English attempt at understanding the bizarre problem.

A reliable computer system administering a trustless decentralized Public Blockchain must be able to function (see below diagram) even if one or more of its components fails – for whatever reason – (1) mechanical unintentional failure or (2) failure due to intentional malicious activity of a bad actor (an external hacker or malicious attack by an insider).



A failing component in a network, known as a **Byzantine Fault** (whether unintentional or intentional), if not managed, results in the delivering of contradicting data to different sections of the network.

The Byzantine Generals Problem is an abstract expression of the problem of dealing with **Byzantine Faults** associated with computer component failure (whether mechanical failures or malicious failures) and how the system can continue to reliably operate even with these faults.

Understanding Byzantine Fault Tolerance (BFT)

There are several reasons why a Public Blockchain trustless decentralized computer system could fail, or incur a Byzantine Fault. In the military scenario, Byzantine failures are essentially traitors attempting to interrupt communications between loyal Generals.

In contrast, in a Blockchain network scenario, a computer failure could be an unintentional software defect, an unintentional hardware malfunction, or an intentional malicious attack (by an insider or hacker outsider). Byzantine Faults don't always have to be the result of a well-coordinated effort by a bad actor.

There can be difficulties that hinder Full Node validators or miners from reaching consensus (agreement) on decentralized Public Blockchain networks (more on that later in the below Proof-of-Work consensus discussion). Any system failure that exhibits diverse symptoms to different observers is a **Byzantine fault**. It contains no constraints and assumptions about the type of behavior that a node can exhibit (for example, a node can innocently or intentionally generate arbitrary data – a Byzantine Fault - while posing as an honest actor).

In every distributed computer system, Byzantine fault failures are virtually unavoidable. Let's imagine there's a power outage and all of the Public Blockchain nodes go offline simultaneously. Now, the question arises if the network is still operational and capable of sustaining reliable communication? Or does the system as a whole stop working or become open to attacks all of a sudden?

In a reasonably secure decentralized network, anything as minor as a few offline nodes has no discernible effect on the network, the network continues to operate 'Business As Usual'. **Byzantine Fault Tolerance** is the ability of the system to defend against Byzantine fault conditions. Networks that can endure many **Byzantine faults** are said to have **High Byzantine Fault Tolerance**, implying that they are more secure than those that don't.

There are two types of fault tolerance:

- **Crash fault-tolerance (CFT):** non-malicious, benign faults (node off line, etc.)
- **Byzantine fault-tolerance (BFT):** arbitrary or malicious faults.

Byzantine 'flaws', a special type of Byzantine fault, are the most serious and difficult to correct. Extremely high Byzantine fault (flaw) tolerance is required in nuclear power plants, aviation engine systems and pretty much any system whose actions are dependent on the results of a large number of sensors whose risk of failure (faults or flaws) can have dramatic impacts, particularly impact on human life.

Central vs. Decentralized Systems – The Problem

In a centralized online internet connected network managed and controlled by a trusted central authority, such as a bank, that central authority provides the service of coordinating transactions between clients as well as providing the necessary trust and security that a transaction is conducted between authentic parties and the transaction takes place as intended by the parties (that's code that a party's funds to be sent to another, is not accidentally sent to the wrong party, or a malicious party is prevented from interfering with the transaction causing the funds to be routed to the wrong party). Consequently, there is one trusted central authority that makes the necessary transaction decisions for the transaction to take place truthfully and as intended by the client parties. Of course a centralized system is subject to the risks (even though a small probability of happening) of the trusted central authority becoming corrupt or a single point of failure by a successful hacker attack.

Centralized trust systems do not necessarily solve the Byzantine Generals Problem (which Byzantine problem requires that truth of a decision be established without trust). Centralized systems are vulnerable to corruption (at least conceptually) by the central authority, resulting in a breach of trust (comparable to the Byzantine Generals Problem – who and what do you trust?).

In contrast, the central issue (an intended pun) with a trustless decentralized network such as a Public Blockchain, is that there is no reliable central trusted source of information and no way of centrally verifying the information received among members of the Public Blockchain network, especially when the participants in the trustless network are complete strangers – have equal characteristics of trusting and not trusting each other, spread out across the world and use the internet as the only means to communicate. Participants in a trustless decentralized network cannot conveniently all be in a common room at the same time and with a easily verifiable show of hands determine a decision by majority count.

Trustless Public Blockchains and The Byzantine Generals Problem

The problem with gaining consensus or agreement among complete strangers in a decentralized trustless Public Blockchain network (especially gaining consensus – majority approval vote - among participating Full Node validators or miners, that a proposed permanent recording of Blocks of information onto the Public Blockchain network is approved because the information and the sponsor of the proposed recording are authentic), is a **Byzantine Generals Problem**. Can the validators trust the vote of the other validators?

Solving The Byzantine Generals Problem

The theory and proofs behind the Byzantine General’s Problem are nicely presented in two references:

The Byzantine Generals Problem, L Lamport, R Shostak, M Pease, ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382-401 (<https://dl.acm.org/doi/10.1145/357172.357176>); and <https://cointelegraph.com/blockchain-for-beginners/how-does-blockchain-solve-the-byzantine-generals-problem>)

The references discuss that the Byzantine Generals Problem is solvable if and only if more than two-thirds of the Generals are loyal. This means the Generals can assume the truth of the decision to Attack or Retreat, if and only if more than 2/3rds of the messages received are from loyal Generals.

In mathematical equation form regarding **Byzantine fault tolerance** (arbitrary or malicious faults):

- Let **[N]** equal the total number of Generals (both loyal and traitors, and a loyal General is deemed to also be a traitor General if that General’s messenger is bribed by the enemy and the messenger reports a bogus message);
- Let **m** equal the number of traitor Generals (including a loyal General whose messenger is successfully bribed by the enemy);
- Therefore THE GOLDEN FORMULA⁴⁴: **$N \Rightarrow 3m + 1$ (the total number of Generals must be equal to or greater than 3 times the number of traitor Generals plus 1).**
- Or, **$m \leq (N - 1)/3$ (the total number of traitors must be equal to or less than the total number of Generals (loyal + traitor) minus 1 and that result divided by 3).**

The below table illustrates the necessary minimum total number of Generals (both loyal and traitor), assuming an example range of traitors (ranging from 0 to 5, and one extreme case, 3,333), required to solve the Byzantine Generals Problem. (The minimum number of Generals will provide an acceptable **Byzantine Fault Tolerance**).

Number of Traitor Generals, m (Counting a loyal General as a traitor if their messenger is bribed)	Minimum Number Of Generals $N = 3m + 1$ (Loyal + Traitor)
0	1
1	4
2	7
3	10
4	13
5	16
3,333	10,000

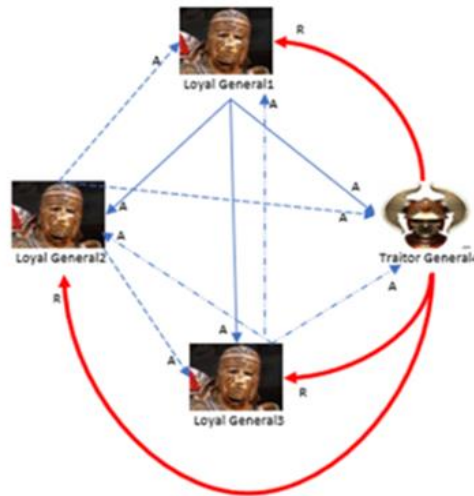
The following observations are made:

- If there are no traitors, a minimum of 1 General is required for a truthful decision (the trivial solution);
- If there is only 1 traitor, a minimum of 4 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat; In other words the Byzantine Generals Problem is not solvable if there is one 1 traitor and less than 4 total Generals;
- If there are 2 traitors, a minimum of 7 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat;

⁴⁴ If only Crash fault tolerance (CFT) were considered, the GOLDEN FORMULA in that case is **$N \Rightarrow 2m + 1$ or $m \leq (N - 1)/2$**

- If there are 3 traitors, a minimum of 10 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat;
- If there are 4 traitors, a minimum of 13 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat;
- If there are 5 traitors, a minimum of 16 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat;
- If there are 3,333 traitors, a minimum of 10,000 Generals are required to result in the Generals taking as the truth receipt of 2/3 consistent messages to Attack or Retreat.

As an example (refer to the below diagram example)



	# Attack Messages	# Retreat Messages	Attack Decision
Loyal General1	3	1	75%
Loyal General2	3	1	75%
Loyal General3	3	1	75%
Traitor General4	3	1	75%

...assume 4 Generals, 1 is an unknown traitor General and 3 loyal, they all agree that they will all collectively honor the decision to Attack or Retreat, based on receiving at least 2/3rds consistent messages, and the loyal Generals all send messages to Attack and the 1 traitor General sends a message to Retreat. As shown in the diagram and table, the overall decision is for all Generals to Attack since they each received 75% Attack messages, which is equal to or greater than the 2/3rds minimum message count.

There is one special calculation shown in the table that illustrates if there are 3,333 traitor Generals, then there needs to be at least 10,000 total Generals. This observation is **very relevant** when applying the Byzantine Generals Problem decision solution in regard to trustless Public Blockchains (open to the public and anyone can participate), a decentralized trustless network in which all Full Nodes (the Generals) can be complete strangers, communicate by internet contact and no way to know which Full Node is loyal or disloyal (a malicious participant). Trustless Public Blockchain's typically will have much more than 10,000 Full Node validators or miners participating in validation services and Consensus Protocol voting, and likely **only a handful** of malicious Full Node validator/miners or 3rd party interfering hackers (certainly much much less than 3,333) attempting to attack or disrupt the trustless Public Blockchain authentication and Consensus Protocol approval voting process (Full Node consensus vote to permanently record Blocks of information in the trustless Public Blockchain network). Consequently, the Full Nodes each receiving a huge Consensus Protocol vote count toward reliance on a common decision that a transaction is truthful, is a **highly reliable** indication that the approval process is extremely truthful and not subject to being hacked or bogus – a solution to the bizarre Byzantine Generals Problem!

The key for a successful Consensus Protocol used in a trustless Public Blockchain network, is to ensure a large number (many many thousands) of Full Node validator/miner participants, which would then require an enormous (extremely highly unlikely) number of malicious participants to successfully attack the Consensus Protocol!

Since Private and Permissioned trusted Blockchains involve their participants already knowing each other and already trust each other, the Byzantine Generals Problem for all practical purposes does not apply, and therefore verification of authenticating Blocks of data to be permanently recorded in the Private/Permissioned Blockchain, is much much less complex, requiring only a few trusted participants to approve Block additions.

Money and the Byzantine Generals Problem

Money is a very real example of the Byzantine Generals Problem. How should a society establish a money that all members of a society can trust and agree upon? For much of history, societies have selected (recognized as having tradeable value) precious metals (gold) or other rare goods, such as sea shells or glass beads, as money. In some ways, gold solved the Byzantine Generals Problem: it was ‘trusted’, won’t rust, and recognized across decentralized systems, such as international trade. However, its weight and purity remained unreliable, and still does to this day. The failure of gold to completely solve the Byzantine Generals Problem (trusting the quality of gold) resulted in trusted central parties, governments, taking over the establishment and issuance of fiat⁴⁵ money. Governments monopolized the minting of money in order to inspire trust in the ‘*weight and purity*’ of the money. Centralized systems did not solve the Byzantine Generals Problem. Governments, the trusted central authorities for money, have regularly manipulated that trust by seizing, debasing, or changing the money. Sometimes such manipulation is for the good and motivated by a desire to help society and seek justice and fairness (such as controlling inflation), and sometimes motivated by selfish governance reasons designed to enrich a selected politically powerful few. Like in all things human’s touch, things can be used, misused or abused – even cryptocurrency is not immune.

⁴⁵ Fiat money is currency exclusively governed, minted and issued by a sovereign government and the management of the money assigned to a central bank, and laws passed to punish counterfeiters. The value of such money is recognized by those using the money based on the users recognizing and accepting the full faith and credit of the issuing sovereign. Western nation fiat money (referred to as hard currency) is generally recognized as having world-wide tradeable value and its conversion/exchange to other currencies readily acceptable, whereas lesser sovereign’s money (referred to as soft currency) and not readily convertible/exchangeable to other currency especially hard currency, generally only has recognized value within the sovereign nations boundaries, and that value recognition often existing only because of local governments mandating that in-country transactions only take place with the local fiat (soft) money. (Folks in soft money countries still covet hard currency, and the reason underground black market money exchange markets develop. A quick test of how valuable and secure soft currency is, is to check out the black market trading exchange rate with hard currency – the higher the black market trading rate compared to local government recognized exchange rate, the more risky the soft currency). Which would you want in your wallet?: U.S. Dollars, British Pounds, Euro, Australian Dollar, Canadian Dollar (as examples of hard tradeable ‘Western’ currency) or North Korean Won (example of a soft non-tradeable currency). An investor investing in a foreign land, desires for any local soft currency earned profit they make in the investment, be easily converted/exchanged for hard currency as well as the right to freely export that hard currency to the investor’s home nation. Conversion of soft currency to hard currency can be a challenge especially if the local central bank has tight exchange regulations or limited hard currency holdings making the conversion/exchange challenging. Typically hard currency conversion/exchange is sourced from the local central bank since rare is an international third party currency trader willing to exchange its hard currency holdings for local soft currency, since the risk of converting such soft currency back to hard currency may be challenging especially since the exchange value has widened, meaning it takes more soft money to trade for a given amount of hard currency (example: trading 1 US dollar today for 900 North Korean Won, is not a good investment if tomorrow it takes 1000 North Korean Won to buy back the same 1 US Dollar. Contradicts the buy low sell high strategy).

Trustless Public Blockchain decentralized cryptocurrency incentives, focuses on the root ‘problem’ with conventional fiat money, is all the trust in the government that's required to make it work – (which isn't necessarily a bad thing). The central bank must be trusted not to debase the fiat money (the ideal circumstance, but the reality is that fiat money is managed and often its value adjustment needed to manage policies that affect society, such as controlling inflation or recession). In contrast, ‘independent’ decentralized cryptocurrency investors value cryptocurrency in centralized controlled ‘stable’ fiat money (the value of 1 Bitcoin in U.S. Dollars) so even the purity of decentralized cryptocurrency is influenced by the manipulations of impure centralized controlled fiat money – which would you want in your wallet?).

Centralized banks are trusted to hold money and transfer it electronically, but they also lend it out in waves of credit bubbles with potentially insufficient *reserve cover* (for each dollar loaned there needs to be a minimum amount of unloaned funds in the vault available to cover any bad loan debts that are not repaid), however, there are restraints on those reserves as established by regulatory authority whose motive is to ensure a certain level of liquidity (easily accessible cash on hand to pay debts) as well as debt coverage. Central authority also involves a degree of trusting them with user's privacy, trust them not to let identity thieves drain accounts. These central management and security services come at a cost since there is an administrative expense to do so. And yes, administrative costs in a centralized system may make microloans and micropayments impractical. Administering a \$10 loan is not cheap for anyone as the same expense to administer a \$10 loan may be the same expense administering a \$100,000 loan.

These cited trust issues have some degree of merit – but caution is advise to assume these issues are persistent material day-to-day occurrences – last time I checked, my bank account balance has been pretty steady freddy and haven't noticed in the newsprint regular occurrences of hacker attacks on bank accounts. Such attacks can exist as a risk in any currency/money management system – inclusive of cryptocurrency.

Risk is: ***the likelihood of an event happening, good or bad, and the magnitude of its impact.*** Examples:

- From a risk perspective, an airplane crash can have the most impactful event ever – loss of ones life, however the likelihood of such an event is extremely low so we all fly in airplanes;
- Being hit with tornadoes or a hurricane can have sever impactful events ranging from extensive property damage to loss of life, however, the likelihood of such an event is low so we continue to build structures out of sticks of wood above ground and supplement the risk with insurance – sometimes insurance claim money helps, at least for property rebuild, not so much for life resurrection;
- Investing in cryptocurrency has its risks – buy low sale high strategy is pretty common even for conventional stocks and bond investments controlled by central authorities, but the swing in value of crypto against fiat money is pretty startling and not for the faint of heart - sort of a tough business model when it comes to microloans, the participants of which have a challenge with even a 10 cent swing in value.

A generation ago, multi-user time-sharing computer systems had a similar problem. Before strong encryption, users had to rely on password protection to secure their files, placing trust in the system administrator to keep their information private. Privacy could always be overridden by the admin based on their judgment call, weighing the principle of privacy against other concerns, or at the behest of superiors. Then strong encryption became available to the masses, and trust was no longer required. Data could be secured in a way that was physically impossible for others to access, no matter for what reason,

no matter how good the excuse, no matter what. Even so these new cryptographic encryption successes, such as RSA and elliptic curve cryptography (discussed in Article 10), are not limited to the Blockchain domain, but can be used in any domain, even centralized networks.

With trustless e-currency (digital cryptocurrency like Bitcoin or Ethereum) based on encryption cryptographic proof, without the need to trust a third party middleman, currency can be secure and transactions effortless. Of course such efficiency must also take into account citizen's contribution to its societal obligations and in particular adhering to relevant regulations such as taxing, anti-money laundering and know your customer anti-terrorist rules. Privacy is ok, but if it is used as a tax, sanction busting or money laundering dodge, regulators will appear. Usually happens when a thing is misused or abused.

In order for a tradeable currency to solve the Byzantine Generals Problem, the ingredients in a successful money trust soup recipe would include it would have to be **verifiable, counterfeit-resistant, and trustless**. It was not until the invention of Bitcoin that the purity of these ingredients were achieved.

Even so, the 'value' of decentralized trustless bitcoin (and other convertible virtual currency such as cryptocurrency) is measured in centralized trusted hard currency fiat money, strange as that may be, given crypto is viewed as an 'independent' non-manipulable trustless decentralized wealth generator yet dependent on manipulable fiat money, for its value...that's a big deal. (Sort of like electric car drivers touting their purity and disruptive commitment to being green because they drive electric cars...yet forget that the electricity they use to charge up their batteries was more than likely generated by a fossil fuel power generator somewhere. Sure alternatives such as solar, wind maybe even nuclear or hydrogen, supplement the power generation pool...but society is still very much reliant on fossil or carbon fuels. Thus endith the lesson on reality and inconvenient truths).

Bitcoin and the Byzantine Generals Problem

Bitcoin was the first cryptocurrency that realized a solution to the Byzantine Generals Problem with respect to money. Many proposals and projects preceding Bitcoin had attempted to create money separate from the government, but all had failed in one way or another. It's all about giving people a way to communicate safely and securely in an unpredictable world. In the actual world, most transactions occur between strangers who do not know or trust one another.

Blockchain Solves the Double Spend Problem

As a tradeable currency system, Bitcoin cryptocurrency needed a way to manage ownership in the currency and prevent the risk of double spends (illegally spending the digital coin twice – or more). To achieve this in a trustless manner, Bitcoin uses a Public Blockchain, (complete trustless strangers can participate), decentralized ledger which permanently stores and records a chronological history of all convertible virtual currency transactions. In the Byzantine Generals Problem analogy, the truth (or decision in an uncertain trustless Consensus Protocol voting environment) that all parties must agree to **is the Blockchain**, and how digital currency transactions are approved as being authenticated and then chronologically recorded in the chain (the Consensus Protocol) and become immutable, can't be changed (auditable) and non-repudiated (once recorded, transacting parties cannot then deny the validity of the transaction).

If all members of the Bitcoin network, called nodes, could objectively agree on which transactions occurred and in what order, they could then verify single ownership of Bitcoin and establish a functioning,

trustless tradeable currency without a centralized authority and avoid double spending by a common owner of the same coin. Blockchain establishes that objective agreement.

What about security?

Since the decentralized trustless Public Blockchain system dealing in cryptocurrency (such as Bitcoin or Ethereum) was designed to be utilized by the general public (and dealings between complete trustless strangers),

- (1) **Byzantine fault-tolerant** mechanisms (a method to achieve consensus on a request to add information to the Blockchain – the Consensus Protocol) and
- (2) **encryption/decryption cryptographic** security techniques

are two essential security and truth ingredients required and used in Public Blockchains.

The unknown and mysterious Satoshi Nakamoto released the first Bitcoin whitepaper in October 2008. Satoshi devised a means to use cryptographic security and public-key encryption to provide a solution to the Byzantine General Problem in a trustless decentralized digital electronic network. To prevent data tampering, cryptographic security uses hashing, a process of encryption. To prevent user identity falsification, the identity and authentication of a network user is verified via public key encryption.

A buy/sale convertible virtual currency, such as cryptocurrency, transaction is secured in a Block of data, comprised of many such transactions, that is connected and linked to other Blocks in the Blockchain by its hash value, a cryptographic secure security method. All individual transaction hashes contained in a single Block, may be tracked down to a root of all hashes in a Block, and all Block hashes chronologically tracked down to an initial or first (Genesis) Block recorded in the Blockchain. This chain and tracking of hashes for transactions and Blocks, and tracing all back to a root/Genesis transaction or Block, is a data management technique known as a Merkle Tree, used to prove authenticity and search for information by using the hashes, that grows from the first or genesis Block to be recorded. A Merkle Tree is an efficient way of combining multiple Block hash identifiers into a group of hashes, whereby many hashed Blocks are ultimately defined by just one master hash. The master hash can be reversed, resulting in disclosing ultimately individual hashed Blocks as well as individual transactions (an auditable structure).

In the Bitcoin Blockchain network, every Block in the Blockchain network that comes from the first Block, the genesis Block, are reliably valid. Full Node validators or miners that verify the authenticity of Block participants and Block content, validate Blocks, compete with other to solve complex mathematical cryptographic puzzles to win the right to produce and record Blocks to the Blockchain and be paid a cryptocurrency Block validation service fee, as part of the **Proof-of-Work** consensus method.

The Bitcoin network uses asymmetric cryptography private and public cryptographic key pairs, for digitally signing a communication to verify identity of participants in a transaction as proof and that it comes from authentic persons. Once message content, in addition to participants identity, have been authenticated, the transaction information is chronologically and immutably (can't be changed) and non-repudiable (once recorded, transaction parties cannot deny the truthfulness of a transaction) recorded in the Blockchain for transparency and historical auditable proof of accountability.

Hackers (such as man-in-the-middle) cannot readily attack the Blockchain because data and information in the Blockchain use cryptographic security. To prevent manipulation, the data or transactions are bundled into Blocks and hashed for added protection. Satoshi makes things more probabilistic by putting

Full Node validator miners in a competition to validate the Blocks. This makes it more decentralized because no single Full Node validator miner can receive all of the rewards by monopolizing the validation process.

Proof-of-Work Solves the Byzantine Generals Problem

The Byzantine Generals Problem can be solved by implementing a Blockchain protocol (set of procedures) that employs Byzantine fault tolerant mechanisms. When faced with uncertainty, adopting a procedure (Consensus Protocol) among the Full Node validators or miners is the best method to make reliable truthful decisions – game theory thinking.

As a result, the decentralized trustless Public Blockchain Consensus Protocol decision making process becomes a probabilistic game (Full Node validators or miners accepting, for example, 51% likelihood yes vote of the truthfulness of their cast votes) rather than a fixed deterministic vote (an event either happens or it does not, same input results in same output every time) because nothing can be guaranteed (well maybe death and paying taxes are exceptions). That is the case when there is trustless decentralized peer-to-peer network and less (none) direct communication among peers, and each is independent and self-contained and trustless.

Bitcoin Blockchain manages to solve the Byzantine Generals Problem by using a **Proof-of-Work** mechanism in order to establish a clear, objective, reliable, consensus ruleset for the Blockchain. Here's how that works...

Useful references:

A Decomposition Of The Bitcoin Block Header <https://www.datadriveninvestor.com/2019/11/21/a-decomposition-of-the-bitcoin-block-header/>;

Bitcoin Proof of Work, YouTube, by Ioni Appelberg, Sep 9, 2020;

What is Proof of Work (PoW) Explained for Beginners, Bianca Academy,

<https://www.youtube.com/watch?v=3EUAcxhuoU4&list=RDLVXLcWy1uV8YM&index=8>

What Is Proof of Work (PoW)?

In a decentralized trustless Public Blockchain digital currency – cryptocurrency – network (such as Bitcoin), there is no central single authority to securely confirm the authenticity of transacting parties or their transaction content.

Proof-of-Work (PoW) describes a substitute secure transaction validation system that works in a decentralized trustless Public Blockchain environment in the absence of a centralized authority.

The Proof-of-Work system requires a not-insignificant but feasible amount of effort (or **work**, in the nature of costly computer capacity and upgrade investments and computer computation time and associated electricity operating costs) as a mechanism for competing third parties (Full Node validators also known as miners) interested in securely validating Public Blockchain transactions (confirming the truthfulness of transacting parties and their digital currency - cryptocurrency - transaction content) in order to efficiently discourage frivolous or malicious uses of computing power that must be expended in the Bitcoin network for sending spam emails or launching hacker denial of service attacks. Such spammers or attackers will not invest the time and electricity cost when their spammed email or hacker attack are after such expenditure vetoed from being allowed on the network. A waste of money incentive.

The Proof-of-Work concept was adapted to securing cryptocurrency (convertible virtual currency or digital currency) through the idea of "reusable proof-of-work" using the **Secure Hash Algorithm 256 (SHA-256)** hashing one-way encryption algorithm. Hashing is an encryption technique that converts any size plaintext data set into a fixed size (64 bits) unintelligible alpha-numeric number (a cypher) called a 'hash'. The same hash will be generated every time if the exact same data is processed in the hash algorithm. If the data is changed in the slightest, such as a small letter changed to a capital, or a space added in the data, then a completely different hash number is generated. A hash may be used to confirm that original data has not been altered or tampered with by comparing the original hash with later hashes.

Proof-Of-Work work forms the basis of many Public Blockchain cryptocurrencies, allowing for secure, decentralized consensus.

When cryptocurrency, such as Bitcoin, transactions occur, such transaction information is first temporarily stored in a 'Mempool' (also referred to as the transaction pool). Groups of transactions retrieved from the Mempool, go through a security verification check (authentication process of transacting parties identity and transaction content – truthful buying and selling of cryptocurrency) and then such transactions are grouped into a fixed size (example 1MB, or 1 million bytes, about 2,000 transactions, with 8 bits to a bit, binary code bits, the language of computers) Block of transactions. The transaction authentication service is provided by certain members of the Public Blockchain network referred to as Full Node validators or 'miners'. In order to incentivize third party Full Node validators or miners to validate transactions, they are rewarded with a cryptocurrency fee if they are the first miner to expend computer work to validate Block transactions and solving a time consuming mathematical puzzle, after which the first place winner puzzle solver is confirmed by a majority vote of the other miner's, and upon that approving vote, the winning validator then permitted to permanently record the Block of data into the Public Blockchain (and receive a cryptocurrency fee). The probability of double spending the same cryptocurrency is virtually eliminated after the Validator receives at least six confirmations of the Block add request.

Understanding Proof of Work

This explanation will focus on Proof-of-Work as it functions in the Bitcoin decentralized trustless Public Blockchain network. Bitcoin is a digital currency (cryptocurrency) that is underpinned by an electronic decentralized trustless ledger known as a "Public Blockchain." This ledger contains a permanent immutable (can't change it), non-repudiable (parties can't deny the truthfulness of information once recorded) record of all Bitcoin transactions (buy and sale Bitcoin activity), organized chronologically in sequential "Blocks" of transaction information – (size of each Block determined by a maximum binary code length, so many bytes of data, the 0s and 1s binary code language of computers). No user can spend any of their Bitcoin holdings twice (the double spend problem) since any attempt will flag the system that a previous conflicted record of the coin exists and the system denying the double spend conflicted transaction. In order to prevent tampering, the ledger is public, or "decentralized and distributed"; an attempt at altering a recorded version would quickly be flagged and rejected by other users, plus to change any data in a recorded Block, would require that the entire Blockchain of data be remined, which would entail an astronomical and prohibitively expensive amount of work and cost (especially electricity).

The way that users detect tampering of recorded transactions in practice is through the comparison of one-way cryptographic encryption *hashes*, long strings of fingerprint like identification alpha-numeric numbers (whose characters are numbers and letters and some cases symbols, such as "0000Abc34&%nD...") that serve as Proof-of-Work.

Put any size set of information or data through a computer program hash algorithm function (Bitcoin uses SHA-256 algorithm), and it will only ever generate one unique fingerprint like alpha-numeric *hash* having 64 bits of characters, and repeat that same result no matter how many times the data is hashed. Due to the so called "avalanche effect," however, even a tiny change to any portion of the original data (such as changing a lower case letter to a capital, or adding an extra space) will result in a new unrecognizable hash totally different and readily recognizable from the original hash. Whatever the size of the original data set, the hash generated by a given function will always have the same length (64 bits if the SHA-256 hash algorithm is used). The hash is a one-way encryption function (change plaintext to an unrecognizable alpha-numeric cyphertext): it cannot be used to obtain (decrypt) the original data (convert cyphertext back to plaintext), only to check the authenticity of data that the hash of the original data exactly matches the hash of the data received – if the same, then the data is exactly the same as the original, if different, a red flag that the data has been altered or tampered with.

How Does Proof-of-Work Work?

Two purposes fulfilled by Public Blockchain Full Node validators or ‘miners’ include:

1. **Purpose 1: To secure the Bitcoin Public Blockchain network** – how?
 - a. By validating transactions following certain rules (set up in the Consensus Protocol) – in practice the Blockchain mathematical rules ***automatically (no human being is involved)*** provide the validation service, and the only thing the miner needs to do is make sure the network computers are up and running...here’s what happens automatically...
 - i. When a Blockchain transaction (buy or sale Bitcoin between two parties) first takes place on the Blockchain, it is first stored in a temporary storage area referred to as the ‘mempool’;
 - ii. Miners retrieve temporary unconfirmed stored transactions from the mempool, authenticates or validates the transaction (confirms the truthfulness of the identity of transacting parties and transaction content);
 1. Confirms a transacting party is not trying to spend the same Bitcoin twice (the double spend problem),
 2. Confirms the alleged owner of Bitcoin has a valid digital signature that they truthfully are who they say they are. (Digital signatures are special electronic encryption keys, alpha-numeric coded numbers, that represent the identity card or fingerprint – sort of like a passport - of a Bitcoin owner)
 - iii. Then a certain number of validated transactions (~2,000) are grouped together into a fixed size (~1MB) Blocks of data, and that Block awaits being recorded into the Blockchain.
2. **Purpose 2: Create new Bitcoin** (by way of being paid in newly ‘minted’ (digitally created) cryptocurrency – convertible virtual currency, (the validation service fee) by competing miners expending Proof-of-Work effort.
 - a. Miners have to provide proof that they have invested computational ‘work’ or power (electricity and computer computational time) in order to permanently record validated Blocks of data onto the Blockchain. And the first miner who provides that proof, can then permanently record Blocks of data to the Blockchain AND receive the cryptocurrency validation incentive fee (freshly minted Bitcoin) for providing the service.
 - b. The invested computational work is in the nature of the miners having spent funds to secure special costly computer equipment AND spend a large amount of computer time and electricity costs to operate the special computer equipment to solve, by ‘brute force’ (a trial and error pseudo-random number guessing procedure – somewhat like a lottery

game) and be the first miner to determine the solution to a mathematical cryptographic (encrypted hash) puzzle. Solving the puzzle will inherently require large amount of expended work, thus by showing the solution to the puzzle has been achieved, is proof that work has been expended – ergo: Proof-of-Work.

- c. Each Block of data is organized with certain metadata in the Block Header (sort of like a table of contents in a book, example metadata could include author, date created, date modified and file size). Listed in the Block Header is a special number called a ‘nonce’ (number used once, whose characters are a 32 bit number of 1s and 0s computer speak computer binary code, the origin of the nonce is from the Blockchain administration and operating computer program protocol and is unique for each Block. It is the seed value or initial nonce used by a Full Node Validator or miner in Proof-of-Work Consensus Protocol puzzle analysis. The nonce is a fundamental component in the Proof-of-Work process.
 - i. The security *captchas* we often find on websites requiring human intervention to manually type in certain letters and numbers into and website, is designed to deter internet bots from entering the site, are nonces (albeit with letters included) as they are a string of characters used just once.
 - ii. "Nonce" is a portmanteau [a word blending the sounds and combining the meanings of two others, for example *motel* (from ‘motor’ and ‘hotel’) or *brunch* (from ‘breakfast’ and ‘lunch’)] of *nonce*, "**n**umber used only **o**nce." It is a four-byte pseudo-random generated number added to a hashed—or encrypted—Block Header, that, when that combination is hashed again, returns a new hash number that is checked as being a solution to being equal to or less than a target hash number (set by the Blockchain administration protocol). The nonce is the mathematical puzzle number or ‘password’ that Blockchain miners are solving for. The nonce in the Blockchain header is the seed value or initial nonce used by a Full Node Validator or miner in Proof-of-Work Consensus Protocol puzzle analysis. This all sounds confusing, but an example below of the flow of the solution...
 - iii. The Block Header is composed of an 80-byte long alpha-numeric string of characters, unique and cryptographically secured, which is what gives it the property of immutability (can’t be changed).
 - iv. The Block Header (examples below) is comprised of 6 components (version, prior Block hash, Merkle root, timestamp, difficulty target or bits (or target hash) and nonce:

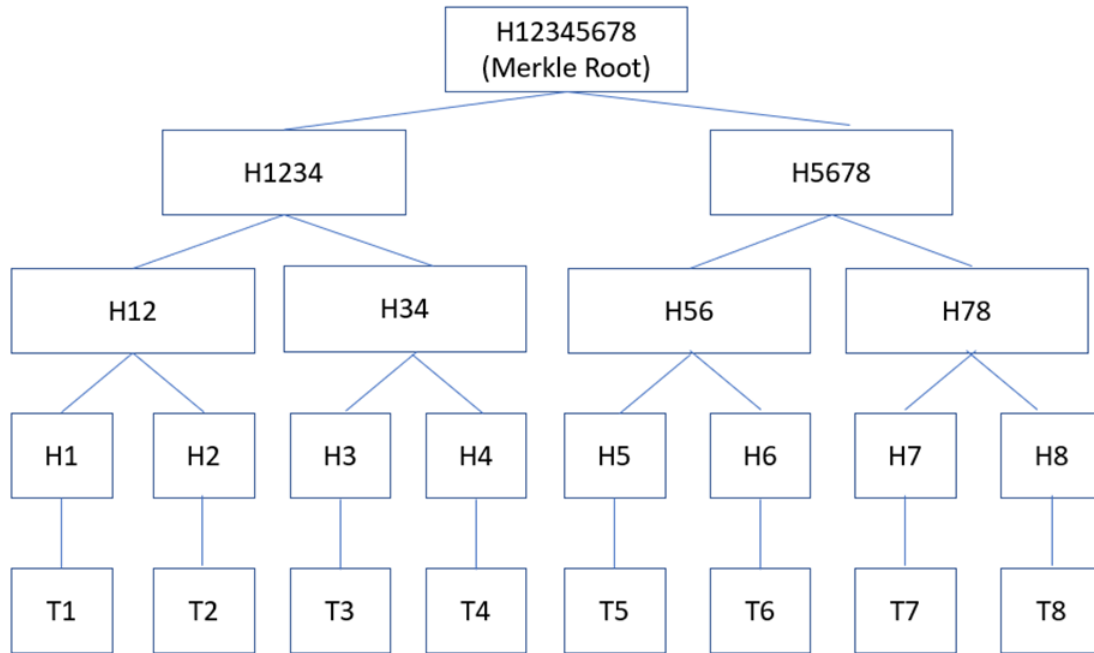
version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c817010000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833

Block hash

0000000000000000
e067a478024addfe
cdc93628978aa52d
91fabd4292982a50

```
{
  "bits": 419486617,
  "mrkl_root": "359d624d37aee1efa5662b7f5dbc390e996d561afc8148e8d716cf6ad765a952",
  "nonce": 1225187768,
  "prev_block": "000000000000000ced0958bd27720b71d32c5847e40660aaca39f33c298abb0",
  "time": "datetime.datetime(2014, 4, 5, 7, 49, 18, 0, tzinfo=tzutc())",
  "ver": 2
}
```

1. the 4-byte long Bitcoin **version** number,
 - a. The **version** (currently 1 thru 4) refers to which Block validation rules (Consensus Protocol) the Block follows. If the Block version number is different from other Blocks in the Blockchain, then they cannot be a part of the same Blockchain. In that case, the different version Block would be running on a different Blockchain and could result in a **hard fork (breaking a Blockchain into two Blockchains)**.
2. 32-byte **previous Block Header hash**,
 - a. This hash is what links a Block to the rest of the Blockchain. The first Block, a genesis Block in a Blockchain, is the only Block that will not have a previous Block Header hash.
 - i. If you had 200,000 Blocks and you attempt to change transaction data in a block which would require a change in the hash value of the previous Block, the **avalanche effect** will occur which means it requires changing all the other previous hash values. This ensures that no previous Block can be changed without also changing the Block's Header. This feature provides a tamper resistance measure (the immutability of Blocks – can't be changed) because of the huge amount of cost (in time, electricity, computer computational power) to change a Block.
3. 32-byte long **Merkle root** (hash of the underlying Block data),
 - a. The Merkle Root is derived from the hashes of all transactions (~2,000) included in a Block. The Secure Hash Algorithm 256, SHA256, is used for determining the hash of transaction data included in the Block. Hashing makes sure that that none of those transactions can be modified without modifying the entire Block Header. This is a tamper resistance measure that provides security for the Block, an important feature of a public, decentralized and trustless Public Blockchain system. Example of a Merkle tree and root structure:



Merkle Tree example. The root of the tree is H(12345678) which is the hash of all transactions from T1 to T8.
 'T' = a transaction comprised of unhashed data; H is the hash of the data or of other hashes
 'H' = the hash result.
 The hash (H#) of the root is a concatenation – linked together in a chain - of all the transactions in the Blockchain.
 A simple time efficient use of the root Merkle Tree hash can quickly confirm all transaction data below the root is authentic.

4. 4-byte long **timestamp** of the Block,
 - a. This is the number of seconds that have elapsed since January 1970. The timestamp begins when the miner started hashing the Block Header of a new Block. The timestamp must be strictly greater than the median time of the previous 11 Blocks. Full Node validators or miners will not approve Blocks with Block Headers that are more than 2 hours in the future according to their clock.
5. 4-byte long **difficulty target** for the Block,
 - a. Generating any hash for a set (Block) of Bitcoin transactions would be trivial (easy) for modern computers (converting the plaintext Block Header data to a fixed length 64 bit alpha-numeric fingerprint like hash value), so in order to turn the hashing process into strenuous "work," and useful in Proof-of-Work Consensus Protocol (transaction authentication process) applications, the Bitcoin network administration computer software, establishes a certain level of hash "difficulty."
 - b. The difficulty **target** hash is a 4-byte file also referred to as the **Bits**. The encoded version of the target (set and fixed automatically by the Blockchain administration protocol) is a threshold that the rehash of the Block's Header hash with a guessed nonce value, is less than or equal to the target hash. The more leading zeros in the target hash, the more difficult it is to

find a nonce value. The difficulty target is adjusted every 2,016 Blocks on the Bitcoin network.

- c. The difficulty target is coded as a Bitcoin Blockchain protocol. This determines how difficult the target hash value should be based on the Blockchain network's total hash rate. When there are too many miners, or more powerful hashing computers are added to the network or the target hash difficulty is too low, the Block addition time could fall below a targeted 10 minutes on average (i.e. Blocks are added too fast), and when this happens, the hash target difficulty increases in order to slow up the rate Blocks are added and control (reduce) the supply of Bitcoin issued to miners. When the Block addition time is more than 10 minutes, the difficulty is too high, so the protocol code ensures that the difficulty must be decreased, and more Blocks allowed to be added in a shorter period of time.
- d. The lower the value of the difficulty target (which means the hash target has more leading zeros in its alpha-numeric structure), the harder it is to generate a Block. The difficulty value has a hexadecimal form (example 0000000000419486617).
- e. The target is used by Blockchain administration protocol to determine how quickly (easily) or slowly (more difficult) miners can produce new Blocks which directly impacts the rate of minting new Bitcoin used as payment to validators. The target hash is automatically adjusted every two weeks to change the difficulty of mining. The Blockchain administration protocol has set a target of adding new Blocks every 10 minutes, and this time limit used to adjust the target to achieve the 10 minute objective. The target is the pacemaker that keeps the Blockchain Block addition timing focused on 10 minutes. Don't want that pulse too high or too low or don't want to pay out newly minted cryptocurrency too fast or too slow, but just right. For all practical purposes what is important about the target hash is the number of leading zeros in the hash..."000000...0000.ab6&vg*Uy4.... Since the number of leading zeros is the puzzle the miner is trying to match with the below nonce Proof-of-Work effort.
- f. The difficulty level for mining in March 2022 was 27.55 trillion. That is, the chances of a computer producing a hash below the target is 1 in 27.55 trillion. To put that in perspective, you are about 91,655 times more likely to win the Powerball jackpot with a single lottery ticket than you are to pick the correct hash on a single try.
- g. Mining difficulty changes every 2,016 Blocks or approximately every two weeks. The succeeding difficulty level depends on how efficient miners were in the preceding cycle. It is also affected by the number of new miners that have joined Bitcoin's network because it increases the hash rate or the amount of computing

power deployed to mine the cryptocurrency. In 2013 and 2014, as the price of bitcoin rose, more miners joined its network, and the average time to discover a Block of transactions fell to nine minutes from 10 minutes.

6. 4-byte long **nonce** used by miners.
 - a. The nonce is a 4-byte field that is an arbitrary pseudo-random number the initial or seed value of which is automatically encoded in the Block Header by the Blockchain administration protocol. Miners arbitrarily change the nonce, add it to the Block Header hash, then rehash that combination in order to find a nonce value (can be more than one) that produces the combined rehash hash value less than or equal to the target hash threshold (**only matching the same number of leading zeros, not the entire hash**). The nonce is like the “magic number” because whoever discovers it first and produces a rehash hash that matches the number of leading zeroes in the target hash, will become the winning Full Node Block validator and right to permanently record the Block into the Blockchain and earn a cryptocurrency fee incentive for being the first to solve the puzzle.
 - b. Discovering the nonce is the main activity miners or Full Node validators engage in during the consensus mechanism of validating Blocks. The miners are competing with one another by trying to solve the cryptographic puzzle that must be equal to or below the difficulty target.
 - c. Miners pool together (form joint ventures and expand their computer power capacity of the joint participants by sharing each others systems) to increase their chances of being the first to successfully solving the hash puzzle and mining Blocks, which generates the shared earning of validation service fees and, for a limited time, the fee reward paid in newly-created (minted) Bitcoins. [It has been debated that mining pools tend to detract from the decentralized aspiration of Blockchains].
 - d. It is the seed value or initial nonce found in the Blockchain header used by a Full Node Validator or miner in Proof-of-Work Consensus Protocol puzzle analysis.
- d. The Proof-of-Work mining process is as follows:
 - i. The Full Node validator or miner accesses the Header Block HSA256 hash value (which includes the initial or seed nonce in that hash determination).
 - ii. The miner then randomly guesses (brute force) 32 bit nonce values, combines that with the Block Header hash and rehashes that combination, (in effect a hash of the hashed Block Header) and compares that result (comparing the number of leading zeros in the hash, 64-digit hexadecimal hash number, not solving for the entire hash number) with the target hash number of leading zeros. If the new hash number is less than or equal to the target hash number – has the same number of leading zeros, the miner has successfully found the magic number nonce (or ‘password’ – a ticket into the winners circle) used in the rehash determination and if the first to do so, earns the right to earn a cryptocurrency

fee and permanently record the Block into the Blockchain, and then that Block of data is then considered authentic, part of the Blockchain and that Blocks transactions no longer temporarily stored in the mempool. There can be more than one nonce that achieves a hash value less than the target. But the first miner to find a less than or equal solution, is deemed to be the winning miner. In reality it takes trillions of nonce guesses per second to find a solution to the puzzle...hence the demand on electricity computer computational power. An example rehash solution and target hash...

```
00002a014abd0b5c
5ddb3b1f78bd0c77
206692734db5ebe1
e9625935089d8721
```

- iii. A miner's systems use considerable brute force computing power in the form of multiple processing units stacked together and spit out hashes at different rates—megahashes per second (MH/s), gigahashes per second (GH/s), or terahashes per second (TH/s)—depending on the unit, guessing all possible 64-digit nonce combinations until they arrive at a solution.
- iv. The Proof-of-Work pseudo-random number guessing game is in a logical sense somewhat useless and wasteful...but is an effective way, by using the cost of electricity, as a mechanism to create a competitive environment and secure the Blockchain.
- v. Here's a simple illustrative example to explain the process to expend work and determine a target hash and be the first to do so and earn a cryptocurrency fee. Say you ask friends to guess a number between 1 and 1000 that you have thought of and written down in secret on a piece of paper. Your friends don't have to guess the exact number; they just have to be the first person to guess a number less than or equal to your number.
- vi. If you are thinking of the number 200 and a friend comes up with 210, they lose because 210 is greater than 200. But if someone guesses 160 and another friend guesses 180, then the latter (180) wins because 180 is closer to but less than 200 than 160. The Bitcoin mining math puzzle is a similar example situation described except with 64-digit hexadecimal numbers and thousands of computing systems competing to be the first to calculate a nonce hash number equal to or the closest lesser rehash number (of the combined Blockchain hash and nonce hash) to the target hash.
- vii. Proof-of-Work requires a computer to randomly engage in brute force computing hashing function calculations (by arbitrarily adding characters to the input nonce data which results in a new rehash calculation for the altered Blockchain/nonce hash data), until it arrives at an output with the correct minimum amount of leading zeroes in the solution hash – the calculated hash being equal to or less than the target hash (which also means all the leading 0s in the rehash (using the guessed nonce hash) must match exactly the target hash). For example, the target hash for Block #660000, mined on Dec. 4, 2020 is:

00000000000000000000000008eddcdf078f12c69a439dde30dbb5aac3d9d94e9c18f6.
(19 leading zeros).

The Block reward for that successful hash was 6.25 BTC, at the time worth about \$18,400 per BTC or about \$115,000 fee, thus not a trivial pay check.

That Block #660000 will always contain 745 transactions involving just over 1,666 traded Bitcoins, as well as the header of the previous Block. If somebody tried to change any transaction amount by even 0.000001 Bitcoin, the resultant hash would be unrecognizable (not matching the original data hash), and the network would reject the fraud attempt.

- e. A key reason why Proof-of-Work is so expensive to operate is to make it nonsensical to try and cheat the Blockchain system. Why spend a lot of effort and money to try and cheat the system (try to save false information or double spend cryptocurrency), when at the end of that effort, the system will determine you have not engaged in useful work (hash values won't match) and be denied payment of cryptocurrency fee and denied adding Block data to the Blockchain?! Miners must provide valid Blocks, valid transactions and valid Proof-of-Work. Blockchain rules are automatically enforced by mathematical rules that cannot be fooled. Bitcoin thus has rules without rulers. If it didn't cost a lot of money to mine, there would be no incentive for miners to be quite so honest.
- f. Bitcoin operations while not subject to formal financial regulations, are in fact governed by very strict, automatically enforced mathematical rules. The concerns with Bitcoin Public Blockchain and convertible virtual currency or cryptocurrency transactions are not because of the Blockchain structure and functioning network...but peripheral issues involving how users use, misuse or abuse cryptocurrency activities. For instance:
 - i. Have user's properly paid taxes associated with taxable incomes generated by cryptocurrency transactions? And how can such taxable events be conveniently audited other than self-reporting especially if privacy is protected? The honor system...
 - ii. How are anti-money laundering, anti-terrorist (know your customer), or suspicious activity reporting compliance concerns under the Banking Secrecy Act, implemented?
 - iii. Have user's manipulated the risks associated with the valuation of cryptocurrency such that it preys on the ignorance of investors? Has there been appropriate full disclosure of the risks an investor should be aware of?
 - iv. Are crowd funding projects (such as raising funds by soliciting many participants each investing small amounts of cash to set up a Public Blockchain cryptocurrency operation) actually dealing in a security, necessitating regulatory disclosure oversight and penalties if fraudulent activities, just like raising capital for initial public offering corporate stock?
 - v. Are smart contracts legally enforceable and who are the accountable parties? since smart contracts are merely electronically automatically performed contracts disconnected from human contact performance.
 - vi. Since Blockchain networks do not have a reasonably recognized accountable party if something goes wrong, should Public Blockchain operations be required to maintain a reserve fund (a liability insurance policy?, self-funded) used to compensate victims that have been damaged by failures in Blockchain protocols and provide some form of protection and restitution for innocent users victim of misuse or abuse?
 - vii. If a cryptocurrency owner is determined to be liable to another for damages in a cause of action unrelated to the Public Blockchain, that liable party's assets, if

not otherwise protected at law (such as one's homestead) that a successful plaintiff in a lawsuit is entitled to, will the cryptocurrency account be misused as a mechanism to conceal, hide or store such asset in a cryptocurrency account that cannot be frozen and attached by a damaged plaintiff?

Proof-of-Stake Solves the Byzantine Generals Problem

Proof-of-Stake is another decentralized trustless Public Blockchain Consensus Protocol mechanism that seeks to address the Byzantine Generals Problem. Proof-of-Stake based networks, unlike Proof-of-Work based networks, are not reliant on proving the amount of work expended, or reliant on using large electricity power and incentive to be paid in cryptocurrency for doing so, cryptocurrency mining. Instead, a technique called **staking** is performed.

Full Node verification 'validators' put up as a 'cash' (actually staking digital currency) deposit or stake their digital wallet that shows the amount of cryptocurrency owned,— sort of like putting up a bet in a game of roulette. Validators who put up as a bet or digital currency 'cash' deposit, and the greater value of staked cryptocurrency coinage, are preferentially entitled to validate Blocks and earn validation service fee rewards paid in cryptocurrency – similar to the cryptocurrency fee paid in Proof-of-Work systems. Users that attempt to validate incorrect transactions risk losing their staked cryptocurrency deposit. But those that successfully validate correct transactions, and approved (using the Consensus Protocol ruleset) by other Full Nodes validators for having done so, receive their stake back and paid a fee. Sort of like winning in roulette, if you win you get your bet back then some. Because it takes computer time to validate Blocks, no one Full Node validator, no matter how much cryptocurrency owned, will be able to validate all Blocks because they cannot timely provide the validation service, thus in a practical sense, more than one Full Node validator will always be participating in Block validation services. Can't use the same stake more than once while an outstanding Block validation process is not completed.

Users can stake coins using normal home computers instead of needing specialized costly computers (since they don't have to solve complex mathematical problems), otherwise required if mining in a Proof-of-Work based network. Decentralized trustless Public Blockchain Proof-of-Stake based networks also prevent double-spending attacks and other potential security vulnerabilities caused by Byzantine fault failures. For example, Ethereum 2.0 (Serenity) Public Blockchain will use the **Casper Proof of Stake algorithm**, which requires a two-thirds majority vote of Full Node validators to approve a Full Node validator request to permanently record an authenticated Block in the Blockchain. [A recording Full Node validator successfully put up the highest cryptocurrency stake to preferentially validate the next set of Blocks, and when validated, two thirds of the other competing Full Node validators, vote to approve the authentication of the identity and transaction content are truthful, then permit the Block to be permanently recorded in the Blockchain network and the successful validator being credited back with its stake and paid a cryptocurrency fee for the approved validation service]. The 2/3 majority vote Consensus Protocol hurdle is consistent with the solution to the Byzantine Generals Problem solution ($N = 3m + 1$). In September 2022 the number of Ethereum 2.0 Full Node validators exceeded 426,000, which means it would take about (a highly unlikely) 142,000 malicious hacker attackers to mount a successful attack on the Consensus Protocol algorithm.

The more cryptocurrency tokens held in an electronic wallet, the more mining power is effectively granted to a Proof-of-Stake Full Node validator. While Proof-of-Stake is far less resource-intensive (much less electricity cost), it has several other concerns including (1) a greater chance of a 51% attack in smaller altcoins (other newbie cryptocurrency networks) because of validator control by a few users who have material altcoin balances in their electronic wallets – such 51% control could allow mischievous validators

to change Block data and spend the same cryptocurrency twice or more times, and (2) incentives to hoard cryptocurrency tokens (to keep electronic wallet crypto balances high and better chance of being selected a Full Node validator, and not use them.

Delegated Proof-of-Stake Solves the Byzantine Generals Problem

Delegated proof-of-stake is another decentralized trustless Public Blockchain Consensus Protocol technique that operates similarly to Proof-of-Stake and was first developed in 2014. Both require user validators to put cryptocurrency on the line as a stake or cash deposit and the highest stake wins the preferential right to validate Blocks and earn a cryptocurrency fee for doing so. Only a few user validators (known as *delegates*) can validate transactions and generate Blocks in Delegate-Proof-of-Stake based networks.

In general, any user can contribute any quantity of a decentralized trustless Public Blockchain cryptocurrency coin, to cast a vote in support of a delegate user candidate to be designated as a Delegated Proof-of-State user validator, then authorized to validate Blocks and earn a fee. Any earned cryptocurrency fee of a Delegated Proof-of-Stake user validator, having successfully recorded a Block of information in the Blockchain, is usually shared and distributed to the users who voted for the delegate, in proportion to the amount of cryptocurrency staked or contributed by the voting user and the Delegate user validator. Consequently, this process is a pooling type arrangement where a group of user nodes, contributes cryptocurrency stake funds to a pool and the proposed delegate user validator node using its and the pooled stake, to compete to be the highest stake amount on offer and win to be a preferential delegated user validator to validate a Block and when successful, the delegate user validator and pool members receive back their stake and any earned cryptocurrency service fee for having successfully validated a Block (and the 2/3rds of the other delegate user nodes approving such validation) is shared among the delegate user validator and the pooling participants.

Delegate user nodes can reach consensus (both (1) selection of a delegate user validator and (2) approval by other delegates of a successful delegate user validators efforts) considerably faster using Delegated Proof-of-Stake than they can with Proof-of-Work or Proof-of-Stake Consensus Protocol techniques. At scale, this means Blockchain transactions can be handled significantly quicker. Maintaining a high level of Byzantine fault tolerance with Delegated Proof-of-Stake may become problematic in some cases due to the tradeoff that there are less participants in the Consensus Protocol approval process (because of users participation in a pool and less independent users in the system...theoretically meaning because there are less number of total 'Generals' in the network, it takes a fewer number of 'traitors' or 'malicious hacker attackers' to successfully attack a network). Thus reverting, because of the pool structure, somewhat to a more centralized looking structure and the higher risk of a malicious attack being possible since there are less validating participants in the system because of the pooling arrangement.

Because fewer nodes are accountable for keeping the network safe, it is potentially easier for nodes to conspire against the majority's best interests. Delegated-Proof-of-State based networks, try to avoid this conspiracy scenario by holding delegate elections regularly to ensure that delegates are held accountable for their decisions, and not lock user's into a long term pool.

Practical Byzantine Fault Tolerance (PBFT) Solves The Byzantine Generals Problem

Reference: <https://www.naukri.com/learning/articles/byzantine-fault-tolerance-in-blockchain/>

Practical Byzantine Fault Tolerance (PBFT) is a Consensus Protocol that solves the Byzantine Generals Problem in decentralized Blockchains (including Public, Permissioned and Private Blockchains) that may or may not use cryptocurrency as an incentive payment to Block validators. It is a process that seeks consensus of Block validation decisions by ensuring objective and truthful results in the PBFT approval voting process. The process involves sequential voting procedure of participating voters that do not overlap with each other, but the collective vote affecting the voting outcome (in this case approving or not approving of the recording of Block data to the Block chain). This **sequential voting process** affecting a common system is known as **asynchronous approach**. The PBFT Consensus Protocol entails:

- All **nodes** are assembled in a **sequence**.
- One network node serves as a **leader node**, and the rest of them are **backup nodes**.
- The primary or leader node serves users transaction request. It works as a moderator between user and backup nodes.
- All backup and leader nodes are capable of communicating with each other to check the honest nodes.
- Honest nodes should be able to reach a consensus for the next change (Block addition) in the Blockchain network based on **majority rule**.
- Each node identifies the source of the message to make sure it's sent by the correct sender.
- Ensures the message has not been modified or corrupted in between.

In practice:

- A transaction request is sent by a user or client to the leader node.
- Then the leader node floods the request to all backup nodes.
- All the nodes work on the request and send a reply to the client (the transaction and transaction parties are authentic, approve or disapprove addition of the transaction to the Blockchain).
- The requesting client waits for $(m + 1)$ replies from all the nodes with the same result. Here, m = number of possible (Byzantine) faulty nodes. [m is an estimated number of potential Byzantine faults and will vary from transaction to transaction].

PBFT entails three phases:

The PBFT mechanism consists of 3 phases:

1. Pre-prepare phase: The leader node sends out a pre-prepared message to each backup node.
2. Prepare: After receiving the pre-prepared message from the leader, the backup nodes send the prepared message as a reply to all other nodes including the leader. A node is considered prepared only if it has received pre-prepared by the leader and seen $(2m + 1)$ number of prepared messages from other nodes.
3. Commit: If the nodes are prepared, they send a commit message. If a node receives $(m + 1)$ commit messages, they carry out the client's request.

The Pros and Cons of PBFT:

Pros:

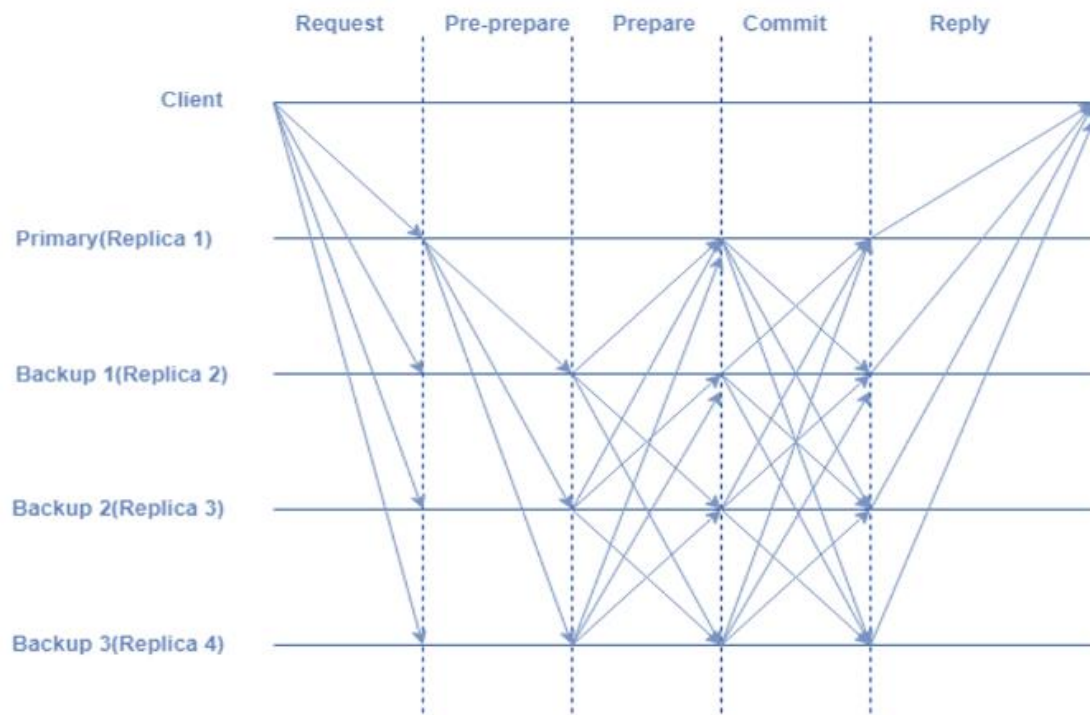
- A PBFT doesn't require carrying out high mathematical computations like Proof-of-Work.
- It is an energy-efficient consensus model.
- A Block of transactions does not need to follow multiple confirmations by each node. Hence, it requires less time.

- As PBFT requires every node to participate and serves the client request, each node gets the reward. Hence low reward variance between each node, and less conflict of interest.

Cons:

- PBFT has a high communication overhead that will increase with the number of nodes in the network.
- It has scalability (can't be too large) issues with more extensive networks.
- PBFT is susceptible to *Sybil hacker attacks* in which one node controls or acts as multiple network nodes.

The below diagram shows the working of the pBFT algorithm.



ARTICLE 9

WHAT IS A DIGITAL SIGNATURE?

KEY TAKE AWAY: A DIGITAL CODED SIGNATURE THAT PROVES ONE’S IDENTITY.

Digital signatures are like electronic “fingerprints.” They are a specific type of electronic signature (e-signature), and signifies the ‘signing’ party to a document or message, is authorized to so sign the document and attests to the documents authenticity and legitimacy and that the ‘signer’ has the legal credentials so approve and ‘sign’ the underlying document. (Sort of like one signing their name to their Will where their signature attests they are the testator/testatrix named in the Will, their identity has been truthfully established and the contents of the Will are approved by the signer as being authenticate and truthful).

In the form of a digital coded key or alpha-numeric string of characters, the digital signature securely and truthfully associates the signer with a document in a recorded transaction. Digital signatures use a universally recognized, accepted format, called **Public Key Infrastructure (PKI)**, to provide the highest levels of security and universal acceptance that the signature is authentic and truthful. PKI involves using a **digital certificate – sort of like conventional hard copy paper third party notarized affidavit or acknowledgment, confirming one’s identity based on the third parties witnessing attestation** (which can be relied upon by other parties that if they receive a digital signature backed up by a digital certificate, they will have high confidence it is authentic because of the issuance of the digital certificate.) Such certificates are typically issued by an independent recognized and publicly acceptable third party **Certificate Authority**, who certifies to the identity and verification of the digital signature (authentication of the identity of the signing party – they are who they say they are). The Certificate Authority can provide various levels or tiers of signature certification (similar to good, better, best type rating) that is dependent on the desired level of authentication. The level of certification is based on the degree of effort a Certificate Authority expended investigating the identity of a party to be certified, when it issues a digital certificate. That effort can vary anywhere to merely confirming a person’s home address and telephone number to investigating public identity documents (such as driver’s license, passport, birth certificate, etc.) or more intense personal contact and face-to-face interviews and review of certain sensitive information such as financial statements or tax returns. In some instances, a party can issue a self-issued digital certificate (for example one can be generated by Microsoft products) which basically means the issuing party has an Internet Protocol address and email, and such self-serving evidence of authenticity of identity may be sufficient, depending on the requesting party needs.

What’s the difference between a digital signature and an electronic signature?

A digital signature is a type of electronic signature that requires a greater rigorous level of identity assurance through digital certificates.

The broad category of **electronic signatures** (*e-signatures*) encompasses many types of electronic signatures (such as a scanned picture or graphic digital representation of a person’s handwritten signature) as well as digital signatures, which are a specific secure technology digitally implementing electronic signatures. Both digital signatures and other e-signature solutions allow a person to sign documents and authenticate their identity (they are who they say they are). However, there are differences in purpose, technical implementation, geographical use, and legal and cultural acceptance of

digital signatures versus other types of ‘written’ e-signatures. Local laws may affect the format and type of e-signature considered to be legally enforceable.

The use of digital signature technology for e-signatures varies significantly between countries that follow open, technology-neutral e-signature laws, including the United States, United Kingdom, Canada, and Australia, and those that follow tiered eSignature models that prefer locally defined standards that are based on digital signature technology, including many countries in the European Union, South America, and Asia. In the European Union, there are two levels of digital signatures: **Advanced Electronic Signature** (AES, an electronic signature which is additionally uniquely linked to and capable of identifying the signatory; created in a way that allows the signatory to retain control; linked to the document in a way that any subsequent change of the data is detectable) and **Qualified Electronic Signature** (QES, is an Advanced Electronic Signature which is additionally created by a qualified signature creation device (QSCD); and is based on a qualified certificate for electronic signatures). QES is more rigorous than AES in regard to certifying to the authenticity of the identity of a signing party. Which one applies depends on the specifications of the entity or person accepting the digital signature.

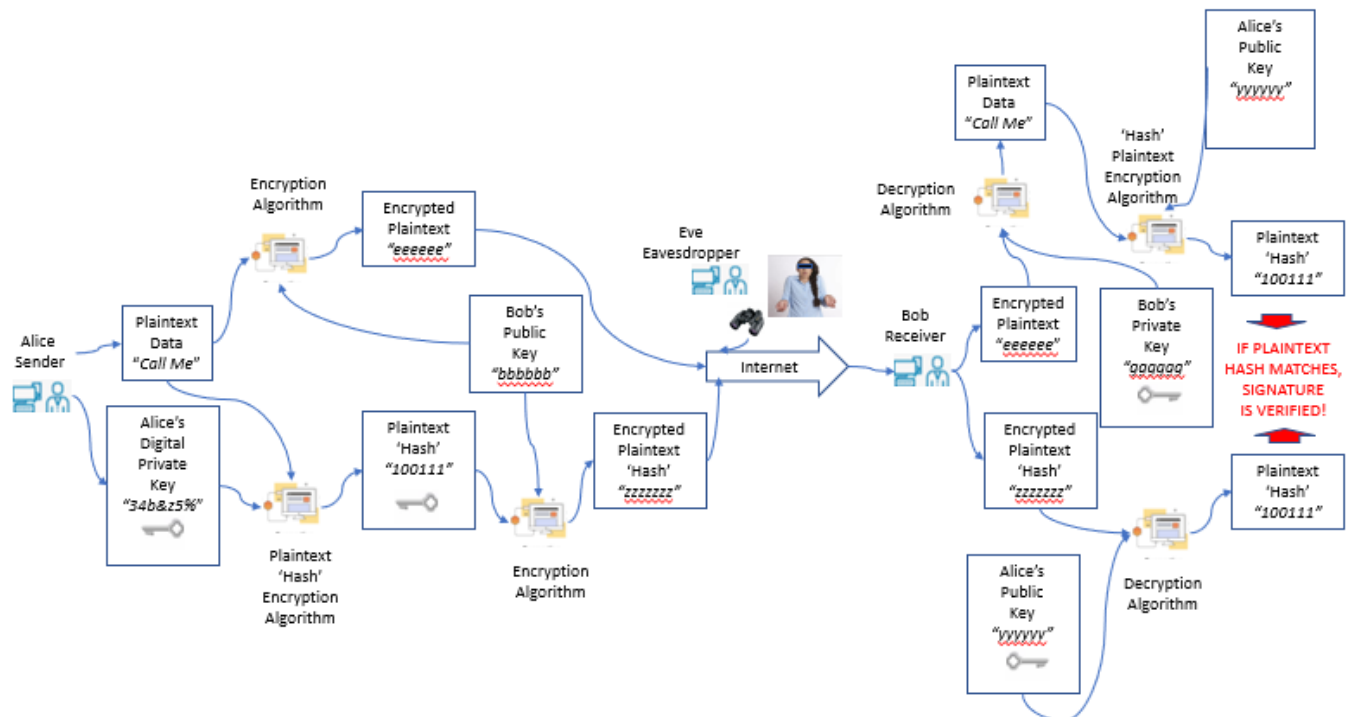
How do digital signatures work?

Digital signatures, like handwritten signatures, are unique (like a ‘fingerprint’) to each signer. Third party digital signature **PROVIDERS**, follow a specific protocol (standards or set of rules and procedures), called Public Key Infrastructure (PKI). PKI requires the PROVIDER to use a cryptographic mathematical algorithm to generate two long alpha-numeric numbers, called keys. One key is public (which is disclosed to the public), and one key is private (which is kept secret by the owner). The PROVIDER provides the services of a recognized and authorized Certificate Authority, who has the authority to confirm the authenticity of a person and issue a digital certificate attesting to that authentication, certifying the person holding the certificate is who they say they are. To protect the integrity of the signature, PKI requires that the keys be created, conducted, and saved in a secure manner, and often requires the keys can only be issued by a reliable Certificate Authority (CA).

As an example of how digital signature works is illustrated in the below diagram and described as follows:

- Assume Alice, sender, desires to send confidential message (“Call me”) along with her Public Key digital signature to Bob the receiver. Bob desires to verify Alice’s digital signature.
- Alice encrypts the confidential plaintext message using Bob’s public key and sends that encrypted ciphertext message to Bob over the internet.
- At the same time, Alice uses her private key to create hash (see Article 7 explaining what a Hash is and how it works) of the plaintext message.
 1. Hash is a ‘one-way’ encryption process. Data can be converted to an unintelligible ciphertext hash code. There is no way to convert the hash ciphertext back to the original plaintext message. Thus Alice’s use of her private key is secure.
- Alice then uses Bob’s public key to encrypt the hashed message and sends this encrypted hash message to Bob over the internet.
- Bob receives the encrypted message (previously encrypted by Alice using Bob’s public key), and uses his private key to decrypt that message to plaintext, he can now read the message.
- Bob uses Alice’s public key to convert the plaintext message to an encrypted hash.
- Bob also received the encrypted Hash message that he decrypts using Alice’s public key. That decryption will result in Bob determining Alice’s Hash value of the plaintext message (but Bob will not know Alice’s private key).

- Bob then compares the two hash values (compare Bob’s hash encrypted plaintext message with Bob’s decryption of the encrypted hash), both of those hash values should be the same and if they are, then Bob has verified the digital signature is authentic, originated from Alice.
- Had the hashes not been the same, Bob will treat the message and/or the digital signature as bogus and proceed with additional verification tactics or bin the entire message.
- This example only allows Eve the eavesdropper to only view encrypted data. The plaintext data is kept secure since Eve does not have access to Bob’s private key. Even if Eve decrypted the encrypted plaintext hash value with Alice’s public key, the result will only be the unintelligible hash which cannot be decrypted any further.



How do I create a digital signature?

Depending upon the Certificate Authority you are using, you may be required to supply specific information. There also may be restrictions and limitations on whom you send documents to for signing and the order in which you send them. When you receive a registration document from a Certificate Authority for signing via email, you must authenticate the document per the Certificate Authority’s instructions and then “sign” the document by filling out a form online. The Certificate Authority uses this registration process to issue a digital certificate, along with the issuance of public and private keys.

What is Public Key Infrastructure (PKI)?

Public Key Infrastructure (PKI) is a set of requirements that allow (among other things) the creation of digital signatures. Through PKI, each digital signature transaction includes a pair of keys: a private key and a public key. The private key, as the name implies, is not shared and is used only by the signer to electronically sign documents. The public key is openly available and used by those who need to validate the signer’s electronic signature. PKI enforces additional requirements, such as requirements to be recognized as an independent third party Certificate Authority (CA), details of digital certificate (how created, how issued), end-user enrollment software used to register and be issued a digital certificate as

well as issuance of public and private keys, and tools for managing, renewing, and revoking keys and certificates.

What is a Certificate Authority (CA)?

Digital signatures rely on public and private keys. Those keys have to be protected in order to ensure safety and to avoid forgery or malicious use. When you receive or send signed documents, you need assurance that the documents and the keys are created securely and that the sender and receiver are using valid keys. Certificate Authorities (CAs), a type of Trust Service Provider, are independent third party organizations that have been widely accepted as reliable for ensuring key security and an entity that can issue digital certificates. Both the entity sending the document and the recipient signing it must agree to use a given CA.

Why would I use a digital signature?

Many industries and geographical regions have established eSignature standards that are based on digital signature technology, as well as specific certified CAs, for business documents. Following these local standards based on PKI technology and working with a trusted Certificate Authority can ensure the enforceability and acceptance of an e-signature in each local market. By using the PKI methodology, digital signatures utilize an international, well-understood, standards-based technology that also helps to prevent forgery or changes to the document after signing.

Are eSignatures, based on digital signature technology, legally enforceable?

Yes. The EU passed the EU Directive for Electronic Signatures in 1999, and the United States passed the Electronic Signatures in Global and National Commerce Act (ESIGN) in 2000. Both Acts made electronically signed digital contracts and documents legally binding, just like paper-based signed contracts.

What is a digital certificate?

A digital certificate is an electronic document issued by a Certificate Authority (CA). It contains the public key for a digital signature and specifies the identity associated with the key, such as the name of an organization. The certificate is used to confirm that the public key belongs to the specific organization. The CA acts as the guarantor of the authenticity of the identity of the entity owning the public key. Digital certificates must be issued by a trusted authority and are only valid for a specified time. They are required in order to create a digital signature.

What is the difference between Advanced Electronic Signature and Qualified Electronic Signature?

AES and QES are digital signature standards regulated by EIDAS, European Union's regulatory framework for electronic signatures that is adopted by all EU member states. EIDAS defines the 3 levels of electronic signatures: electronic signature (sometimes referred to as a "simple" signature, such as a visual or graphic copy of one's handwritten signature inserted in a digital document), advanced electronic signature, and qualified electronic signature, which is the most stringent.

AES adds identity verification. Signatures must be uniquely linked to, and capable of identifying, the signer. Electronic signature records can show evidence of document tampering after it as originally sent.

QES requires face to face identity verification or the equivalent. It's the only form of digital signature that European Union law considers as the equivalent of a handwritten signature. They are often used for high-value, regulated, or cross-border agreements.

An example template for registering for a digital certificate and being issued a public and private key pairs is illustrated by the following internet online registration procedure Secure Shell (an Oracle procedure):

1. Start the key algorithm generation program.

```
myLocalHost% ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/johndoe/.ssh/id_rsa): XXXXXXXXXXXXX
```

2. Enter the path to the file that will hold the key.

By default, the file name `id_rsa`, which represents an RSA v2 key, appears in parentheses. You can select this file by pressing Return. Or, you can type an alternative filename.

```
Enter file in which to save the key (/home/johndoe/.ssh/id_rsa): <Return>
```

The public key name is created automatically and the string `.pub` is appended to the private key name.

3. Enter a passphrase for using your key.

This passphrase is used for encrypting your private key. A good passphrase is 10–30 characters long, mixes alphabetic and numeric characters, and avoids simple English prose and English names. A null entry means no passphrase is used, but this entry is **strongly discouraged** for user accounts. Note that the passphrase is not displayed when you type it in.

```
Enter passphrase(empty for no passphrase): <Type the passphrase>
```

4. Re-enter the passphrase to confirm it.

```
Enter same passphrase again: <Type the passphrase>
Your identification has been saved in /home/johndoe/.ssh/id_rsa.
Your public key has been saved in /home/johndoe/.ssh/id_rsa.pub.
The key fingerprint is:
0e:fb:3d:57:71:73:bf:58:b8:eb:f3:a3:aa:df:e0:d1 johndoe@myLocalHost
```

5. Check the results.

The key fingerprint (a colon-separated series of 2 digit hexadecimal values) is displayed. Check that the path to the key is correct. In the example, the path is `/home/johndoe/.ssh/id_rsa.pub`.

At this point, you have created a public/private key pair.

6. Copy the public key and append the key to the `$HOME/.ssh/authorized_keys` file in your home directory on the remote host. (The public key may be shared with any third party).

Example image of a public key...

```
AAAAB3NzaC1yc2EAAAADAQABAAQBAQC9LYa2dUjIjEIhBwpKfu7DYVZtNSiI
ElggiwzbJGzfdBr5Uz783w1zZ3HTzEkejiSeH/HQDXkEktS1dsW+tkl6xorKK1BXHIFgOpEcT3Ki5K8Ya8HvAFeqrlnHweNkj5NoyXjzIKszVA
yJsqwJTBxulUnhSbotQguWDZ5cLBlSc9+xuZsrI6pKJVSMuQmsA71iOjH04Cg
1aej6YfEJRD/9pZnmhhsZVlI0hIMDOkck4nfi41c7pEsVNJT sp@
```

ARTICLE 10

GENERATING PUBLIC AND PRIVATE KEYS USING THE RSA AND ELLIPTIC CURVE ENCRYPTION ALGORITHMS

(THE RSA NUMBER MODULO p [$n \bmod p$] Factoring
and
Elliptic Curve Discrete Logarithm Problem, Methods)

[HONEST...DON'T RUN TO THE HILLS IF THESE OFTEN USED GEEK BUZZ WORDS ARE DISTURBING, I PROMISE THEY WILL BE CONVERTED TO UNDERSTANDABLE PLAIN ENGLISH]

KEY TAKE AWAY: CRYPTOGRAPHY IS BASED ON MATH EQUATIONS CREATING DIGITAL KEYS.

(Recognition and thanks to Comparitech website resource, used to supplement this Appendix regarding RSA explanation; <https://www.comparitech.com/blog/information-security/rsa-encryption/>)

[WHILE THE BELOW DISCUSSION CONTAINS QUITE ABIT OF ARITHMETIC, IT IS WRITTEN FOR A NON-MATH PERSON TO UNDERSTAND. BE PATIENT, AND IF YOU TAKE THE TIME TO READ THE ARTICLE'S DISCUSSION, USER FRIENDLY EXAMPLES AND GRAPHICS, I AM CONFIDENT THE READER WILL SUCCESSFULLY OBTAIN A SOLID UNDERSTANDING OF SOME OF THE BASIC PRINCIPLES BEHIND THE MYSTERY OF HOW PUBLIC AND PRIVATE DIGITAL CRYPTOGRAPHIC KEYS ARE GENERATED AND USED – AND THIS WILL ADD IMMENSELY TO THE UNDERSTANDING OF BLOCKCHAIN TECHNOLOGY]

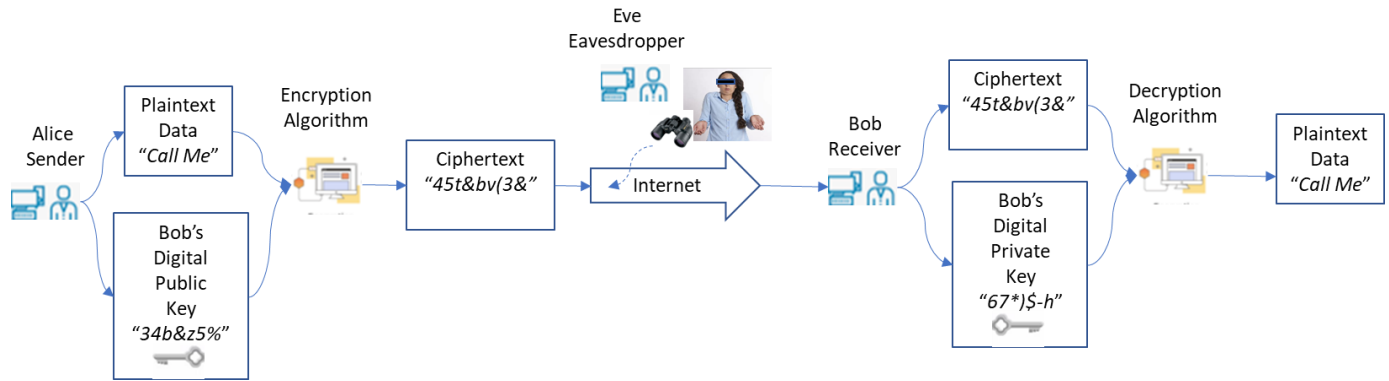
There are many ways to securely generate or determine:

- digital cryptographic **public keys** (used in algorithms to encrypt data, converting plaintext to unintelligible ciphertext, a public key is published and viewable by anyone); and
- digital cryptographic **private keys** (used in algorithms to decrypt data, converting unintelligible ciphertext back into plaintext, a private key is kept secret and only known by its owner).

The process of encrypting data is illustrated in the below simplified diagram which encompasses:

- a sender (Alice) of confidential data, using another party's (Bob the receiver of the confidential data) public key (a large alphanumeric digital 'number' whose characters are made up of numbers and letters and possibly symbols, example $34b\&z5\%$... – might have 256 characters) inputs Bob's public digital key and the plaintext message into an agreed algorithm computer program to encrypt plaintext data into ciphertext (example plaintext "Call Me" converted by the algorithm into unintelligible ciphertext $45t\&bv(3\&)$),
- Alice then sends the unintelligible coded ciphertext data ($45t\&bv(3\&)$) to Bob, over the internet to protect it from prying eyes (Eve the eavesdropper), and

- the receiver (Bob) of the coded message, reverses the process by reconvertng the unintelligible ciphertext data ("45t&bv(3&"), back into understandable plaintext ("Call Me") by Bob inputting his private digital key (67*)\$-h") and the ciphertext into the agreed decryption algorithm.
- The encryption and decryption processes can be thought of as using a special lock on a door used to secure a room. The door lock works as follows:
 - Alice wants to lock the special lock installed on the door and only allow Bob to open it. Alice uses a special public key given to her by Bob to lock the special lock. The special door lock can then only be unlocked by Bob using a separate special secret private key, no other key will unlock the door. Both keys are required to lock and unlock the special door lock.
 - Alice can permit another person to unlock the special door lock but to do so she will need their public key to first lock the special door lock, then the other person using their secret private key to unlock it. Bob can't unlock the special door lock and open the door since his public key was not used to lock the special door lock. The special lock (comparable to an encryption algorithm) must use the 'paired' (linked) public and private keys to lock and unlock the door (comparable to encrypting and decrypting data).



Special lock using 'paired' public and private keys both required to lock and unlock the lock...



Two often used computer program algorithms used to generate cryptographic digital key pairs are known as the **RSA algorithm** and **elliptic curve algorithm**. Each of these will be discussed below (in plain English) because understanding how these cryptographic algorithms work and what they do, is important to help understand how Blockchain technology works since it is dependent on cryptography to function.

- RSA is used to generate public and private keys AND encryption.

- The elliptic curve method only generates public and private keys and those keys used in a separate encryption/decryption algorithm.

To begin our journey, an important and often used cryptographic mathematical process used by both RSA and elliptic curve cryptography is called the **Discrete Logarithm Problem**.

What is the Discrete Logarithm Problem?

First...refresher on logarithms...

In mathematics, for any given **real number**⁴⁶ (any whole positive number not including 0 such as 1, 5, 202, etc., in contrast to **Integer numbers** and **Whole numbers**) such as letting “a” = 100 and “b” = 10, the logarithm of a real number “a” is written as [logarithm_b (a)] or more commonly, $\log_b a = x$, which means the logarithm of the real number “a” is defined as the exponent number, represented as “x”, that a ‘base’ number is raised to that results in determining the number “a”, or in arithmetic terms, $b^x = a$.

\log_b is referred to as “log base b”. If b the base is assumed to be 10, then \log_b means “log base 10” or \log_{10} . For example, if “b” = 10 (the base number), “a” = 100 (the desired real number) and $x = 2$ (the base number exponent); then, $b^x = a$ or 10^2 (10×10) = 100 or \log_b of a = x or $\log_{b=10}$ of a = 100 is $x = 2$ (x is the exponent or power the base number 10 is raised to, because 10 squared or raised to the 2 power equals 100, $10 * 10 = 100$). (The logarithm x is the exponent or power a base number (such as 10) is raised, to achieve determining the resultant answer).

As an aside, the logarithm of a negative number is determined by first taking the logarithm of the positive number (called taking the logarithm of the ‘magnitude’ of the number) and multiplying that answer by “-1”. For example, the $\log_{10} -100 = (-1)\log_{10}(100) = -2$. Same as $(-1)[10^2] = -100$

Trust that explanation of logarithms was not too painful and makes sense. Later we shall see why a short refresher of high school logarithm arithmetic is important for cryptography.

Logarithms are useful to simplify complex mathematical calculations, since by converting a number to an exponent, adding and multiplying exponents and then converting back into the real number answer, is often an easier task.

A simplified illustrative example of the usefulness of logarithms follows, lets say we want to multiply two large numbers together:

1 million times 1 billion or $1,000,000 * 1,000,000,000$. While we can do this by hand or on a calculator, and the result is 1,000,000,000,000,000, using logarithms, the calculation is simpler and as follows:

$\log_{10} 1,000,000 = 6$ (x exponent = 6, since $10^6 = 1,000,000$) and $\log_{10} 1,000,000,000 = 9$ (x exponent = 9, since $10^9 = 1,000,000,000$). When logarithms are multiplied together you add the exponents: $(\log_{10} 1,000,000) * (\log_{10} 1,000,000,000) = (6 + 9 = 15)$. Since $a = 10^x$, $a = 10^{15}$, or 1,000,000,000,000,000, the

⁴⁶ **Real numbers** are any whole **positive** numbers **not including zero** (examples, 1, 4, 10, but not fractions or decimals, 12.5 is not a real number). In contrast, **Integer numbers** are any **positive or negative** whole numbers **including 0** (examples, -3, 0, 5, not a fraction or decimal) and **Whole numbers** are any real **positive numbers, including zero** (examples, 0, 1, 4, 10, not a fraction or decimal).

same answer we obtained when doing the calculation, the long and harder way (the long and harder 'hand' calculation way is referred to as a 'naïve' calculation).

Any base "b" number can be used to determine logarithms and their exponents. The most commonly used bases are base 10 (common logarithm, \log_{10}) and base e (natural logarithm, \ln_e or just \ln)⁴⁷. For example, if "a" equals 100, $\log_{10} 100 = 2$ ($100 = 10^2$) and $\ln_e 100$ or $\ln 100 = 4.605170186$ ($100 = e^{4.605170186}$). These logarithm concepts are used in cryptography.

- In any group of numbers G (positive, negative, whole or decimal numbers, including 0, such as: -0.01, -2, 0, 1, 1000),
- **each member** number in Group G, represented by G_i where i represents the individual number in a Group (example: In Group of numbers G_i ; $G_1 = -0.01$, $G_2 = -2$, $G_3 = 0$, $G_4 = 1$, $G_5 = 1000$)
- can be determined by raising a base number "b" to an exponent power to obtain the number, or
- $G_i = b^x$.
 - (Example: if $b_{base} = 10$ and $G_5 = 1000$, then $G_5 = b^x$; or $1000 = 10^2$ or $x = 3$).
 - In logarithm speak: $\log_b a = x$, or $\log_{10} 1000 = 3$; or $b^x = a$ or $10^3 = 1000$

Second...refresher on modular arithmetic

In mathematics, **modular arithmetic** is a system of arithmetic for integers (an integer is any positive or negative whole number including zero, such as -2, 0, 12), where the numbers "wrap around" when reaching a certain value, called the modulus.

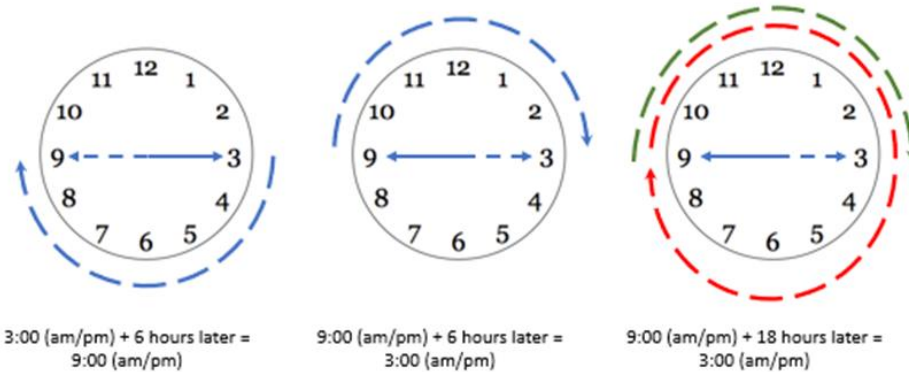
What this means is to consider a familiar use of modular arithmetic is referring to the 12-hour clock, in which the day is divided into two 12-hour periods.

Example 1: If the time is 3:00 now (a.m. or p.m.), then 6 hours later it will be 9:00 (a.m. or p.m.) (see diagram below).

Example 2: If the time is 9:00 (a.m. or p.m.) and 6 hours later, simple addition would result in $9 + 6 = 15$, but clocks "wrap around" every 12 hours, so the clock answer would be 3:00 (a.m. or p.m.) – resulting in 3 hours having 'wrapped' around the 12 hour clock after reaching 12 and starting the clock counting process all over again.

Example 3: Taking the same example, if the time is 9:00 (a.m. or p.m.) and 18 hours later, simple addition would result in $9 + 18 = 27$ hours, but clocks "wrap around" every 12 hours, so the clock answer would be 3:00 (a.m. or p.m.)

⁴⁷ The number e is a special number used in mathematics, also known as Euler's number (the famous historical mathematician who discovered the number), is a mathematical constant approximately equal to 2.71828.... It is the base of the 'natural' logarithms. It is determined by the equation and taking the limit – assessing its extreme value - of $(1 + 1/n)^n$ as n approaches infinity. $e = (1 + 1/\infty)^\infty$. Probably more than you need to know...



In these examples if you were using a 24 hour (military time) instead of a 12 hour clock (a.m. and p.m.) the 0900 (p.m.) + 1800 = 2700 military time is the same as 0300 or 3:00 a.m.

Because with a 12 hour clock, the hour number starts over at zero when it reaches 12, this is what is called **arithmetic modulo 12**.

In modular arithmetic speak, the above three clock examples would be expressed as follows:

Example 1: 9 is congruent to 9 modulo 12

Example 2: 15 is congruent to 3 modulo 12

Example 3: 27 is congruent to 3 modulo 12

What this means in simple terms is

- Y is congruent to x modulo z means
- to take the number z and
- determine how many even times it divides into Y and
- the remainder is x.

For example 1: 12 (z) divided into 9(Y) goes 0 even times, and the remainder is $(9 - 0 * 12, \text{ remainder } 9)$ or 9:00

For example 2: 12 (z) divided into 15(Y) goes 1 even time, and the remainder is $(15 - 1 * 12 = 3)$ or 3:00

For example 3: 12 (z) divided into 27(Y) goes 2 even times, and the remainder is $(27 - 2 * 12 = 3)$ or 3:00.

Further examples:

- What is the modulus of 3 raised to the 29th power(Y) mod 17(z) (modular math, which means what is the remainder when 3^{29} is divided by 17, or 3^{29} is congruent to x mod 17, x (the remainder) = ?; answer: 12 (the remainder since $3^{29}/27$ has a remainder of 12).
- A convenient online modulo calculator is <https://www.symbolab.com/solver/modulo-calculator> (3^{29} is too large a number for Microsoft Excel mod (modulus) formula to handle, hence use of the online symbolab modulo calculator).



- This modulus (remainder) calculation is straight forward and easy to determine when the exponent x is known and looking for the remainder.
- However, if the example is reversed and 12 (the remainder) is known but not the whole number (or discrete) exponent x : $3^{x?} \bmod 17 = 12$, it becomes apparent that solving for the whole number discrete x exponent is very difficult if not impossible, since there are many (infinite?) solutions to this problem, since there are many x 's that can result in a remainder of 12 because in this example 17 can be divided evenly into many numbers that will result in a remainder of 12). Raising a number by an exponent, is connected to the logarithm of a number, hence the discrete **logarithm** problem branding.
- **Defining a mathematical procedure that is easy to calculate in one direction but very difficult (if not impossible) to reverse the process, is referred to as the discrete logarithm problem (or **trapdoor function** – easy to fall through, tough to get out).** This discrete logarithmic problem result is used in cryptographic algorithms for determining secure cryptographic keys.
- **A discrete logarithm scheme is extremely difficult to find the input from the result (output).**

Discrete logarithm_b of G_i or $\log_b G_i = x$ (or $G_i = 10^x$ if $b_{base} = 10$) is defined as that power or exponent “ x ”, determined by the logarithm, must be a ‘discrete’ or **whole integer number**⁴⁸ (such as $x = “1”$ or “-3” or “0”, but not, any number with a decimal such as $x \neq 1.5$). This same discrete number restriction can be applied to modular arithmetic where, $b^x \bmod p$, and x must be a discrete whole positive number. And if x is a non-negative number and b is a whole number then b^x will be a whole discrete number.

Discrete whole number examples of the powers of base 10 are: 10^{-3} (or $[1/1000] = 0.001$, $10^{-2} = 0.01$, $10^{-1} = 0.1$, $10^0 = 1$, $10^1 = 10$, $10^2 = 100$, $10^3 = 1000$, etc. Thus the exponent power logarithm_{base10} (“ x ”) of each of the positive or negative or decimal numbers are discrete, whole positive or negative, including 0, numbers:

$\log_{10} 0.001 = -3$, $\log_{10} 0.01 = -2$, $\log_{10} 0.1 = -1$, $\log_{10} 1 = 0$, $\log_{10} 10 = 1$, $\log_{10} 100 = 2$, $\log_{10} 1000 = 3$; or exponent power integer x equals the discrete logarithm exponent whole positive or negative numbers: -3, -2, -1, 0, 1, 2, 3, etc.

Other base-10 logarithms in the G_i list of real numbers⁴⁹ (positive or negative whole numbers but not 0) may not be instances of the **discrete logarithm problem**, because they involve non-integer exponents that have a non-discrete decimal value. For example, the $\log_{10} 53 = 1.724276\dots$ or $10^{1.724276\dots} = 53$, but $x = 1.724276$, a non-integer decimal number, and thus not a discrete (whole negative or positive number) logarithm.

⁴⁸ See footnote 6

⁴⁹ See footnote 6

In cryptography, it is a fundamental requirement that when determining a cryptographic key, the discrete logarithm problem be applied to ensure only discrete whole positive exponent logarithm numbers, “x”, be determined. No decimals are allowed!

Cryptographic applications take advantage of the fact that discrete logarithm problems are hard (difficult) to compute and thus cryptographic digital keys based on the discrete logarithm problem will be difficult or hard if not impossible to crack. This hardness test can be used to generate secure cryptographic digital keys using the discrete logarithm problem process.

The discrete logarithm problem procedure uses, **modular arithmetic** (expressed as $x \bmod p$, also referred to as a *number (n) modulo p*, which just means finding the **remainder** of the process of dividing n (the *number*) by p (the *modulus*), where n/p and its remainder are used to develop a cryptographic discrete logarithm problem process.

An $n \bmod p$ example, assume the mathematical modular expression: $n \bmod p$, has $n = 46$ (the number) and $p = 12$ (the modulus); then $46 \bmod 12$ means divide p, the modulus (12), into n, the number (46), which equals 3 even times ($36 = 3 * 12$) and the remainder⁵⁰ is equal to 10 or ($46 - 3*12$), the modular whole discrete integer number ‘remainder’ answer.

Applying this number **n modulo p** discrete logarithmic problem process to cryptography (describing a process that is difficult to solve, making its use to secure digital keys very useful)...

- First, define a prime number (a number only divisible by 1 or itself, for example 17, a positive number) as the modulus or p;
- Next find the **primitive root** of the prime modulus number p, called the **generator**, in this case the primitive root of 17 is 3
 - a primitive root of a prime number is the **smallest** whole number divisible into the prime number that will result in the remainder being a whole number integer, example the root 5 can equally be divided into 17, 3 times leaving a whole number remainder of 2, but the smaller and lowest number, 3, the primitive (smallest) root, can also be divided into 17 resulting in a whole number remainder of 2;
- An unusual property of using a prime number modulus and its primitive root,
 - if you raise the primitive root number to any whole positive integer (x) power,
 - the resultant $n \bmod p$ *remainder* of the prime number will always be a whole number ranging between 0 and maximum of the prime number or in this example between 0 and 17.
 - For example, $3^x \bmod 17$, results in the following modulo table results with various x exponent values (whole number integer powers), noting the whole number remainder of $3^x \bmod 17$, does in deed lie between whole numbers 0 and 17:

⁵⁰ More than you need to know, but if two different numbers, n1 and n2, using the same modulus p, result in both of them giving the same remainder, the duplicate remainder answer result is known as *modulus congruence*. For example, for both n1 and n2 numbers, 11 and 16, have the same remainder (are congruent) after the same modular (mod 5 example): $11 \bmod 5$ has a remainder of 1 ($11/5 = 2, 11 - 2*5 = \text{remainder } 1$); $16 \bmod 5$ also has a remainder of 1 ($16/5 = 3, 16 - 3*5 = \text{remainder } 1$). Another way of writing congruence is $11 \equiv 16 \pmod{5}$, which means $11 \bmod 5 (=1)$ is identical to $16 \bmod 5 (=1)$ because both modulo functions give the identical, same remainders.

$a^x \bmod p; a=3; p=17$			
x	3^x	$3^x \bmod 17,$ remainder, R (Range: 0 to 17)	
0	1	1	
1	3	3	
2	9	9	
3	27	10	
4	81	13	
5	243	5	
10	59049	8	
15	14348907	6	
20	3486784401	13	

- The reverse procedure (trying to determine, x the exponent power when the modulo or R remainder is known, is very difficult.
 - Given a remainder or modulo R, find the exponent, or x (?) a primitive prime root of a prime number needs to be raised to?
 - For example, $a^{x^?} \bmod R$, example $3^{x^?} \bmod 12$. Finding x is extremely difficult, and is the example of the $n \bmod p$ discrete logarithm problem (a trapdoor, **easy to calculate in one direction but very difficult to reverse the process**, – easy to fall through, tough to get out).
 - An example illustration follows of the easy one way and difficult the other....



- The difficulty or hardness of the discrete logarithm problem (determining the whole positive number exponent x), becomes quite taxing, for $g^x \bmod p$, when the modulus p prime number is say 100's of digits long, it could take thousands of years to trial and error all potential answer possibilities. The strength (or hardness, 'high entropy') of a one-way function is based on the time it takes to reverse it.
- Consequently, digital cryptographic key numbers can be generated as an x exponential function possessing the quality of being secure and not easily crackable by a malicious intruder.

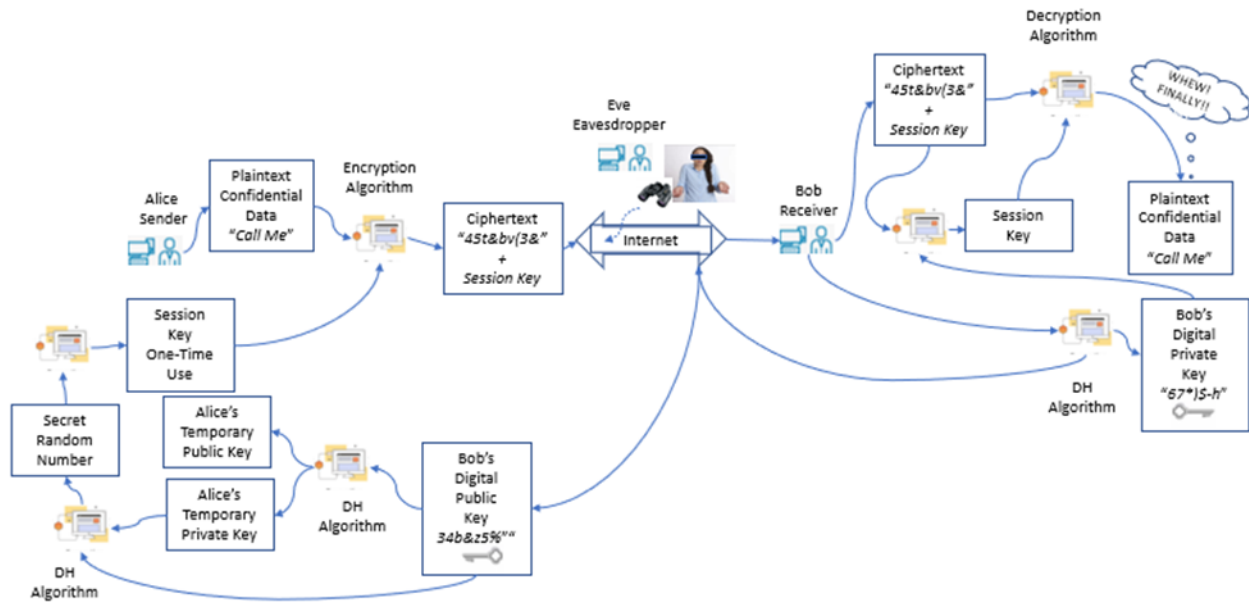
RSA Encryption and Use of the Discrete Logarithmic Problem For Securing Cryptographic Keys

Under RSA asymmetric encryption, plaintext data is first encrypted into ciphertext with a code called a **public key**, then second the **ciphertext decrypted by another key, known as the private key**.

Each RSA user (sender and receiver, Alice and Bob) has a key pair consisting of their public and private keys (and these one-of-a-kind key pairs are uniquely linked together by mathematical computations, thus when Alice, using Bob's public key, sends encrypted data to Bob the receiver who can decrypt the message using his private key, Alice will own her own unique public and private keys and Bob will likewise own his unique public and private keys).

The process of encrypting data is illustrated in the below diagram which encompasses:

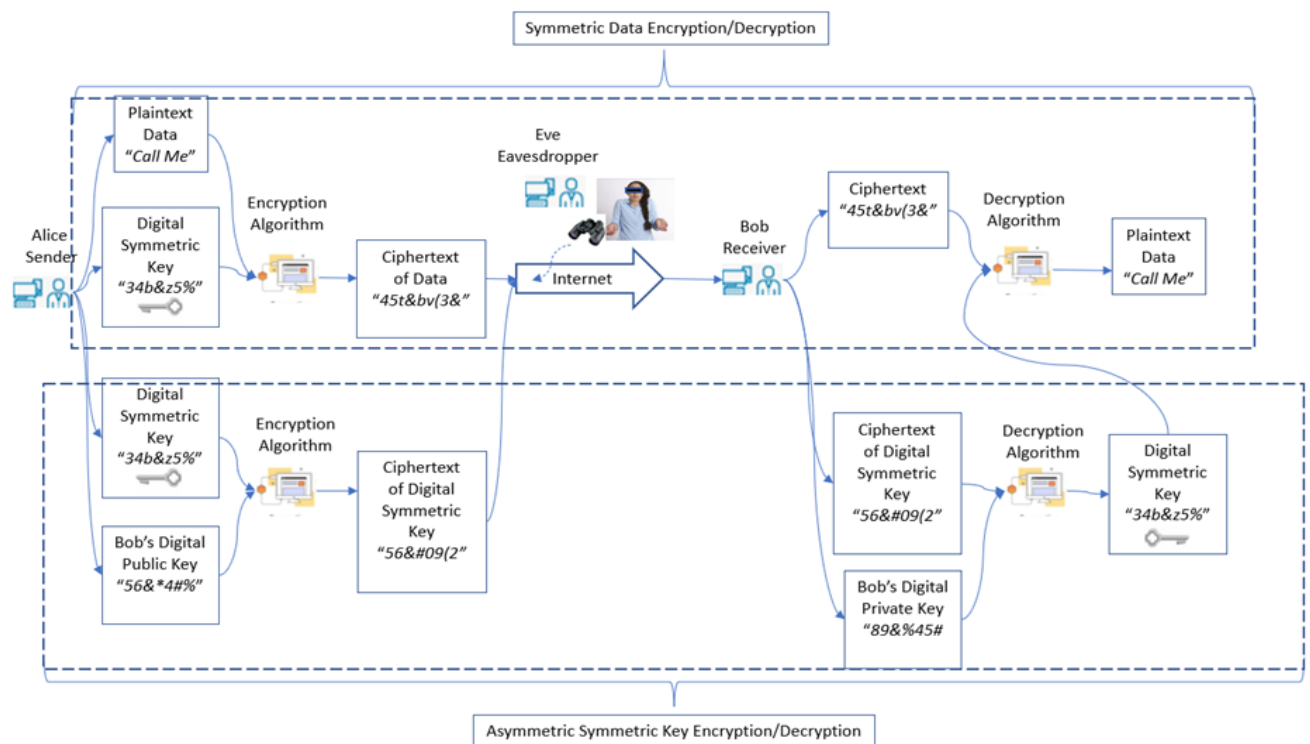
- Bob a receiver of confidential data, uses what is called a Diffie-Helman (DH) cryptographic algorithm to generate his pair of public and private keys;
- Bob sends over the internet his public (non-secret) key (a large alphanumeric digital 'number' whose characters are made up of numbers and letters and possibly symbols, example *34b&z5%...* – might have 256 characters) to Alice;
- Alice a sender of confidential data to Bob, uses Bob's public key and the DH algorithm to create a temporary private key and a temporary public key for herself – thus because Alice used Bob's public key to create her temporary public and private keys, all three keys are linked or **derivative** of each other;
- Alice uses her newly created private key and Bob's public key with the DH algorithm to generate a secret random number key;
- Alice uses the newly created secret random number key (and not a random number obtained from a generator algorithm such as RandomNumberGenerator-RNG or PsuedoRandomNumberGenerator-PRNG, algorithms) to generate with the DH algorithm a special one-time use *session* key for a transaction between Alice and Bob;
- Alice uses the session key (sort of like a temporary public key) to encrypt the confidential data (plaintext "*Call Me*" converted by an encryption algorithm into unintelligible ciphertext "*45t&bv(3&")*) and protected from prying eyes, and then sends the encrypted data along with the encrypted session key to Bob over the internet. The encrypted quasi 'public' session key can be disclosed to prying eyes – but useless to use without an accompanying private key;
- When Bob receives the encrypted message and session key, he can derive the secret session key, which was derived from Alice's temporary private key, with Bob's private key, because Alice's temporary private key and Bob's private key have a common link or derivative – Bob's public key;
- With the secret session key pulled out of the encrypted message, Bob can now decrypt the message; reverses the process by reconverting the unintelligible ciphertext data ("*45t&bv(3&")*), back into understandable plaintext ("*Call Me*") by Bob inputting the session key and the ciphertext into the agreed DH algorithm.



Where is RSA encryption used?

RSA encryption is often used in **combination with other encryption schemes**, or for **digital signatures** which can prove the authenticity and integrity of a message. It isn't generally used to encrypt entire messages or files, because it is less efficient and more resource-heavy than symmetric-key encryption (where only one key is used to encrypt and decrypt data).

To make things more efficient, **a file will generally be encrypted with a symmetric-key algorithm (only one key is used to encrypt and decrypt data)**, and then **only the symmetric key, but not the data, will be encrypted with RSA asymmetric encryption (two pair keys, public and private, are required for asymmetric encryption)**. Under this process, only an entity that has access to the RSA private key will be able to decrypt the symmetric key. See below illustration of this symmetric/asymmetric encryption process.



The above simplified flow of encryption activity is as follows:

- Alice the sender of confidential plaintext data, “Call Me” encrypts the data with a digital symmetric key (not her private key, could be her public key – though this key is available to the public and an eavesdropper might guess that key to decrypt the coded message - or any other digital key selected by Alice, she selected plaintext digital symmetric key “34b&z5%”) into ciphertext, “45t&bv(3&”.
- The ciphertext message (“45t&bv(3&”) is transmitted over the internet to Bob the receiver but not the digital symmetric key used by Alice. Bob cannot decrypt that ciphertext message without knowledge of the symmetric key Alice used to encrypt it.
- Alice uses Bob’s public key (“56&*4#%”) she obtained from Bob (can also obtain from the internet since Bob’s public key is available to the public) to encrypt the digital symmetric key data (“34b&z5%”) into a ciphertext digital symmetric key (“56	(2”).
- Alice sends over the internet to Bob, the ciphertext digital symmetric key (“56	(2”).
- When Bob receives the ciphertext digital symmetric key (“56	(2”), encrypted with his public key (56&*4#%), he can decrypt that digital symmetric key ciphertext (“56	(2”) using his private key (“89&%45#”), revealing the plaintext symmetric key (“34b&z5”).
- Bob can now input the ciphertext message (“45t&bv(3&”) and the decrypted symmetric key (“34b&z5%”) into his decryption algorithm, and reveal the plaintext message (“Call Me”).
- All the while any eavesdropper such as Eve will only see over the internet only the unintelligible ciphertext data (45t&bv(3&”) and symmetric digital key ciphertext data (“56	(2”).

- The efficiency in all this is that the plaintext confidential data is only encrypted with a one key symmetric key, which takes less computing power and energy to encrypt and decrypt data, than using a two key asymmetric encryption process for the plaintext confidential data.

Without being able to access the symmetric key, the **original ciphertext file can't be decrypted**. This method can be used to keep messages and files secure, without taking too long or consuming too many computational resources.

As one of the first widely used public-key asymmetric encryption schemes, RSA laid the foundations for much of today's secure communications.

The background of RSA encryption

Early RSA encryption work was conducted by the United Kingdom intelligence agency, the Government Communications Headquarters (GCHQ). Partly due to technological limitations, the GCHQ couldn't see a practical application use for public-key asymmetric cryptography at the time, so the development sat idly on the shelf gathering dust. **It wasn't until 1997 that the work was declassified and the original inventors of RSA were acknowledged.**

How does RSA encryption work?

The following discussion will **simplify RSA concepts, illustrated by using much smaller number example calculation assumptions**. In reality, RSA encryption uses prime numbers (numbers that are only dividable by 1 or itself) that are much much larger in magnitude and other complexities, and this largeness in cryptographic digital key numbers and complexity, required to ensure the security of transmitted data and information. That's code for stopping malicious third parties such as hackers from successfully attacking and cracking a digital system, resulting in the theft of sensitive information or unauthorized disclosure of confidential information.

There are three important mathematical concepts used in RSA:

- (1) **trapdoor functions,**
- (2) **generating primes,**
- (3) **Carmichael's totient function**

and the separate processes involved in **computing and generating digital public and private asymmetric keys** used in the encryption and decryption processes.

Trap door functions

RSA encryption works under the premise that the encryption algorithm is **easy to compute in one direction, but almost impossible in reverse (like a trap door easy to fall into, complicated to escape) our old friend, the discrete logarithm problem**. As an example, if you were told that 701,111 is a product of two prime numbers (a number that is only divisible by 1 or itself), would you be able to figure out what those two numbers are?

A tough assignment even with a calculator or computer, but if we flip things around, it becomes much easier. What's the result of: 907×773 ?

Generating Primes – Seeking Randomness and Unpredictability

The trap door function mentioned above forms the basis for how public and private-key cryptographic encryption schemes work. **Their properties allow public digital keys to be shared openly to the public without endangering an encrypted message or revealing the private digital key.** The trap door function also allow data to be encrypted with one (prime number) key in a way that can only be decrypted by the other (prime number) key from the pair, since the pair are related or linked to one another (the private key is determined mathematically from the public key as illustrated below) in determining a target prime number when the pair are multiplied together.

The first step of encrypting a message with RSA is to **generate a pair of related digital keys, the public and private prime number digital keys.** To do this, we need to start with **two prime numbers (p_1 and p_2)** which are selected with a **primality test.**

A **primality test** is an algorithm that *efficiently* finds prime numbers, such as the Rabin-Miller primality test (which finds large prime numbers based on probabilistic analysis – more likely than not a target large number is prime). Determining large prime numbers is not easy to do, thus computers are used to generate large prime numbers. Computer software has the ability to randomly (sort of) compute random numbers but the more secure random number generators are called pseudorandom generator or PRNG.

RSA Random Number

The selected p_1 and p_2 random prime numbers used in RSA need to be very large, and also relatively far apart. Prime numbers that are small or close together are much easier to crack (discover).

The below example will use much smaller p_1 and p_2 prime numbers to make things easier to follow and compute with the objective of explaining how RSA works.

Let's say that the primality test gives us the p_1 and p_2 prime numbers that we used above, 907 and 773 (each can only be divided by 1 or by itself since no other number can be divided evenly into these numbers). The next step is to discover the modulo (n) ... (remember our above modular arithmetic discussion)...

[modulo is the mathematical process of dividing a number by another number, and the remainder is the answer, for example, n modulo p , where n is the dividend and p is the divisor, such that if n is 21 and p is 5, 21 modulo 5 , means 5 divided into 21 will go 4 even times with a remainder of 1, thus the modulo answer is 1]

Applying the following formulas:

$$p = p_1 \times p_2; \quad \text{Where } p_1 = 907 \text{ and } p_2 = 773$$

Therefore: $p = 907 \times 773; p = 701,111$ (the divisor in the modulo formula) or n modulo $701,111$.

Carmichael's totient function

Once we have p , the modulo divisor, we use **Carmichael's totient function:**

$$\lambda(p) = lcm(p_1 - 1, p_2 - 1)$$

$\lambda(p)$ represents *Carmichael's totient* for p , where *lcm* means the “lowest common multiple”, which is the lowest number that both the prime numbers p_1 and p_2 can divide into. (We already know the answer for p_1 and p_2 which is 701,111; but now we subtract 1 from each prime number p_1 and p_2 , to determine new prime numbers). Putting our numbers into the Carmichael totient equation:

$$\lambda(701,111) = lcm(907 - 1, 773 - 1)$$

$$\lambda(701,111) = lcm(906, 772)$$

Using an internet online Carmichael totient lowest common multiple calculator (example: <https://comnuan.com/cmnn02/cmnn02006/cmnn02006.php>) gives the answer for the lowest common multiple:

$$\lambda(701,111) = 349,716$$

Generating the public key

Now that we have Carmichael's totient of our prime numbers, it's **time to figure out the public key**. Under RSA, **public keys are made up of a prime number e , as well as modulus**. The number e can be anything between 1 and the value for $\lambda(n)$, which in our example is 349,716.

Because the public key is shared openly, it's not so important for e to be a random number. In practice, e is generally set at **65,537**, because when much larger numbers than 65,537 are chosen randomly, it makes encryption much less efficient (more computer computation capacity and energy is used when using larger numbers). We will keep the numbers small in this discussion to make this illustrative calculation example easier to follow. Let's say:

$$e = 11$$

Our target encrypted data is called the ciphertext (c , *plaintext data already encrypted*). We derive it from our plaintext message (m), by applying the public key with the following formula:

$$c = m^e \bmod n \quad [\text{ciphertext} = \text{plaintext}^e \bmod n] \quad (\text{the plaintext is converted from non-number characters to number characters}).$$

[$e \bmod n$] **expression is the public key**. We have already devised e and we know n as well. The only thing we need to explain is *mod*, the **modulo operation**, which as earlier defined means the remainder left

over when you divide one side by the other. For example: $10 \bmod 3 = 1$. This is because 3 goes into 10 three equal times, with a remainder of 1.

Now assume **the message (m) that we want to encrypt and keep secret is just a single number, 4**. Let's plug everything in:

$$c = m^e \bmod n \text{ (} m^e \text{ means } m \text{ raised to the } e \text{ power, like } 2^2 = (2 \times 2 = 4)\text{)}$$

$$c = 4^{11} \bmod 701,111$$

$$c = 4,194,304 \bmod 701,111$$

To make the **modulo operation** easy, we will be using an online calculator (<https://www.calculatorsoup.com/calculators/math/modulo-calculator.php>). By entering $a = 4,194,304$ and $b = 701,111$, into the online modulo calculator, it gives us:

Mod / Modulo Calculator English

When you divide a number 'a' by 'b', it can be expressed as 'a mod b' which is the remainder. This is also called as the modulus. Use this mod / modulo calculator to perform the mod operation and find the remainder of the division with ease.

Enter the number (a)	<input type="text" value="4194304"/>
Enter the number (b)	<input type="text" value="701111"/>
Calculate Reset	
Mod value of (a%b)	<input type="text" value="688749"/>

$$c = 688,749$$

Therefore when we use RSA to encrypt our message, **4**, with our public key [$e \bmod n$], it gives us the **ciphertext of 688,749**.

In summary, we had a plaintext **message of 4**, which we wanted to keep secret. We applied a public key [$e \bmod n$] to it, which gave us the **encrypted result of 688,749 (c)**. Now that it is encrypted, **we can securely send the number 688,749 to the owner of the key pair (Bob)**. They are the only person who will be able to decrypt it with their private key. When they decrypt it, they will see the plaintext message that we were really sending, **4**.

Generating the private key (which is mathematically linked to the public key as shown below)

In RSA encryption, once data or a message has been turned into ciphertext with a public key, it can only be decrypted by the private key from the same digital public and private key pair (using the same algorithm).

Private keys are comprised of "d" and "n". We already know n , and the following equation is used to find d :

$$d = 1/e \text{ mod } \lambda(n)$$

In the **Generating the public key** section above, we already decided that in our example, e would equal 11. Similarly, we know that $\lambda(n)$ equals 349,716 from our earlier work under **Carmichael's totient function**. Things get a little more complicated when we come across this section of the formula: $1/e \text{ mod}$

This equation may look like it is asking you to divide 1 by 11, but that's not the case. Instead, this just **symbolizes** that we need to calculate the **modular inverse of e** (which in this case $e = 11$) and $\lambda(n)$ (which in this case is 349,716).

This essentially means that **instead of performing a standard modulo operation, we will be using the inverse instead.**

Understanding Modular Inverses

What is an inverse?

A number multiplied by its inverse equals 1. From basic arithmetic we know that:

The inverse of a number A is $1/A$ since $A * 1/A = 1$ (e.g. the inverse of 5 is $1/5$ and $5 * 1/5 = 1$)

All real numbers other than 0 have an inverse (can't divide by 0, $1/0 = \infty$, an undefined number). Multiplying a number A by the inverse of A is equivalent to dividing by A (e.g. $10/5$ is the same as $10 * 1/5$).

What is a modular inverse?

In modular arithmetic we do not have a division operation. However, we do have **modular inverses**. The modular inverse of $A \pmod{C}$ is $A^{-1} = A^{-1} = (1/A^1) = (1/A)$

$(A * A^{-1}) = (A/A) \equiv 1 \pmod{C}$ or equivalently $(A * A^{-1}) \pmod{C} = 1$, (the remainder is 1)

Math truism: Only the numbers co-prime to C (numbers that share no prime factors with C other than 1) have a modular inverse \pmod{C} . Wow! What tha' hec' is that? Stay tuned...

A prime number is defined as a number that has no factor other than 1 and itself. In contrast, co-primes are two prime numbers considered in pairs and are co-prime if they have no common factors other than 1. That is 1 is the only and Highest Common Factor (HCF).

As an example, 18 and 35 are co-prime numbers because, the factors of 18 are 1, 2, 3, 6, 9, and 18 (each will divide into the base number such that there is no remainder) while the factors of 35 are 1, 5, 7, and 35. Since the Highest Common Factor or HCF is 1 and no other higher factors match, they are coprime. **Yeah! Not bad so far...**

In contrast 18 and 40 are not co-prime numbers because, the factors of 18 are 1, 2, 3, 6, 9, and 18 (each will divide into the base number such that there is no remainder) while the factors of 40 are 1, 2, 4, 5, 8, 10, 20 and 40. Since the Highest Common Factor or HCF is 2 for both numbers and not 1, they are not co-prime. **I think we are getting this coprime thing...**

How to find a modular inverse?

An *unsophisticated simple method (I like simple...)* of finding a modular inverse for $A \pmod{C}$ is:

- Step 1. Calculate $A * B \pmod{C}$, for B values, 0 through C-1
- Step 2. The modular inverse of $A \pmod{C}$ is the B value that makes $A * B \pmod{C} = 1$

Note that the term $B \pmod{C}$ can only have an integer value 0 through C-1, so testing larger values for B is redundant.

An example: $A=3, C=7, B = 6,5,4,3,2,1,0$

- Step 1. Calculate $A * B \pmod{C}$ for B values 0 through C-1
 - $3 * 0 \equiv 0 \pmod{7}$ (0/7, no remainder)
 - $3 * 1 \equiv 3 \pmod{7}$ (remainder of 3)
 - $3 * 2 \equiv 6 \pmod{7}$ (remainder of 6)
 - $3 * 3 \equiv 9 \pmod{7}$ (remainder of 2)
 - $3 * 4 \equiv 12 \pmod{7}$ (remainder of 5)
 - $3 * 5 \equiv 15 \pmod{7} \equiv (\text{remainder of } 1 \text{ or } 1 \pmod{7})$ <----- FOUND INVERSE!
 - $3 * 6 \equiv 18 \pmod{7} \equiv (\text{remainder of } 4)$
- Step 2. The modular inverse of $A \pmod{C}$ is the B value that makes $A * B \pmod{C} = 1$
Thus 5 is the modular inverse of $3 \pmod{7}$ since $5*3 \pmod{7} = 1$ ($15/7 = 2$ equal times remainder of 1)

The **modular inverse** is normally found with the Extended Euclidean Algorithm (not discussed here as outside the scope of necessary understanding for Blockchain, and besides I'd have to spend a lot of time getting up to speed on such), but we will take another simple short cut (yeah!, simple and a shortcut) and just cheat and use an online modular inverse calculator instead (<https://planetcalc.com/3311/>). Plugging our information into the formula:

$$d = 1/11 \pmod{349,716}$$

To perform this operation, simply input 11 (the e value) where it says **Integer in the online calculator** and 349,716 (or $\lambda(n)$ value) where it says **Modulo** in the online calculator. The result is:

Inverse Modulo Calculator

Integer 11	Modulo 349716
---------------	------------------

CALCULATE

Modular Multiplicative Inverse
254339

$$d = 254,339$$

This result indicates the co-prime numbers are $(254,339 \cdot 11)$ or 2,797,729 and 349,716, since $(254339 \cdot 11) / 349716 = 8$ even times $(2,797,728)$ with remainder of 1, or the Highest Common Factor for the two co-prime numbers is 1.

Now that we have the value for d , we can decrypt messages that were encrypted with our public key, using the following formula:

$$m = c^d \text{ mod } n \text{ (where } c^d \text{ means raise } c \text{ to the } d \text{ power, such as } 2^3 = 2 \times 2 \times 2 = 8)$$

We can now go back to the ciphertext that we encrypted under the **Generating the public key** section. When we encrypted the message with the public key, it gave us a value for c of **688,749**. From above, we know that d equals **254,339**. We also know that n equals **701,111**. This gives us:

$$m = 688,749^{254,339} \text{ mod } 701,111.$$

As you may have noticed, trying to take a number to the 254,339th power might be a little bit much for most normal calculators. Instead, we will be using an online RSA decryption calculator (https://www.cs.drexel.edu/~jpopyack/Courses/CSP/Fa17/notes/10.1_Cryptography/RSA_Express_EncryptDecrypt_v2.html). If you wanted to use another method, you would apply the powers as you normally would and perform the modulus operation in the same way as we did in the **Generating the public key** section.

In the calculator linked above, enter 701,111 where it says **Supply Modulus: N**, 254,339 where it says **Decryption Key: D**, and 688,749 where it says **Ciphertext Message in numeric form**, as shown below:

RSA Express Encryption/Decryption Calculator

This worksheet is provided for message encryption/decryption with the RSA Public Key scheme. No provisions are made for high precision arithmetic, nor have the algorithms been encoded for efficiency when dealing with large numbers.

To use this worksheet, you must supply:

- a modulus N , and either:
 - a plaintext message M and encryption key e , OR
 - a ciphertext message C and decryption key d .

The values of N , e , and d must satisfy certain properties. See [RSA Calculator](#) for help in selecting appropriate values of N , e , and d .

J. Pospisil, December 2002. Revised December 2012

The largest integer your browser can represent exactly is 9007199254740991 .

To encrypt a message, enter valid modulus N below. Enter encryption key e and plaintext message M in the table on the left, then click the **Encrypt** button. The encrypted message appears in the lower box.

To decrypt a message, enter valid modulus N below. Enter decryption key d and encrypted message C in the table on the right, then click the **Decrypt** button. The decrypted message appears in the lower box

Supply Modulus: N <input type="text" value="701111"/>	
Supply Encryption Key and Plaintext message M:	Supply Decryption Key and Ciphertext message C:
Encryption Key: e <input type="text"/>	Decryption Key: d <input type="text" value="254339"/>
Plaintext Message to encode: <input type="text"/>	Ciphertext Message in numeric form: <input type="text" value="688749"/>
<input type="button" value="Encrypt"/>	<input type="button" value="Decrypt"/>
Plaintext Message in numeric form: <input type="text"/>	Decrypted Message in numeric form: <input type="text" value="4"/>
Encrypted Message in numeric form: <input type="text"/>	Decrypted Message in text form: <input type="text"/>

Once you have entered the data, hit **Decrypt**, which will put the numbers through the decryption formula that was listed above. This will give you the original message in the box below. The decrypted answer is...**4 the correct (and miraculous) result. We did it!**

Consequently, the generation of the public key produces certain numbers that are used to generate the private key, hence why those keys are described as a pair of linked keys. It has been proven (not by me) that this linkage is unique and no other key pairs exist.

How does RSA encryption work in practice?

In the steps listed above, we have shown how two entities can securely communicate without having previously shared a code beforehand. First, they each need to **set up their own key pairs** and **share the public key with one another**. The two entities need to keep their private keys secret in order for their communications to remain secure.

Once the sender has the public key of their recipient, they can use it to encrypt the data that they want to keep secure. **Once it has been encrypted with a public key, it can only be decrypted by the private**

key from the same key pair. Even the same public key can't be used to decrypt the data. This is due to the properties of **trap door functions** that we mentioned above.

When the recipient receives the encrypted message, they use their private key to access the data. If the recipient wants to return communications in a secure way, **they can then encrypt their message with the public key of the party they are communicating with.** Once it has been encrypted with the public key, the only way that the information can be accessed is through the matching private key.

In this way, RSA encryption can be used by previously unknown parties to securely send data between themselves. Significant parts of the communication channels that we use in our online lives were built up from this foundation.

How are more complicated messages encrypted with RSA?

In our example, we simplified things a lot to make it easier to understand, which is why we only encrypted a message of "4". Being able to encrypt the number 4 doesn't seem particularly useful, so you might be wondering **how you can encrypt a more complicated set of data**, such as a symmetric key (which is the most common use of RSA), or even a message.

Some people may be perplexed at how a key like "n38cb29fkbjh138g7fqijnf3kaj84f8b9f..." or a message like "buy me a sandwich" can be encrypted by an algorithm like RSA, which deals with numbers and not letters. The reality is that all of the information that our computers process is stored in binary code or bits (1s and 0s) and we use encoding standards like **ASCII (American Standard Code for Information Interchange or Unicode** (an information technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems to represent them in ways that humans can understand (letters representing numbers and numbers representing letters).

Encryption algorithms use complex calculations and procedures for converting plaintext data into unintelligible ciphertext. Plaintext data includes not only letters, but also numbers, tables, graphs, pictures and symbols. The plaintext encryption conversion process is linked to the digital encryption keys, needed for the algorithm to encrypt and decrypt the data messaging. The keys are actively involved in the algorithm encryption and decryption processes. Since the encryption process is primarily based on mathematical principles, the encryption process requires that the plaintext data be converted to a numerical type code so that the encryption mathematical operations can function. Computers only understand binary code and think in bits of data or 1s and 0s. Consequently plaintext needs to be converted to a mathematical language and the encrypting/decrypting algorithm, along with the keys, convert that mathematical language to unintelligible cipher text which is then converted to binary code, 0s and 1s, the computer understands.

The plaintext to number or math speak conversion process is complex as well as being performed by many different processes. Below is a discussion in simple terms of the principles involved behind converting plaintext letters to numbers and then that encrypted to ciphertext to computer speak. Whatever conversion system is used, reference is made to tables of conversion information (letters to numbers and numbers to letters).

To illustrate the conversion process of letters to numbers, let's assume a simple plaintext data message: *" Call me at 6"*.

Computer compilers convert plaintext into binary code, and that binary code can be converted to other code such as American Standard Code for Information Interchange, decimal or hexadecimal code. Below are example charts illustrating such standard codes.

ASCII - Binary Character Table

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	?	0011 1011
I	0100 1001	g	0110 0111	!	0011 1111
J	0100 1010	h	0110 1000	i	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(0010 1000
N	0100 1110	l	0110 1100)	0010 1001

Standard ASCII Chart / ASCII Table - Hex to Decimal Code Conversion

Dec	Hex	Oct	Bin
0	0	000	0000000000
1	1	001	0000000001
2	2	002	0000000010
3	3	003	0000000011
4	4	004	0000000100
5	5	005	0000000101
6	6	006	0000000110
7	7	007	0000000111
8	8	010	0000010000
9	9	011	0000010001
10	A	012	0000010010
11	B	013	0000010011
12	C	014	0000011000
13	D	015	0000011001
14	E	016	0000011010
15	F	017	0000011011

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SCH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	>	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-~	63	3F	?	95	5F	~	127	7F	DEL

Thus “Call me at 6” or “Call[space]me[space]at[space]6” [white space and brackets are inserted by the author to separate letters and show spaces, but such whitespace /brackets are not part of the conversion code] translated into binary, decimal and hexadecimal code is illustrated below.

In binary code:

```

C    a    l    l    [space]    m    e    [space]
01000011 01100001 01101100 01101100[0010 00002]01101101 01100101[0010 00002]
a    t    [space]    6
01100001 01110100[0010 00002]00000110
  
```

In decimal code:

```

C a l l [sp] m e [sp] at t [sp] 6
067 097 108 108[032]109 101[032]097 116[032]006
  
```

In hexadecimal code:

```

C a l l [sp] m e [sp] a t [sp] 6
43 61 6C 6C[20]6D 65[20]61 74[20]06
  
```

As a bit of a temporary saunter into encryption techniques, I found the following to be interesting and entertaining...

An interesting historical encryption process is the *Vigenere* cipher illustrated in the below chart, and is just one of many that could be illustrated.

cipher	VVVRBACP
key	COVERCOVER...
plaintext	THANKYOU

In encrypting plaintext, the cipher letter is found at the intersection of the column headed by the plaintext letter and the row indexed by the key letter. To decrypt ciphertext, the plaintext letter is found at the head of the column determined by the intersection of the diagonal containing the cipher letter and the row containing the key letter.

A very simple encryption is illustrated by the following symmetric (one key) encryption example:

Assume the symmetric key is the number “4”, which means to substitute for each letter of a plaintext message (in English), the fourth letter in the alphabet.

Plaintext: H E L L O

Key: 4 4 4 4 4

Ciphertext: L I P P S

The ciphertext is then converted to the relevant binary, decimal or hexadecimal code.

Of course, modern symmetric encryption algorithms are far more complex than this simple example, making use of sophisticated combinations of substitution (changing one letter for another) and transposition (rearranging the letters of a message). Some typical symmetrical key encryption algorithms include: The Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish, and Twofish.

The Vigenere cipher mentioned above demonstrates encryption processes are quite varied. The key is to have a strong key used in an encryption/decryption algorithm that is random in its make up and long to prevent attackers cracking the code or the key, and the key used to convert plaintext or numerical coded text into unintelligible ciphertext that is difficult (hard) to crack.

This means that **keys like “n38cb29fkbjh138g7fqijnf3kaj84f8b9f...” and messages like “buy me a sandwich” already exist as numbers when converted by one of the computer coding tables**, which can easily be computed in the RSA algorithm. The numbers that they are represented by being complex binary code characters, are much larger and difficult and clumsy for many systems to manage, which is why many prefer to deal with alphanumeric characters rather than a jumble of binary code.

If you wanted to **encrypt a longer session key or a more complex message with RSA, it would simply involve a much larger number.**

Padding

When RSA is implemented, it uses something called **padding to help prevent a number of (hacker) attacks**. To explain how this works, we'll start with an example. Let's say you were sending a coded message to a friend:

Dear Karen,

I hope you are well. Are we still having dinner tomorrow?

Yours sincerely,

James

Let's say that you coded the message in a simple way, by **changing each letter to the one that follows it in the alphabet**. This would change the message to:

Efbs Lbsfo,

J ipqf zpv bsf xfm m. Bsf xf tujmm ibwjoh ejoofs upnpssp x?

Zpvst tjodfsfmz,

Kbnft

If your enemies intercepted this letter, there is a trick that they could use to try and crack the code. They could **look at the format of your letter and try to guess what the message might be saying**. They know that people normally begin their letters with “Hi”, “Hello”, “Dear” or a number of other conventions.

If they tried to apply “Hi” or “Hello” as the first word, they would see that it wouldn't fit the number of characters. They could then try “Dear”. It fits, but that doesn't necessarily mean anything. The attackers would just try it and see where it led them. So they would change the letters “e”, “f”, “b”, and “s” with “d”, “e”, “a”, and “r” respectively. This would give them:

*Dear Laseo,
J ipqe zpv are xemm. Are xe tujmm iawjoh djooes upnpsspx?
Zpvt tjoderemz,
Kanet*

It still looks pretty confusing, so the attackers might try looking at some other conventions, **like how we conclude our letters**. People often add “From” or “Kind regards” at the end, but neither of these fit the format. Instead, the attackers might try “Yours sincerely” and replace the other letters to see where it gets them. By changing “z”, “p”, “v”, “t”, “j”, “o”, “d” and “m” with “y”, “o”, “u”, “s”, “i”, “n”, “c” and “l” respectively, they would get:

*Dear Lasen,
I ioqe you are xell. Are xe tuill iawinh dinnes uonossox?
Yours sincerely,
Kanet*

After that modification, it looks like the attackers are starting to get somewhere. They have found the words “l”, “you” and “are”, in addition to the words that made up their initial guesses.

Seeing as the words are in correct grammatical order, the attackers can be pretty confident that they are heading in the right direction. By now, they have probably also realized that the code involved each letter being changed to the one that follows it in the alphabet. **Once they realize this, it makes it easy to translate the rest and read the original message.**

The above example was just a simple code, but as you can see, **the structure of a message can give attackers clues about its content**. Sure, it was difficult to figure out the message from just its structure and it took some educated guesswork, but you need to keep in mind that computers are much better at doing this than we are. **This means that they can be used to figure out far more complex codes in a much shorter time**, based on clues that come from the structure and other elements.

If the structure can lead to a code being cracked and reveal the contents of a message, then we need some way to hide the structure in order to keep the message secure. This brings us to **padding**.

When a message is padded, **randomized data is added to hide the original formatting clues that could lead to an encrypted message being broken**. With RSA, things are a little bit more complicated, because an encrypted key doesn’t have the obvious formatting of a letter that helped to give us clues in our above example.

Despite this, adversaries can use a number of attacks to exploit the mathematical properties of a code and break encrypted data. Due to this threat, **implementations of RSA use padding schemes like OAEP (Optimal Asymmetric Encryption Padding) to embed extra data into the message**. Adding this padding before the message is encrypted makes RSA much more secure.

Signing messages

RSA can be used for more than just encrypting data. Its properties also make it a useful system for **confirming authenticity or that a message has been truthfully sent by the entity who claims to have sent it, as well as proving that an original message hasn’t been altered or tampered with**.

When someone wants to prove the authenticity of their message, they can compute a **hash** – see **Article 7** (a function that takes data of an arbitrary size and turns it into a fixed-length value, a fingerprint of the data) of the plaintext, then sign it with their private key. They **sign the hash by applying the same formula that is used in decryption** ($m = cd \bmod n$). Once the message has been signed, they send this digital signature to the recipient in addition to the encrypted message.

If a recipient receives a message (either plaintext or ciphertext) with a digital signature, they can use the **signature to check whether the message was authentically signed by the private key of the person who claims to have sent it**. They can also see whether the message has been changed by attackers after it was sent.

To check the digital signature, the recipient first uses the same hash function to find the hash value of the message they received. The recipient then applies the **sender's public key** to the digital signature, **using the encryption formula** ($c = me \bmod n$), to give them the hash of the digital signature.

By **comparing the hash of the message that was received alongside the hash from the encrypted digital signature**, the recipient can tell whether the message is authentic. **If the two values are the same, the message has not been changed since it was signed** by the original sender. If the message had been altered by even a single character, the hash value would be completely different and the data red flagged as being suspect.

RSA security & attacks

Like most cryptosystems, the security of RSA depends on how it is implemented and used. One important factor is the size of the key. The **larger the number of bits in a key (essentially how long the key is), the more difficult it is to crack through hacker attacks** such as brute-forcing and factoring.

Since asymmetric-key algorithms such as RSA can be broken by integer factorization (computers brute force guessing at what the factors of a key might be), while symmetric-key algorithms like AES cannot, RSA keys need to be much longer to achieve the same level of security.

Currently, the **largest key size that has been factored is 768 bits long**. This was done by a team of academics over a two year period, using hundreds of machines.

Since the factoring was completed by the end of 2009 and computing power has grown significantly since that time, it can be assumed that an attempt of similar intensity **could now factor a much larger RSA key**.

Despite this, the time and resources needed for this kind of attack puts it out of the reach of most hackers and into the realm of nation state hackers. The best key length to use will depend on your individual threat model. The National Institute of Standards and Technology **recommends a minimum key size of 2048-bit, but 4096-bit keys are also used** in some situations where the threat level is higher (national security applications).

Factoring is just one way that RSA can be broken. A number of other attacks have the potential to break the encryption with a smaller amount of resources, but these depend on the implementation and other factors, not necessarily RSA itself. Some of these include:

Are the primes really random?

Some implementations of RSA use weak random number generators to come up with the primes. If these numbers aren't sufficiently random, it makes it much easier for attackers to factor them and break the encryption. This problem **can be avoided by using a cryptographically secure pseudo-random number generator**.

Poor key generation

RSA keys need to fall within certain parameters in order for them to be secure. **If the primes p and q are too close together, the key can easily be discovered**. Likewise, the number d that makes up part of the **private key cannot be too small**. A low value makes it easy to solve. It's important that these numbers are of adequate length to keep your key safe.

Side channel attacks

These are a type of attack that don't break RSA directly, but instead use information from its implementation to give attackers hints about the encryption process. These attacks can include things like **analyzing the amount of power that is being used**, or **branch prediction analysis**, which uses execution-time measurements to discover the private key.

Another type of side channel attack is known as a timing attack. If an attacker has the ability to measure the decryption time on their target's computer for a number of different encrypted messages, **this information can make it possible for the attacker to ascertain the target's private key**.

Most implementations of RSA avoid this attack by adding a one-off value during the encryption process, which removes this correlation. This process is called **cryptographic blinding**.

Is RSA encryption safe for the future?

The good news is that RSA is currently considered safe to use, despite these possible attacks. The caveat is that **it needs to be implemented correctly and use a key that falls within the correct parameters**. As we have just discussed, implementations that don't use padding, use inadequately sized primes or have other vulnerabilities cannot be considered safe.

If you want to use RSA encryption, **make sure that you are using a key of at least 1024 bits**. Those with higher threat models should stick to keys of 2048 or 4096 bits if they want to use RSA with confidence.

As long as you are conscious of the weaknesses that RSA has and use it correctly, you should feel safe to use RSA for key sharing and other similar tasks that require public key encryption.

While RSA is safe for now, the rise of quantum computing is expected to pose some challenges in the future.

Will quantum computing affect RSA?

The field of quantum computing continues to make steady improvements, but it will still be some years before it sees much use outside of a research context. While quantum computers have immense potential for advancing our capabilities, they will also bring some complications to the world of cryptography.

This is because quantum computers may be able to easily solve certain problems that are currently considered immensely difficult, and this difficulty is often what makes our cryptographic systems secure. In the case of symmetric-key algorithms like AES (Advanced Encryption Standard), powerful quantum

computers running Grover's algorithm would be able to significantly speed up attacks (**Grover's algorithm**, also known as the **quantum search algorithm**, refers to a quantum algorithm for unstructured search that finds with high probability the unique input to a black box function that produces a particular output value – in effect a cryptographic cracking technique).

While this certainly represents a threat against current cryptographic mechanisms, it is also relatively easy to fix. All we will have to do is double the key size to protect these symmetric-key algorithms.

When it comes to public-key cryptography like RSA, we are presented with a much greater problem. Once quantum computers become strong enough that they can effectively run Shor's algorithm (**Shor's algorithm** is a quantum computer algorithm for finding the prime factors of an integer, used for finding cryptographic keys), it may be feasible to solve the following three mathematical problems:

- The integer factorization problem
 - **integer factorization** is the decomposition of a composite number into a product of smaller integers (in effect finding the primes which can lead to successful hacker attacks). If these factors are further restricted to prime numbers, the process is called **prime factorization**.
- The discrete logarithm problem
 - The discrete logarithm problem is the computational task of finding an integer n with $g^n = t$.
- The elliptic-curve discrete logarithm problem

This is bad news, because the security of our most commonly used public-key algorithms relies on the premise that these are currently impractical to solve with current computational resources. In RSA's case, it's the integer factorization problem.

While quantum computing and Shor's algorithm are certainly a future threat to RSA, the good news is that we have time to change our cryptographic infrastructure to ensure our future security.

Although it's hard to know when exactly it will be feasible for quantum computers to break RSA, significant research and development are already underway. The US National Institute of Standards and Technology (NIST) is currently in the middle of soliciting and evaluating various public-key algorithms that will be secure in a post-quantum world.

RSA Cryptosystem

The RSA cryptosystem cryptography is the most employed cryptosystem today. The system was invented by three scholars Ron Rivest, Adi Shamir, and Len Adleman and hence, it is termed as RSA cryptosystem.

There are two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy. (**At least for some...**)

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- Encryption Function – It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of a private key.
- Key Generation – The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n . An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n . It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken.

The strength of RSA encryption drastically goes down (is not hard or entropy is low) against attacks if the number p_1 and p_2 are not large primes and/ or chosen public key e is a small number.

Many Cryptosystems

Along with RSA, there are many other public-key cryptosystems proposed. Many of them are based on different versions of the *Discrete Logarithm Problem*. The purpose of these systems are to generate public and private keys that are extremely difficult or hard to determine by a hacker. Characteristics of hardness include how random looking the generated number is, how long it is and how hard it is to duplicate the number even though the process or procedure of how it is generated is known. The 'hiding in plain sight' objective.

RSA Summary:

RSA Public and Private Key

- Producing ciphertext with RSA, the following equation is used:
 - $C = P^e \bmod n$, where C is the ciphertext and P is the plaintext
 - **Public Key = $[e \bmod n]$**
- Decrypting ciphertext with RSA the following equation is used:
 - $P = C^d \bmod n$
 - **Private Key = $[d \bmod n]$**

Key Generation	
Select p, q	p and q both prime
Calculate n	$n = p \times q$
Select integer d	$\text{gcd}(\phi(n), d) = 1; 1 < d < \phi(n)$
Calculate e	$e = d^{-1} \text{ mod } \phi(n)$
Public Key	$KU = \{e, n\}$
Private Key	$KR = \{d, n\}$
Encryption	
Plaintext: $M < n$	
Ciphertext: $C = M^e \text{ (mod } n)$	
Decryption	
Ciphertext: C	
Plaintext: $M = C^d \text{ (mod } n)$	

THUS ENDITH THE DISCUSSION ON RSA.

Elliptic Curve Cryptography (ECC)


Elliptic curve cryptography techniques are generally proprietary information owned by their Developers. This discussion outlines the broad generic principles used in such functions.

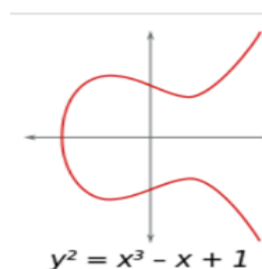
ECC applications, are a solution to the *discrete logarithm problem* (or *trapdoor function* – easy to fall through, tough to get out), a mathematical procedure that is easy to calculate in one direction (the cryptographic public key) but essentially impossible to reverse the process to discover a user's private and secret cryptographic key, used in determining the public key so they are related to each other.

Protecting the disclosure of the secret digital private key is a fundamental requirement of cryptography. For example, assume $3^{(\text{private key})} \bmod 17 = 12$ ($3^{(\text{private key})}$ means to raise the number to the power or exponent of the private key number), where the private key is known and = 29 thus $[3^{29} \bmod 17 = 12$, where 17 divided into 3^{29} , 4×10^{12} times with a remainder of 12] is easy to do,

But the reverse, where private key is unknown : $3^{\text{private key?}} \bmod 17 = 12$, is nearly impossible to know what the private key or exponent value could be. Later it will be shown the private key is the number of random and unpredictable times the elliptic cryptography curve calculation cycles to determine different whole number coordinate Points on the curve.



Elliptic curve cryptography (ECC) uses an algorithm that produces cryptographic public and private keys, that are used with other algorithms to perform critical security functions, including encryption (converting plaintext to unintelligible ciphertext), authentication (confirming the truthful identity of a person or entity) and digital signature (confirming a transaction is truthfully sent by a particular person or entity). ECC is based on the use of an elliptic (polynomial) mathematical function (curve), which generates cryptographic keys through the properties of the elliptic curve equation. An **Elliptic Function or Curve**, also known as the **Weierstrass equation** (which is not in the form of a traditional ellipse which is an oblong circle - ) is a polynomial equation of the form $y^2 = x^3 + Ax + B$ (with a condition that its **determinant**: $4A^3 + 27B^2 \neq 0 \bmod p$, not equal to zero mod p, which means the curve is 'non-singular' or has no repeated roots), where A and B are numerical constants. A graphical example of one of the more traditional continuous Elliptic Curve form examples used in cryptography is illustrated below:



The reason the typical curve is continuous is because it does not have any breaks in it. Some elliptical curves can have breaks (and not useful in cryptography applications). An example of an elliptic curve that is not continuous is:



ECC is an alternative cryptographic technique to determine public and private keys compared to the 'traditional' method of RSA factoring of very large prime numbers (the previously discussed n modulo p factoring process). RSA uses the prime factorization method, which involves taking two large prime random prime numbers and multiplying them to create a public key. Because of advancement in computing technology, RSA keys are slowly being cracked (or discovered by unauthorized persons), resulting in RSA key sizes getting much larger (where the larger number increases the security of maintaining the private key secret). There is a practical limit how large RSA keys can get before computer computational capacity is exceeded as well as unsustainable energy consumption (too much electricity) to power their processing, particularly for small hand held devices such as mobile phones, iPad and laptops which have limited computer and power (battery) capacities.

ECC is based on sets of two dimensional (think of a curve drawn on a flat piece of paper) elliptic curve "x" and "y" coordinate locations or 'Points' (on the curve), that are associated with mathematical objects (graphs) called elliptic curves. There are rules for adding and computing multiples of these Points, just as there are rules for RSA numbers (n) modulo p .

It is believed that the cryptographic secure **discrete logarithm problem (easy to calculate one way but hard the other, based on whole number discrete exponents of numbers, covered in the RSA discussion)** is much harder (meaning a better secured cryptographic process and harder to crack – has high entropy) when compared to RSA and applied to Points on an elliptic curve from which cryptographic keys are generated. This prompts possibly switching from the cryptography RSA numbers (n) modulo p process, to the cryptography Points on an elliptic curve, as a preferred cryptographic key generation alternative. In addition, an equivalent RSA security level can be obtained with ECC much shorter keys (which means faster computer processing time, less energy consumption if elliptic curve-based keys are used, especially on small devices such as laptops, mobile phone and iPad).

Shorter ECC key results in two benefits –

- Ease of key management
- Efficient computation

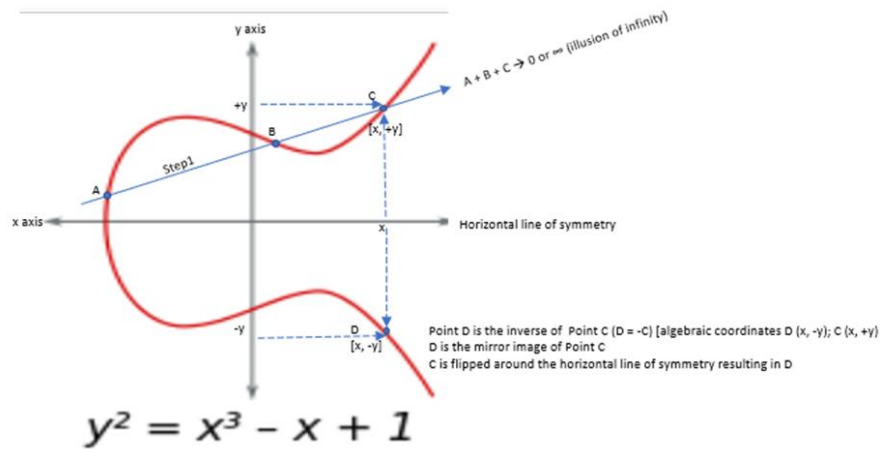
These benefits make elliptic-curve-based key generation schemes highly attractive for applications where computing resources are constrained (limited computer power and energy/battery life in small devices, such as tablet and mobile phones).

Elliptic curves have many interesting mathematical properties that make them well suited for cryptography.

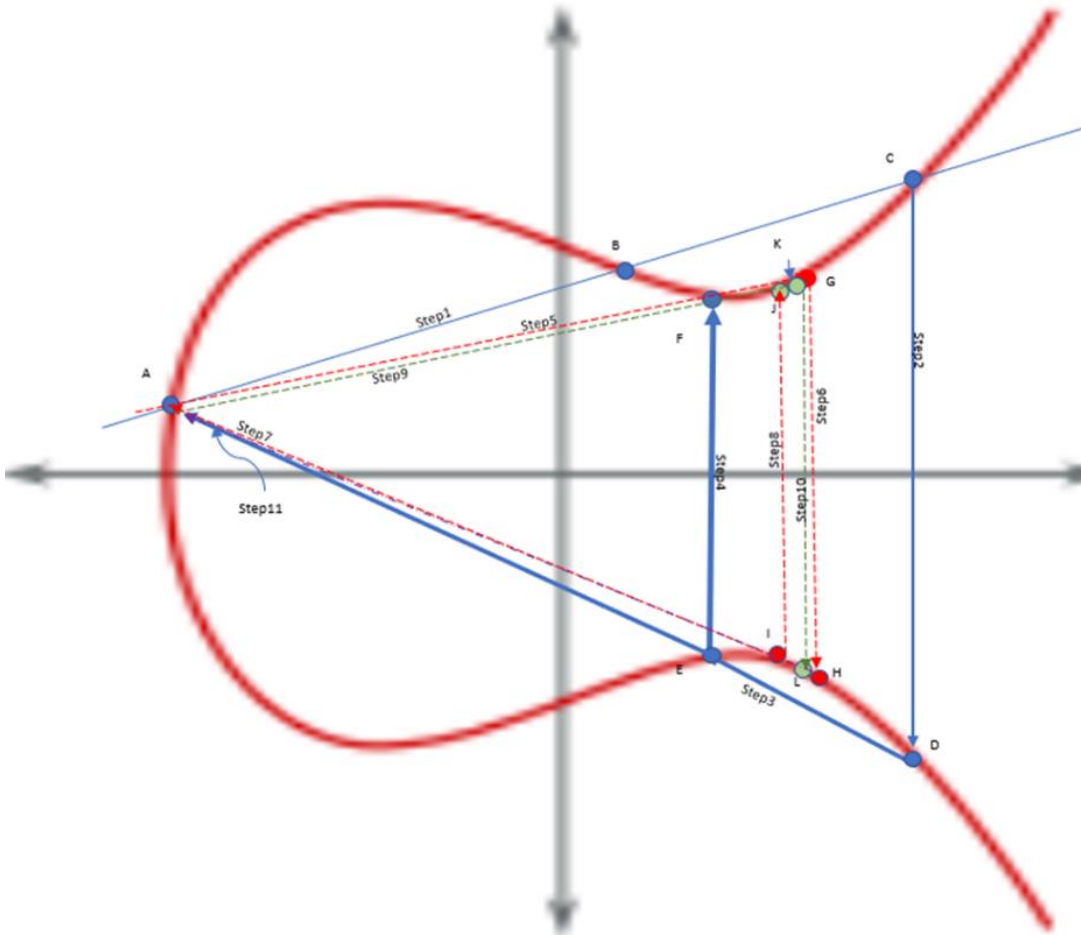
For example, referring to the below typical cryptographic elliptic curve graph, given two whole prime number (a number only divisible by 1 or itself) Points (x and y coordinates of all Points are whole prime numbers) on the curve, $A (x_A, y_A)$ and $B (x_B, y_B)$, and drawing a straight line to join them (Step1), there is automatically a third whole prime number Point $C (x_C, y_C)$ that lies on the curve, such that $A + B = C$. This

elliptic curve property is called “Point Addition” or “Point Add”. This property is not ordinary addition ($1 + 1 = 2$) but illustrates the **symbolic** addition property that Points A and B are associated with Point C on the straight line. This most important observation for cryptographic purposes is an operation that mathematically satisfies a set of criteria that applies to a **group** of data, and is a **symbolic** math ‘addition’ or grouping (or association) of Points on the curve. For a straight line that intersects three Points on the curve, A, B and C (a ‘**group**’ of Points), and represented as a group of points as, $A + B + C = 0$. This **grouping** does not represent an arithmetic addition problem (A is not added to B added to C and the addition result equals 0, zero), but represents the following ‘Point Addition’ concepts and symbolism:

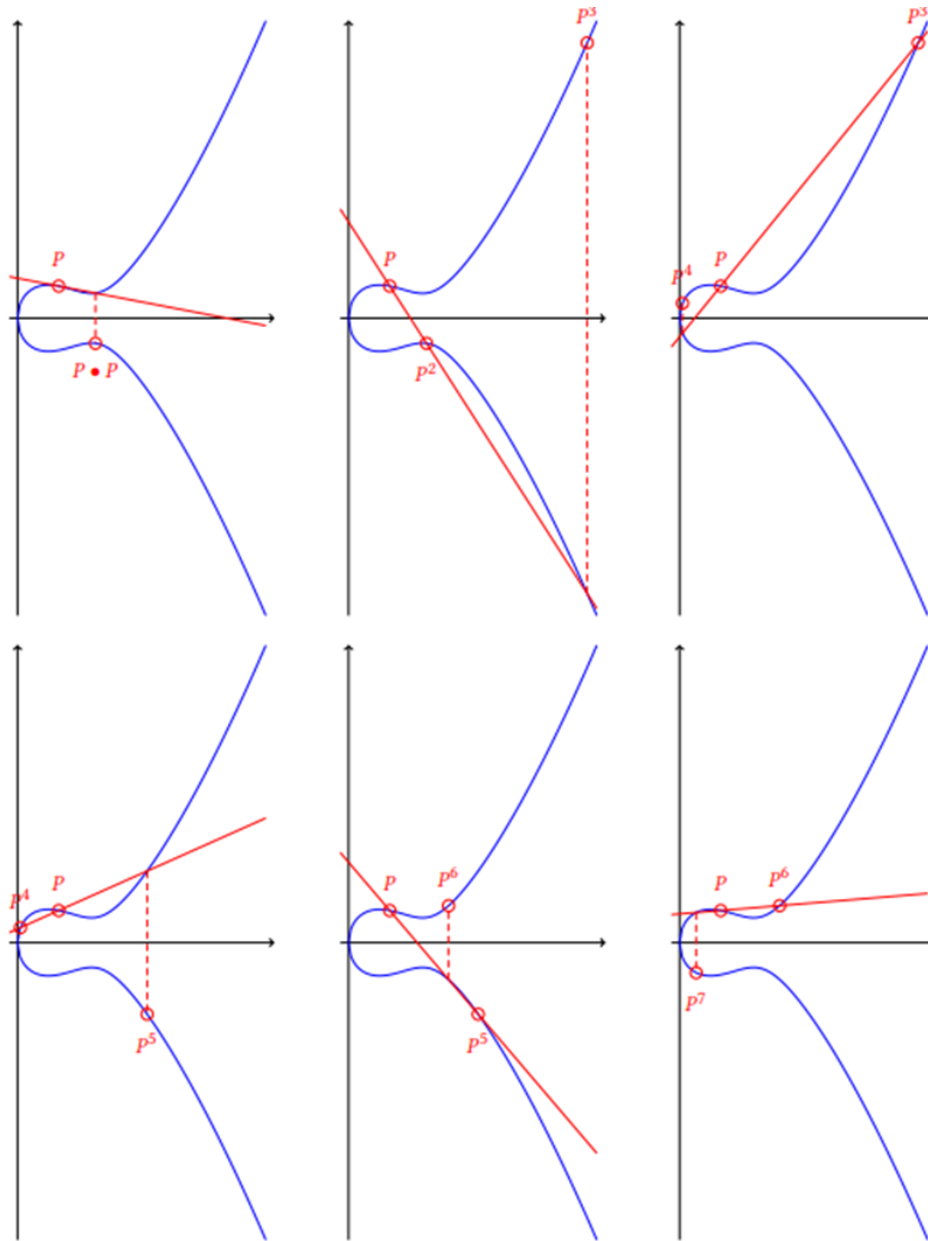
Since the elliptic curve is symmetrical about the x axis, if we take the inverse or negative y coordinate value of Point C ($-y_c$) and maintaining its x_c coordinate position, Point D ($x_c, -y_c$) is determined (Step2) by drawing a vertical straight line down from C to D.



If we next draw a straight line between Point D and Point A (Step3), see below illustration, that line will pass through Point E on the curve (another “Point Add” process). Taking the inverse or negative value of the y coordinate of Point E (same as drawing an upward vertical straight line from E to the curve, Step4), Point F is determined on the curve. Repeating the process of connecting Point F with Point A with a straight line (Step5), Point G is determined on the curve, and so, a recursive of repeatable process.



To further illustrate the Point Add (also written as: $P \cdot P$ or $P \cdot P$) elliptic curve method, repeatedly combining a curve Point P with itself, results in the below illustrative graphs. Each application of the $P \cdot P$ operator (combining the Points) can be visualized by drawing a straight line through P and P , finding the third point where this line intersects the curve, and mirroring that point with respect to the x axis to obtain P^{i+1} . The result is a sequence of Points that jump around in a pseudo random manner, which is an attractive property for cryptography.

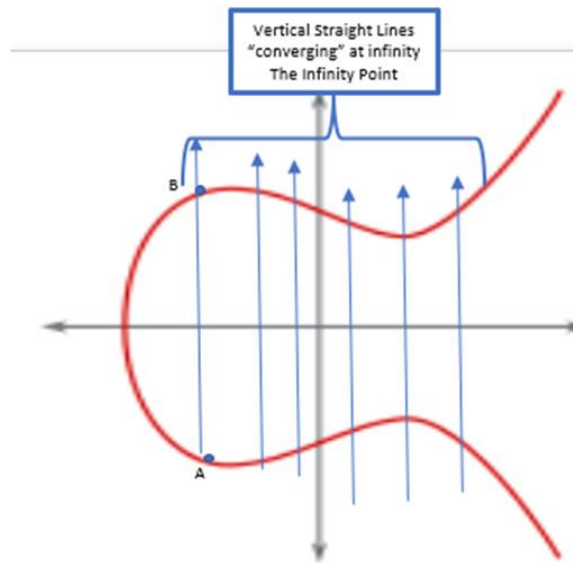


Source: <https://martin.kleppmann.com/papers/curve25519.pdf>

If the Points plotted on a elliptic curve are real numbers (any positive numbers including decimals not including zero) there are an infinite number of Points to plot. But when the coordinates are plotted as prime number **integers (negative or positive whole number including 0) modulo p**, [integer mod p] based on modular arithmetic, there is a finite or fixed number of Points that can be plotted, which is useful in cryptography. Normally, when straight lines are used to connect Points on the plot of **integer mod p**, the line will intersect a third Point, as illustrated above.

However, there are special cases when the two Points connected with a straight line, have the same x coordinate, and their mirror image reverse y coordinates are connected in a straight vertical line, oriented 90 degrees to the x axis. Those straight vertical lines have an infinite slope ($dy/dx = dy/0$ or slope = ∞)

and those lines being vertical go out to infinity (see the below diagram). By definition, the visualized infinite vertical lines are defined as **points of infinity** (a special phantom 'Point' or identity element that does not have any coordinates) to be the 'third *phantom* point' at which the lines "conceptually intersects the elliptic curve" at such third phantom Point. This phantom, coordinate less 'third infinity Point' can be visualized as lying infinitely far up the y axis and all vertical lines associated with the elliptic curve are conceptualized to intersect or converge at that phantom point.



A way to think about this *point of infinity* is to think about parallel railroad tracks that appear to intersect at the horizon (when in fact they do not but appears to 'converge' at infinity) .



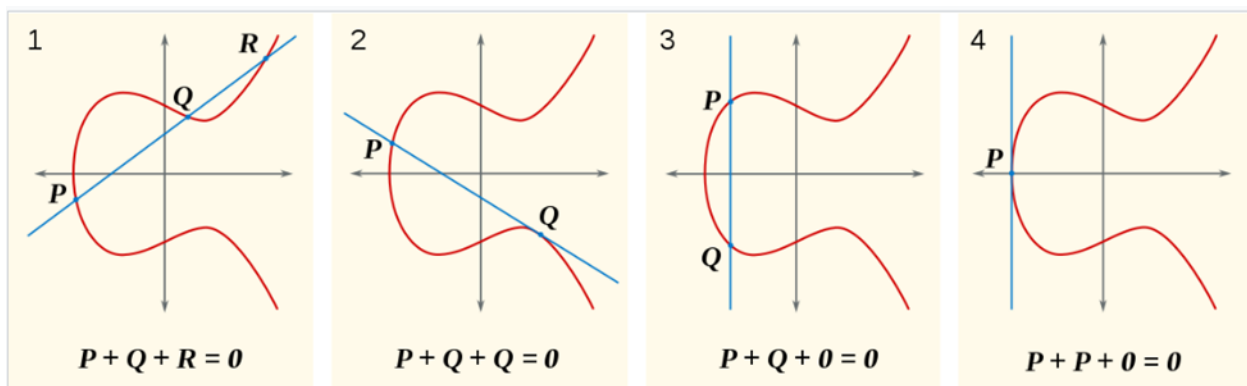
We define Point *inverses* as the Point "D" [$x_D = x_C, y_D = -y_C$] where Point C is flipped over the horizontal x axis line of symmetry (same x coordinate, but inverse y coordinates). From a math perspective, Points $P + Q = Q + P$ (which means 'Point Add' of the Points in either direction results in the same result which is called *Commutative* – the order of the math does not matter: example, $1 + 2 = 3$ and $2 + 1 = 3$). Also, grouping and Point Adding the Points in different order: $P + (Q + R)$ [which means Point Add Q to R first then Point Add that sum to P] is the same as $(P + Q) + R$, and this property means the operation is *Associative* (example: $(3 + (2 + 1)) = 3 + 3 = 6$, is the same as $((3 + 2) + 1 = 5 + 1 = 6)$. These commutative and associative properties of elliptic curves is important when used with cryptography because the calculated Points on the curve will be unique no matter the order or grouping of Points when determining encryption keys.

'Multiplying' a Point on the elliptic curve (called 'Point Doubling') by a number "N" will produce another Point on the curve, but it is very difficult to find what number "N" was used, even if you know the original and final Points.

There are three common operations for elliptic cryptographic curves used to generate public and private keys, though a 4th operation is added that is associated with the **Point of Infinity**:

- **Point Add:** $P + P$ (or $P_1 + P_2$) or $P + Q = R$ (Facet 1 in the below illustration)
- **Point Double:** $2P$ (Facets 2 and 4)
- **Negation** (Facet 3)
- **Point of Infinity**

The three operations, Point Add, Point Double and Negation, are discussed below and illustrated in the below diagrams.



As a reminder, Point Add, Point Negation and Point Double operations used in elliptic curve analyses, are **NOT** algebraic operations (that is, does not mean: $A + A = 2A$ or $2 * A = 2A$, algebraic results), but are **symbolic** operations. Regarding elliptic curves, **symmetric** about the x axis, **group operations** "•" ["dot"] refers to:

- **symmetric** means the elliptic curve lying above the x axis is a mirror image of the curve lying below the x axis and has the same form, just reversed;
- **group** means a collection or grouping of data (or x, y coordinate Points on the elliptic curve) on which common mathematical, operations or rules are applied;
- **operations** means the application of Point Add, Point Double or Point Negation symbolic operations on the curves Points; and
- **"•" or "dot" operation**, means a short hand way of writing non-arithmetic symbolic operations or procedures, performed on a group of data, which may or may not be associated with algebraic or arithmetical techniques, such as add, subtract, divide, multiply. For example, two separate data points, A and B (also individually called **elements** of the group of data) having different (x,y) coordinate locations ($A(x_A, y_A)$ and $B(x_B, y_B)$, where $x_A \neq x_B$ and $y_A \neq y_B$), lying on an elliptic curve, are said to be **Point Add** or $A + B$, which means symbolically, drawing a straight line through the two data Points and extending it to the curve, results in the straight line intersecting a third Point on the elliptic curve. That is a characteristic of elliptic curves, a straight line drawn anywhere through or on the curve will intersect at least 1 Point on the curve, sometimes just 2 but at most

3 (diagrams of these features are illustrated above). The Point Add or ‘addition’ of Points A and B does not mean adding their x and y coordinates together, but means the resultant associated third Point that lies on the elliptic curve that results when Points A and B are connected with a straight line.

Point at infinity

Point at infinity “ ∞ ” also written as “0” is as previously discussed, the identity element of elliptic curve arithmetic. (Recall it is a phantom ghost ‘Point’, does not have x or y coordinates, yet a conceptually recognized ‘Point’ in the distant infinity) and is the conceptual ‘convergence’ illusion of parallel vertical straight lines passing only through either 1 or 2 real Points lying on the elliptic curve, and like the illusion of parallel railroad tracks converging in the distance, appears to converge to a single third Point at infinity). Point Adding the Point of Infinity, branded as “0”, to any Point, results in that other Point, including Point Adding the Point at Infinity to itself. That is: $0 \bullet 0 = 0$ or $0 \bullet P = P$. Point of Infinity is an abstract concept that for all practical purposes depending on its use, can have a ‘value’ of infinity (∞) or no ‘value’, hence “0” (‘zero’). [Yes, I know, not an easy concept to mentally visualize, but this engineer has tried to explain it in my plain English]. I think of Point of Infinity as a convenient excuse of explaining the abstract symbolic movement of data points associated with an elliptic curve when such explanation is needed to describe what is useful about elliptic curves with cryptographic key generation – how to stop eavesdroppers cracking cryptographic codes.

Point negation (Facet 3)

Point negation is finding a Point on the elliptic curve, that adding it to itself will result in Point at Infinity (0), $P + (-P) = 0$. For elliptic curves of the form $y^2 = x^3 + ax + b$, negation are two points on the curve with the same x coordinate (mirror image around the symmetrical x axis) but negated (mirror image one +y the other -y) y coordinates: $(x, y_u) + (-x, y_d) = 0$; $(x, y_u) + (x, -y_d) = 0$; $(x_u, -y_u) = -(x, y_d)$. **In plain English**, Point negations refers to vertical lines passing through the elliptical curve only at the same x coordinate, but mirror image y coordinates (x, y_u) and $(x, -y_d)$, or y_u is the negative value of $-y_d$, thus ‘negated’, and because the lines are vertical they do not intersect a real third Point on the elliptic curve. However, the vertical lines go out to infinity and conceptually converge at the phantom ghost ‘third’ Point on the curve at infinity referred to as the infinity point or point at infinity.

Point addition (Facet 1)

Point Addition is a process where two different points on the curve are added (x and y coordinates). With 2 distinct points on the elliptic curve graph, P and Q, point addition is defined as the negation of the point resulting from the intersection of the elliptic curve, and the straight line defined by the points P and Q, giving the point, R. $P + Q = R$; $(x_p, y_p) + (x_q, y_q) = (x_r, y_r)$. Assuming the elliptic curve, is given by $y^2 = x^3 + ax + b$, the coordinates and location of the third Point R (x_r, y_r) can be calculated from: Slope of the straight line connecting P, Q and R, $\lambda_{slope} = (y_q - y_p)/(x_q - x_p)$; and the R coordinates as: $x_r = \lambda^2 - p - x_q$; and $y_r = \lambda(x_p - x_r) - y_p$. The y coordinate of the R point can be negated to establish a new Point on the curve with the

same R x coordinate value but -y value for the y coordinate (a vertical line drawn from R value to the mirror image value of R on the curve on the other side of the symmetrical x axis).

These dot operation equations are correct when neither Point is the point at infinity, "0", (which means points P and Q do not lie on a vertical line through the curve and the x coordinates of the points are different (in math speak: they're not mutual inverses having the same x coordinate but different y coordinates, + and – mirror image). **In plain English**, placing a straight line through Points P and Q will intersect the elliptic curve at a third Point R. A vertical (or negation point) can be drawn from Point R to its mirror image inverse location and intersect the elliptic curve at $(x_R, -y_R)$ – which means just draw a straight vertical line from point R until it intersects the elliptic curve at a point directly on the other side of the x axis. A straight line can then be drawn from the new inverse Point R back to Point P (the first, initial, seed or generation point – point P that started the cyclic Point Add process) and intersect the elliptic curve again between the new point R and P, and a vertical line drawn from that point to the elliptical curve for another negation point. This cycle process can continue indefinitely randomly creating new intersection Points on the curve, the x coordinate value of which, can be used in defining random cryptographic public and private keys. (Think of an elliptic curve as being an efficient pseudorandom number generator used to generate key values useful in cryptography).

Point doubling (Facets 2 and 4)

Point doubling means that the same point on the curve is added to itself. Where the points P and Q are coincident (at the same coordinates, either a vertical tangent line on the elliptic curve – Facet 4, or tangent straight line on the curve passing through a tangent Point on the curve and Point P or the seed or generation point, and the straight line not intersecting any other third Point on the elliptical curve), Point Add is similar, except that there is no well-defined straight line through P, so the operation is *closed using a limiting case*, the tangent to the curve, at P. In plain English, this means there is not a third Point generated by a straight line, and either a new third Point is arbitrarily defined as bouncing back and forth between Points P and Q, or if the slope of the line is not infinite (vertical) then Point Q assumed to be a new point R and vertical line drawn from that point to the elliptic curve to generate a new Q value, draw a straight through P (the generation point) and the new Q Point and assess for a new R Point value, and continue with the cyclic process to the desired number of cycles. Another way to say a random x coordinate number is being generated for use with cryptography.

If P and Q do not have a 0 (horizontal) or infinite (vertical) slope, taking the first derivative (which in calculus is the same as defining the slope of a line or tangent point) of the elliptic curve equation: $(dE/dx)/(dE/dy)$ (where "E" is just a shorthand naming for any elliptic curve equation E) which is the slope of the straight line connecting P and Q, $\text{slope} = \lambda = (3x_p^2 + a)/(2y_p)$, where "a" is from the defining equation of the elliptic curve.

Thus the Point Add and Point Double elliptic curve operations in effect are generating pseudorandom numbers from which to generate cryptographic keys.

Elliptic curve dot multiplication, similar as Point Add, is the operation of successively adding a Point along an elliptic curve to itself repeatedly. It is used in elliptic curve cryptography as a means of producing a

one-way function (random x coordinate value), or the discrete logarithmic problem. The security of elliptic curve cryptography depends on the inability of an attacker determining n from $Q = nP$ given known values of Q (new generated Points on the elliptic curve) and P (the seed or generation beginning process that started the cyclic operation on the curve to generate new Points) if n (the number of cyclic events) is large (known as the elliptic curve discrete logarithm problem by analogy to other cryptographic systems). The addition of two Points (Point Add) on an elliptic curve (or the addition of one Point to itself, $2P$ or Point Double) yields a third random Point on the elliptic curve whose location has no immediately obvious relationship to the locations of the first two, and repeating this many times over yields a Point nP that may be essentially anywhere (and unpredictable). The process in effect generates pseudorandom numbers that are a fundamental building block of secure cryptographic systems.

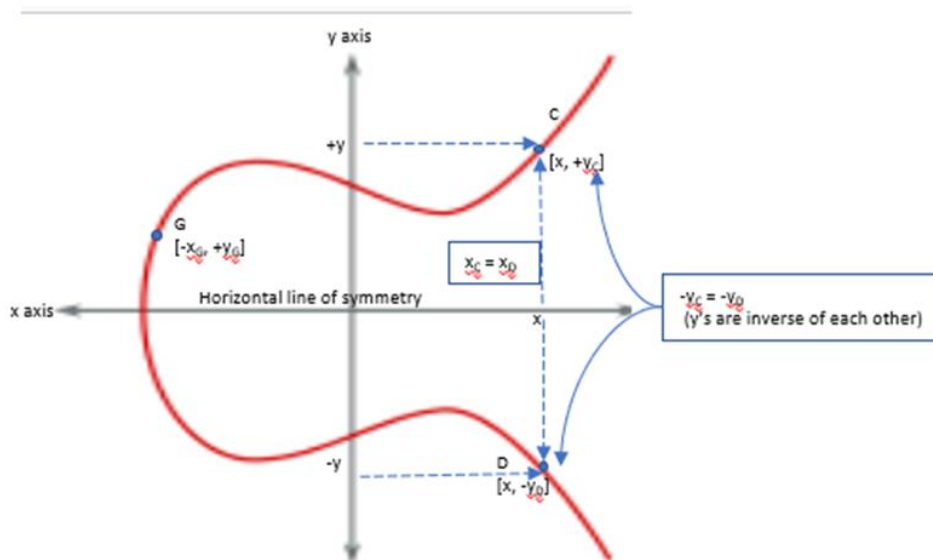
Intuitively, this cyclic process is not dissimilar to taking a Point P on a circle, adding 46.8 degrees to its angle may still be a Point "not too far" from Point P , but adding 1000 or 1001 times 46.8 degrees will yield a point that requires a bit more complex calculation to find the original angle. Reverting this process, i.e., given: (i) $Q = (n?)P$ and (ii) P , and trying to determine n (the number of cycles), can only be done by trying out all possible n values—an effort that is computationally impossible if n is large. Elliptic curves provide the recipe for advancing the discrete logarithmic problem and secure cryptography to send and receive secure messages or data over an insecure network, such as the internet.

Elliptic Cryptographic Curves – How Public and Private Keys Are Generated

Introduction

The following discussion illustrates the process and calculation principles used by ECC for generating the keys.

The general equation form of a two dimensional (meaning you view it in a two dimensional flat plain such as viewed on a sheet of paper or computer monitor screen) symmetrical elliptic curve is: $y^2 = x^3 + ax + b$, where a and b are constants and x and y are the coordinates of the curve's graph. When x and y coordinates are plotted on a graph, a typical elliptic continuous curve looks like:



A special Point, in this example “G” (the x_G and y_G coordinates of which are determined by a special calculation, illustrated later on), is placed on the curve, and such Point known as a “Generator” Point. The Generator Point is a special point on the elliptic curve that starts the beginning of the cyclical mathematical calculation procedure, used to ultimately generate public and private digital keys, where the random and unpredictable selected values of coordinates of the curve are used in determining the number value of the keys.

For cryptography purposes, the general ECC equation form is modified to, $y^2 = [x^3 + ax + b] * [\text{mod } p]$, a modular math form, which is used to determine **whole prime number discrete** x , y coordinate Points on the curve, easier numbers to deal with than decimals.

Recall [mod p] (or modulo in math speak) is the form of modular mathematics which means when the number determined by $[x^3 + ax + b]$ is divided by p (called the modulus), that division result determines the number of even times p is divided into the number and any remainder is the answer to the modulo calculation which is equal to y^2 . Since y is squared, the y value is the square root value (plus and minus) of the remainder answer.

Refresher examples of modulo math:

- assume $x^3 + ax + b = 9$ and $p = 8$, then $9 \text{ mod } 8$ means 8 divides into 9, 1 even time with a remainder of 1 ($9-8=1$), the answer, or $y^2 = 1$;
- $90 \text{ mod } 11 = 3$ (because 11 divides into 90, 8 even times being 88 and $90-88 = 3$, the remainder, $y^2 = 3$).

The **mod p** mathematical process inclusion with the ECC equation is a mathematical technique that will result in certain desirable cryptographic qualities:

- (1) The determination of x or y coordinates on the elliptic curve will be unique whole numbers (such as 7 or 11), no fractions (such as 1,010.1), and picking one of these whole prime numbers to be a G Generator Point that starts a cyclic process that produces random

unpredictable whole prime number Points. The random unpredictable number of cycles can be used as a **private key**.

- (2) **The public key** is determined by **NG** ($N \bullet G$), or Point Add G , N times. (If NG is large, it is practically impossible to determine N , the private key, by knowing NG)⁵¹.

Five essential conditions required to progress ECC public and private key generation calculations using the $n \bmod p$ modular arithmetic process:

- (1) **"a"** must be a constant whole positive number used in the selected ECC equation ($y^2 = x^3 + ax + b$);
- (2) **"b"** must be a constant whole positive number used in the selected ECC equation;
- (3) **"G"** the Generator Point at its x_G, y_G coordinate location on the elliptic curve (selected from the range of possible whole number Points on the elliptic curve, the calculation and determination of which is demonstrated below);
- (4) **"p"** an arbitrarily selected (large) positive prime number (divisible only by 1 or itself);
- (5) **"n"** the "quanta" or ("order") of the elliptic curve being the total number of whole prime number Points on the elliptic curve. As will be shown, although theoretically the number of x, y coordinate Points on the elliptic curve are infinite in number, the whole prime number modulo calculation process results in determining a finite (or fixed) number of whole prime number Points on the curve, a desirable cryptographic result. Recall a prime number is a unique whole number divisible only by 1 or itself. n is used to determine the private key, by obtaining a random number for the x or y coordinate (usually the y coordinate) selected from 1 to $(n-1)$ Points, and that number " N " Point Add or Point Doubling, G , the Generator Point is Added or Doubled (not in the mathematical sense but in the symbolic modular "dot" sense of cycling through the elliptic curve toward its infinity point. Each cycle generates a new Point on the curve, used in generating the public key, NG). Following the below example calculation will make some sense out of this convoluted description.

For illustrative purposes of showing how the elliptic curve cartography technique works to:

Find whole prime number points on the elliptic curve,

Determine a G Generator Point and

Generating example public and private keys,

the following elliptic curve example equation will be used: $y^2 = (x^3 + 7) \pmod{11}$;

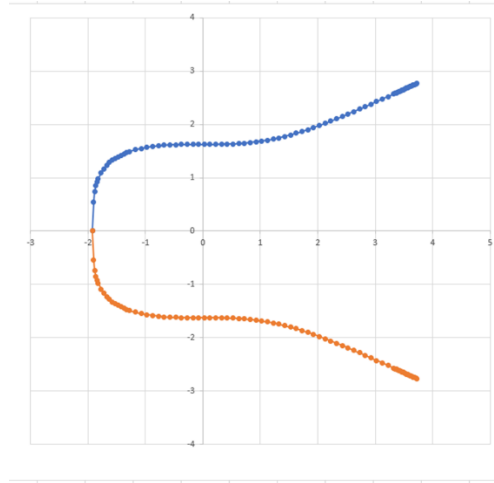
Parameters for this example of a standard curve, $y^2 = [x^3 + ax + b]$ are:

- (1) **"a"** = 0
- (2) **"b"** = 7
- (3) **"G"** Generator Point to be determined
- (4) **"p"** users choice and assumed to be the prime whole number 11 (the modulus)

⁵¹ "Montgomery's method", not shown here, is used to determine NG , an efficient calculation method to do the NG calculation.

- (5) "n" and "N" to be determined
 - (6) private key (randomly selected from n or N) to be determined
 - (7) public key (determined by Point Add and Point Doubling of N times G or N*G)
- A graph of the example elliptic curve $[y^2 = (x^3 + 7) \pmod{11}]$;

...



Step 1: Determining Whole Prime Number Coordinate Points On the Elliptic Curve (no fractions)

The maximum number of whole number coordinate Points, starting at $x = 0$, is determined by $p - 1$, or in this example, since $p = 11$, $p - 1 = 11 - 1 = 10$. Thus the whole number x coordinates to be assessed range from 0 thru 10.

Following are two methods for determining the whole number coordinates.

Method 1:

Step 1

$x = 0$ (whole number)

$y^2 = (0^3 + 7) \pmod{11}$; $7 \pmod{11} = 7$; $y^2 = 7?$; there is no y that will give a whole number answer (no modular answer), since the square root of 7 is not even, so $x = 0$ is not a whole number Point.

Step 2

$x = 1$ (whole number)

$y^2 = (1^3 + 7) \pmod{11}$; $8 \pmod{11} = 8$; $y^2 = 8?$; there is no y that will give a whole number answer (no modular answer), since the square root of 8 is not even, so $x = 1$ is not a whole number Point.

Step 3

$x = 2$; $y^2 = (2^3 + 7) \pmod{11}$; $15 \pmod{11} = 4 = y^2$ and $y = (+ \text{ or } - , \text{ use the positive answer}) +2$, a valid whole prime number (also known as a *quadratic residue*, just is, as I can't make this stuff up).

Another modular value (that gives the same remainder 2 value) is $y^2 = 4 = 4 \pmod{11}$ (11 does not divide into 4, leaving 4 as the remainder, same as prior result).

Since $y = 2$, $y^2 = 4$; $4 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 4$, is when $? = 81$ (since $81/11$: 7 times; $81 - 77 = 4$; thus $? = 81 = y^2$; $y = 9$.)

For $x = 2$, whole number data points on the elliptic curve are (2,2), (2,9)

Step 4

$x = 3$; $y^2 = (3^3 + 7) \pmod{11}$; $34 \pmod{11} = 1$; $y = 1$, a valid whole prime number (also known as a quadratic residue).

Another modular value is $y^2 = 1 = 1 \pmod{11}$ (11 does not divide into 1, leaving 1 as a remainder, same as prior result). Since $y = 1$, $y^2 = 1$; $1 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 1$, is when $? = 100$ (since $100/11$: 9 times; $100 - 99 = 1$; thus $? = 100 = y^2$; $y = 10$.)

For $x = 3$, whole number data points on the elliptic curve are (3,1), (3,10)

Step 5

$x = 4$; $y^2 = (4^3 + 7) \pmod{11}$; $71 \pmod{11} = 5$; $y^2 = 5$; y not a whole number Point.

Another modular value is $y^2 = 5 = 5 \pmod{11}$ (11 does not divide into 5, leaving 5 as a remainder, same as prior result). Since $y^2 = 5$; $5 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 5$, is when $? = 16$ (since $16/11$: 1 times; $16 - 11 = 5$; thus $? = 16 = y^2$; $y = 4$.)

Another modular value is $y^2 = 5 = 5 \pmod{11}$ (11 does not divide into 5, leaving 5 as a remainder, same as prior result). Since $y^2 = 5$; $5 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 5$, is when $? = 49$ (since $49/11$: 4 times; $49 - 44 = 5$; thus $? = 49 = y^2$; $y = 7$.)

Step 6

$x = 5$; $y^2 = (5^3 + 7) \pmod{11}$; $132 \pmod{11} = 0$; $y = 0$, a valid whole prime number (also known as a quadratic residue).

For $x = 5$, whole number data points on the elliptic curve are (5,0),

Step 7

$x = 6$; $y^2 = (6^3 + 7) \pmod{11}$; $223 \pmod{11} = 5$; $y^2 = 3$; y not a whole number Point.

Another modular value is $y^2 = 3 = 3 \pmod{11}$ (11 does not divide into 3, leaving 3 as a remainder, same as prior result). Since $y^2 = 3$; $3 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 3$, is when $? = 25$ (since $25/11$: 2 times; $25 - 22 = 3$; thus $? = 25 = y^2$; $y = 5$.)

Another modular value is $y^2 = 3 = 3 \pmod{11}$ (11 does not divide into 3, leaving 3 as a remainder, same as prior result). Since $y^2 = 3$; $3 \pmod{11}$ same as $y^2 \pmod{11}$; $y^2 = ? \pmod{11} = 3$ is when $? = 36$ (since $36/11$: 3 times; $36 - 33 = 3$; thus $? = 36 = y^2$; $y = 6$.)

Step 8

$x = 7; y^2 = (7^3 + 7) \pmod{11}; 350 \pmod{11} = 9; y = 3;$

Another modular value is $y^2 = 9 = 9 \pmod{11}$ (11 does not divide into 9, leaving 9 as a remainder, same as prior result). Since $y^2 = 9; 9 \pmod{11}$ same as $y^2 \pmod{11}; y^2 = ? \pmod{11} = 9$, is when $? = 64$ (since $64/11: 5$ times; $64 - 55 = 9$; thus $? = 64 = y^2; y = 8$.)

Another modular value is $y^2 = 3 = 3 \pmod{11}$ (11 does not divide into 3, leaving 3 as a remainder, same as prior result). Since $y^2 = 3; 3 \pmod{11}$ same as $y^2 \pmod{11}; y^2 = ? \pmod{11} = 3$ is when $? = 36$ (since $36/11: 3$ times; $36 - 33 = 3$; thus $? = 36 = y^2; y = 6$.)

Note that $x = 0$,

Repeating the above process for $x = 8$ thru 10 , and determining whole numbers for y (which do not have whole number coordinate points), results in the following **elliptic curve Point whole number pairings** (x, y)... (2,2) (2,9) (3,1) (3,10) (4,4) (4,7) (5,0) (6,5) (6,6) (7,3) (7,8)

The order, or n of when $p = 11$, is 11 , the total number of ordered pairs of whole number coordinates.

Method 2 (and for me the much easier of the two methods to use...)

Another technique for determining whole number Points is to compare $(x^3+7) \pmod{11}$ [equation mod p] results with $y^2 \pmod{11}$ [$y^2 \pmod{p}$] and where the remainders match, capture the pair of coordinates. The below table illustrates the calculation. When the modulo remainder result in column 2, matches the modulo remainder result in column 5, determines the whole number Point coordinate pairs.

1	2	3	4	5
x	$(x^3+7) \pmod{11}$		y	$y^2 \pmod{11}$
0	7		0	0
1	8		1	1
2	4		2	4
3	1		3	9
4	5		4	5
5	0		5	3
6	3		6	3
7	9		7	5
8	2		8	9
9	10		9	4
10	6		10	1

For example, for x whole number value of 6 (column 1, row 9), its modulo remainder is 3 (column 2, row 9), and that remainder value 3 matches in two places with the y modulo remainder (column 5), at $y = 5$ (column 4, row 8) and $y = 6$ (column 4, row 9). Thus, the whole number points for $x = 6$ are $(6,5)$ and $(6,6)$, same result as the more cumbersome calculation used in Method 1.

The x and y values ranges from 0 to 10 because $p - 1$ (where $p = 11$ prime number) equals 10 .

Mapping these results to their whole number pairs and noting their modulo remainder matches...

Whole Number Pair		Relationship	
x	y	x	y
2	2	2	0
2	9	3	1
3	1	4	2
3	10	5	3
4	4	6	4
4	7	7	5
5	0		6
6	5		7
6	6		8
7	3		9
7	8		10

Method 2 results in the following **elliptic curve Point whole number pairings** (x, y)... (2,2) (2,9) (3,1) (3,10) (4,4) (4,7) (5,0) (6,5) (6,6) (7,3) (7,8), same result as Method 1.

Another Method 2 example, assume p = 17:

x	$(x^3+7)\text{mod}17$		y	$y^2 \text{ mod}17$
0	7	no match	0	0
1	8		1	1
2	15		2	4
3	0		3	9
4	3	no match	4	16
5	13		5	8
6	2		6	2
7	10	no match	7	15
8	9		8	13
9	5	no match	9	13
10	4		10	15
11	12	no match	11	2
12	1		12	8
13	11	no match	13	16
14	14	no match	14	9
15	16		15	4
16	6	no match	16	1

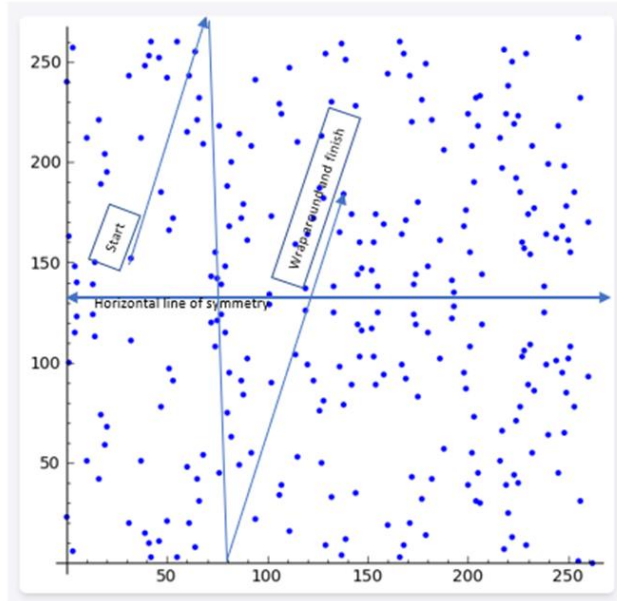
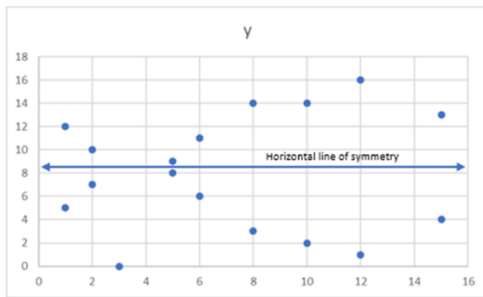
The whole number coordinate pairs are: (1,5), (1,12), (2,7), (2,10), (3,0), (5,8), (5,9), (6,6), (6,11), (8,3), (8,14), (10,2), (10,14), (12,1), (12,16), (15, 4), (15,13)

The following observations are made from these Method 1 and 2 calculations:

- All of the y values are unique and do not repeat.
- The maximum modulo remainder does not exceed p – 1.
- The maximum number of x’s and y’s to assess, ranges from 0 to p - 1
- The total number of whole number x and y whole number coordinates is p
- The sum of the y values for each x pairing, equals p, the chosen p prime number used in the modulo remainder analysis (except in the case where only one ordered x,y whole number

coordinate pair is determined, as in the above example at $x = 3, 7, 9, 11, 13, 14, 16$, there is only one set of x, y coordinates, there is only 1 y and not another to add to).

Only the whole number values determined from Method 1 or 2, can be plotted on a two dimensional (flat piece of paper) x, y coordinate graph. An illustration of such a plot is shown below (the left graph, $p = 17$ for the above example, the right another example with much much larger p), which maintains the symmetry of the whole number Points about the x axis, consistent with the elliptic curve symmetry.



Because decimals are a bit unruly, only take whole number integer Point coordinates and then take a modulus of the function (basically wrap the graph around the edges), so the graph ends up looking like this (note there's still a horizontal line of symmetry).

Note in the right illustration, connecting Points with a straight line, results in the movement of the connecting line 'wrapping around' the graph (the modular arithmetic remainder clock wrapping discussion).

A "G" Generator Point is (arbitrarily) selected from the whole number coordinate (x, y) pairs (and plotted on the elliptic curve graph).

Summary The discrete logarithm problem in ECC is based on the idea that, under certain conditions, all Points on an elliptic curve are whole numbers and form a cyclic group (a group that is formed by a single element, the Generator Point).

The private key (N) is determined by randomly choosing a prime whole number value from the range of n , between 0 and n , where n is the order of the elliptic curve used to generate the public key. The order of the curve or n is the maximum number of whole number pair Points on the elliptic curve. (In the prior example, $n = 17$).

The public key (fundamentally x and y coordinates on the curve) is determined by **NG** ($N \bullet G$), or Point Add G, N times, where NG is a random number multiple of the Generator Point. (If NG is large, it is difficult to determine N by knowing NG)⁵².

⁵² “Montgomery’s method”, not shown here, is used to determine NG, an efficient calculation method to do the NG calculation.

ARTICLE 11

WHAT IS THE INTERNET?

KEY TAKE AWAY: BLOCKCHAIN DOES NOT WORK WITHOUT THE INTERNET.

What is a network?

Before describing the Internet, let's first define what is a "network". A network is a group of connected computers that are able to communicate with each other by sending and receiving data to each other. A computer network is much like a social circle, which is a group of people who regularly exchange information, and coordinate activities together.

Usually all devices in the network are connected to a central hub — for instance, a router or server computer. A network can also include subnetworks, or smaller subdivisions of the network. Subnetworking is how very large networks, such as those provided by Internet Service Providers (Comcast, AT&T, etc), are able to manage thousands of IP addresses and connected devices.

What is the internet?

In contrast, the Internet is a vast, sprawling *collection* of networks that connect to each other. The word "Internet" can be derived from *interconnected networks*. Think of the Internet as a network of networks: computers are connected to each other within networks, and these networks connect to other networks. This enables these computers to connect with other computers both near and far.

Since computers connect to each other within networks and these networks also all connect with each other, one computer can talk to another computer in a faraway network thanks to the Internet. This makes it possible to rapidly exchange information between computers across the world.

Computers connect to each other and to the Internet via wires and cables (wired), radio waves (wireless), and other types of networking infrastructure (fiber optics). All data sent over the Internet is translated into analog signals, pulses of light or electricity, and then interpreted by a receiving computer into "bits" or binary code (0s and 1s) that computers understand. The wires, cables, and radio waves conduct this transmission at essentially the speed of light (that's why when you strike the send button to send an email message to the other side of spaceship earth, the receiver almost instantaneously receives the message on their computer). The more data that can pass over these wires and cables at once, the faster the Internet works.

What is distributed networking, and why is this concept important for the Internet?

There is no central control center for the Internet. Instead, it is a distributed networking system, meaning it is not dependent on any one individual machine and connected parties are distributed all over spaceship earth. Any computer or hardware that connects to the internet can send and receive data in the correct fashion (e.g. using the correct networking protocols set up by the Internet Service Provider and users of the Internet) and when connected becomes part of the Internet network.

The Internet's distributed nature (just like Blockchains) makes it resilient. Computers, servers, and other pieces of networking hardware connect and disconnect from the Internet all the time without impacting how the Internet functions — unlike an individual, centralized computer, which may not function at all if it is missing a component. This applies even at a large scale: if a server, an entire data center, or an entire

region of data centers goes down, the rest of the Internet can still function (even though possibly more slowly).

How does the Internet work?

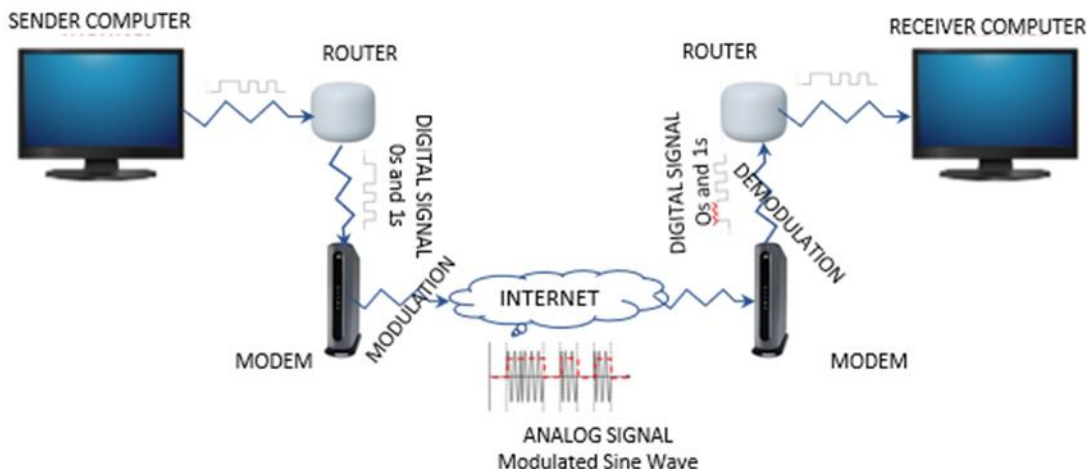
There are two main concepts that are fundamental to the way the Internet functions: **packets** and **protocols**.

Packets

In networking, a packet is a small segment of a larger message. Each packet contains both data and information about that data. The information about the packet's contents is known as the "header" attached to the front of the packet so that the sending and receiving machines know what to do with the packet – headers are thus administrative tools that direct internet traffic. To understand the purpose of a packet header, think of how some consumer products come with assembly instructions and packing slip.

When data gets sent over the Internet, it is first broken up into the smaller packets, which are then translated into bits (computer language binary code, 0s and 1s). The packets get routed to their destination by various networking devices such as routers and switches and the Internet Protocol address used to designate from where and to whom the packet is sent, just like a post office delivering a letter to a certain address. When the packets arrive at their destination, the receiving device reassembles the packets in order and can then use or display the data on a computer monitor that is in plaintext and understood by the human viewer.

The language of computers is binary code. Computers understand the world by converting data and information into a string of 0s or 1s, using binary code. One of these characters, either a 0 or a 1, is called a bit. 8 bits are called a byte. Electronic device performance is reported as how many bits (or bytes) per second the device can transmit data and information – how fast is it (like Amazon who prides itself on first delivering ordered goods in a few days, with Prime, can cut that down to the same day -- I am awaiting Amazon delivering something to me yesterday, as their ever efficient algorithms will know before I do that I want something). While bits of computer information are technically transmitted over the internet, such bits are not transmitted exactly as they are issued by computers. A modem takes the computer digital bits of data and information and encodes it (modulates it) on the internet analog radio sine wave signal, the language the internet understands. So in effect, bits are not truly transmitted over the internet but are piggybacked on to another signal.



Compare this process to some assembly required instructions of a product purchased from Amazon. The product components are first built elsewhere, and too large for Amazon to be shipped assembled, so it is shipped to the buyer in pieces, along with instructions about where each piece belonged. The buyer who received the pieces reassembled them into the finished product. Whalla...the internet and packet assembly example.

Sending digital information in smaller pieces or packets is extremely fast over the Internet.

Packets are sent across the Internet using a technique called packet switching. Intermediary routers and modems and switches are able to process packets independently from each other, without accounting for their source or destination. This is by design so that no single connection dominates the network. If data was sent between computers all at once with no packet switching, a connection between two computers could occupy multiple cables, routers, and switches for minutes at a time. Essentially, only two people would be able to use the Internet at a time — instead of an almost unlimited number of people, as is the case in reality.

Protocols

Connecting two computers, both of which may use different hardware and run different software, is one of the main challenges that the creators of the Internet had to solve. It requires the use of communications techniques that are understandable by all connected computers, just as two people who grew up in different parts of the world may need to speak a common language to understand each other.

This problem is solved with standardized protocols. In networking, a protocol is a standardized way of doing certain actions and formatting data so that two or more devices are able to communicate with and understand each other (an operating agreement).

There are protocols for sending packets

- between devices on the same network (Ethernet),
- for sending packets from network to network (Internet Protocol),
- for ensuring those packets successfully arrive in order (TCP - Transmission Control Protocol is a standard that defines how to establish and maintain a network conversation by which applications can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other over the internet in a secure fashion, and
- for formatting data for websites and applications (HTTPS – secure hypertext transfer protocol – web addresses).
- In addition to these foundational protocols, there are also protocols for routing, testing, and cryptographic encryption. And there are alternatives to the protocols for different types of content — for instance, streaming video often uses UDP (User Datagram Protocol) instead of TCP.

Because all Internet-connected computers and other devices can interpret and understand these protocols, the Internet works no matter who or what connects to it.

What physical infrastructure makes the Internet work?

A lot of different kinds of hardware and infrastructure go into making the Internet work for everyone. Some of the most important types include the following:

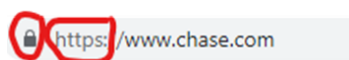
- Routers forward packets to different computer networks based on their destination. Routers are like the traffic cops of the Internet, making sure that Internet traffic goes to the right networks.
- Switches connect devices that share a single network. They use packet switching to forward packets to the correct devices. They also receive outbound packets from those devices and pass them along to the right destination.
- Web servers are specialized high-powered computers that store and serve content (webpages, images, videos) to users, in addition to hosting applications and databases. Servers also respond to DNS (Domain Name System, is the phonebook or unique names of users on the Internet, connecting web browsers with websites) queries and perform other important tasks to keep the Internet up and running. Most servers are kept in large data centers, which are located throughout the world.

How do these concepts relate to websites and applications users access over the Internet?

In order for you to see any article sent over the Internet, it was sent piece by piece in the form of several thousand data packets. These packets traveled over cables and radio waves and through routers, modems and switches from a web server to your computer or device. Your computer or smartphone received those packets and passed them to your device's browser, and your browser interpreted the data within the packets in order to display the text you are reading on the user's computer monitor (easier to read text from a monitor than to read and interpret binary code).

The specific steps involved in this process are:

1. **Domain Name System query:** When your browser loads a webpage, it likely first made a DNS query to find out it's website's Internet Protocol (IP) address.
2. **Transmission Control Protocol handshake:** Your browser opened a connection with that IP address.
3. **Transmission Layer Security handshake:** Your browser also set up encryption between the web server and your device so that attackers cannot read the data packets that travel between those two endpoints.
4. **HyperText Transfer Protocol request:** Your browser requested the content that appears on this webpage. (The safer protocol to use is the HyperText Transfer Protocol Secure or HTTPS).



5. **HTTP response:** Your Internet Server Provider server transmitted the content in the form of HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript code, broken up into a series of data packets. [These are different types of computer coding languages, just like some folk communicate in English others in Spanish]. Once your device received the packets and verified it had received all of them, your browser interpreted the HTML, CSS, and JavaScript code contained in the packets to display the article. The whole process takes only a second or two.
- 6.

What is a network protocol?

In networking (such as the internet, or group of computers linked together and communicating with each other), a protocol is a set of rules for formatting and processing data (a management and operating agreement). Network protocols are like a common language for computers. The computers within a network may use vastly different software and hardware; however, the use of protocols enables them to communicate with each other regardless. (Sort of like each computer talking in its own language, but an agreement reached that when communicating between each other, Spanish will be used).

Standardized protocols are like a common language that computers can use, similar to how two people from different parts of the world may not understand each other's native languages, but they can communicate using a shared third language. If one computer uses the Internet Protocol (IP) and a second computer does as well, they will be able to communicate — just as the United Nations relies on its 6 official languages to communicate amongst representatives from all over the globe. But if one computer uses Internet Protocol (IP being a unique address or fingerprint assigned to each computer linked to the network) and the other does not know this protocol, they will be unable to communicate.

On the Internet, there are different protocols for different types of processes. Protocols are often discussed in terms of which Operating Systems Interconnection or OSI model layer they belong to.

What are the layers of the OSI model?

The Open Systems Interconnection (OSI) model is an abstract representation of how the Internet works. It contains 7 layers, with each layer representing a different category of networking functions.

Protocols make these networking functions possible.

- For instance, the Internet Protocol (IP) is responsible for routing data by indicating where data packets (a packet is a small segment of data; all data sent over a network is divided into packets) come from and what their destination is – where the IP is the destination address. IP makes network-to-network communications possible. Hence, IP is considered a network layer (layer 3) protocol.
- As another example, the Transmission Control Protocol (TCP) ensures that the transportation of packets of data (data is assembled and sent in clumps or packets of data for technical and efficiency reasons) across networks goes smoothly. Therefore, TCP is considered a transport layer (layer 4) protocol.

Which protocols run on the network layer?

As described above, IP is a network layer protocol responsible for routing. But it is not the only network layer protocol.

- **IPsec:** Internet Protocol Security (IPsec) sets up encrypted, authenticated IP connections over a virtual private network (VPN). Technically IPsec is not a protocol, but rather a collection of protocols that includes the Encapsulating Security Protocol (ESP), Authentication Header (AH), and Security Associations (SA).
- **ICMP:** The Internet Control Message Protocol (ICMP) reports errors and provides status updates. For example, if a router is unable to deliver a packet, it will send an ICMP message back to the packet's source.
- **IGMP:** The Internet Group Management Protocol (IGMP) sets up one-to-many network connections. IGMP helps set up multicasting, meaning multiple computers can receive data packets directed at one IP address.

What other protocols are used on the Internet?

Some of the most important protocols to know are:

- **TCP:** As described above, TCP is a transport layer protocol that ensures reliable data delivery. TCP is meant to be used with IP, and the two protocols are often referenced together as TCP/IP.
- **HTTP:** The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, the Internet that most users interact with. It is used for transferring data between devices. HTTP

belongs to the application layer (layer 7), because it puts data into a format that applications (e.g. a browser) can use directly, without further interpretation. The lower layers of the OSI model are handled by a computer's operating system, not applications.

- **HTTPS:** The problem with HTTP is that it is not encrypted — any attacker who intercepts an HTTP message can read it. HTTPS (HTTP Secure) corrects this by encrypting HTTP messages.
- **TLS/SSL:** Transport Layer Security (TLS) is the protocol HTTPS uses for encryption. TLS used to be called Secure Sockets Layer (SSL).
- **UDP:** The User Datagram Protocol (UDP) is a faster but less reliable alternative to TCP at the transport layer. It is often used in services like video streaming and gaming, where fast data delivery is paramount.

What is the network layer?

Network-to-network connections are what make the Internet possible. The "network layer" is the part of the Internet communications process where these connections occur, by sending packets of data back and forth between different networks. In the 7-layer OSI model, the network layer is layer 3. The Internet Protocol (IP) is one of the main protocols used at this layer, along with several other protocols for routing, testing, and encryption.

Suppose Bob and Alice are connected to the same local area network (LAN), and Bob wants to send Alice a message. Because Bob is on the same network as Alice, he could send it directly to her computer across the network. However, if Alice is instead on a different LAN several miles away, Bob's message will have to be addressed and sent to Alice's network before it can reach her computer, which is a network layer process.

What happens at the network layer?

Anything that has to do with inter-network connections takes place at the network layer. This includes setting up the routes for data packets to take, checking to see if a server in another network is up and running, and addressing and receiving IP packets from other networks. This last process is perhaps the most important, as the vast majority of Internet traffic is sent over IP.

What is a packet?

All data sent over the Internet is broken down into smaller chunks called "packets." When Bob sends Alice a message, for instance, his message is broken down into smaller pieces and then reassembled on Alice's computer. A packet has two parts: the header, which contains information about the packet itself (administration stuff), and the body, which is the actual data being sent.

At the network layer, networking software attaches a header to each packet when the packet is sent out over the Internet, and on the other end, networking software can use the header to understand how to handle the packet.

A header contains information about the content, source, and destination of each packet (somewhat like stamping an envelope with a destination and return address). For example, an IP header contains the destination IP address of each packet, the total size of the packet, an indication of whether or not the packet has been fragmented (broken up into still smaller pieces) in transit, and a count of how many networks the packet has traveled through.

What is the OSI model?

The Open Systems Interconnection (OSI) Model is a description of how the Internet works. It breaks down the functions involved in sending data over the Internet into seven layers. Each layer has some function that prepares the data to be sent over wires, cables, and radio waves as a series of bits.

The seven layers of the OSI model are:

- **7. Application layer:** Data generated by and usable by software applications. The main protocol used at this layer is HTTP.
- **6. Presentation layer:** Data is translated into a form the application can accept. Some authorities consider HTTPS encryption and decryption to take place at this layer.
- **5. Session layer:** Controls connections between computers (this can also be handled at layer 4 by the TCP protocol).
- **4. Transport layer:** Provides the means for transmitting data between the two connected parties, as well as controlling the quality of service. The main protocols used here are TCP and UDP.
- **3. Network layer:** Handles the routing and sending of data between different networks. The most important protocols at this layer are IP and ICMP.
- **2. Data link layer:** Handles communications between devices on the same network. If layer 3 is like the address on a piece of mail, then layer 2 is like indicating the office number or apartment number at that address. Ethernet is the protocol most used here.
- **1. Physical layer:** Packets are converted into electrical, radio, or optical pulses and transmitted as bits (the smallest possible units of information) over wires, radio waves, or cables.



It is important to keep in mind that the OSI model is an abstract conceptualization of the processes that make the Internet work, and interpreting and applying the model to the real-world Internet is sometimes a subjective exercise.

The OSI model is useful for helping people talk about networking equipment and protocols, determining which protocols are used by which software and hardware, and showing roughly how the Internet works. But it is not a rigid step-by-step definition of how Internet connections always function.

OSI model vs. TCP/IP model

The TCP/IP model is an alternative model of how the Internet works. It divides the processes involved into four layers instead of seven. Some would argue that the TCP/IP model better reflects the way the Internet functions today, but the OSI model is still widely referenced for understanding the Internet, and both models have their strengths and weaknesses.

In the TCP/IP model, the four layers are:

- o 4. **Application layer**: This corresponds, approximately, to layer 7 in the OSI model.
- o 3. **Transport layer**: Corresponds to layer 4 in the OSI model.
- o 2. **Internet layer**: Corresponds to layer 3 in the OSI model.
- o 1. **Network access layer**: Combines the processes of layers 1 and 2 in the OSI model.

But where are OSI layers 5 and 6 in the TCP/IP model? Some sources hold that the processes at OSI layers 5 and 6 either are no longer necessary in the modern Internet, or actually belong to layers 7 and 4 (represented by layers 4 and 3 in the TCP/IP model).

In other words, the network layer and the Internet layer are basically the same thing, but they come from different models of how the Internet works.

What is the Internet Protocol (IP)?

The Internet Protocol (IP) is a protocol, or set of rules, for routing and addressing packets of data so that they can travel across networks and arrive at the correct destination. Data traversing the Internet is divided into smaller pieces, called packets. IP information is attached to each packet, and this information helps routers to send packets to the right place. Every device or domain that connects to the Internet is assigned an IP address, and as packets are directed to the IP address attached to them, data arrives where it is needed.

Once the packets arrive at their destination, they are handled differently depending on which transport protocol is used in combination with IP. The most common transport protocols are TCP and UDP.

What is a network protocol?

In networking, a protocol is a standardized way of doing certain actions and formatting data so that two or more devices are able to communicate with and understand each other.

To understand why protocols are necessary, consider the process of mailing a letter. On the envelope, addresses are written in the following order: name, street address, city, state, and zip code. If an envelope is dropped into a mailbox with the zip code written first, followed by the street address, followed by the state, and so on, the post office won't deliver it. There is an agreed-upon protocol for writing addresses in order for the postal system to work. In the same way, all IP data packets must present certain information in a certain order, and all IP addresses follow a standardized format.

What is an IP address? How does IP addressing work?

An IP address is a unique identifier assigned to a device or domain that connects to the Internet. Each IP address is a series of characters, such as '192.168.1.1'. Via Domain Name System, DNS resolvers, which translate plaintext human-readable domain names into IP addresses, users are able to access websites without memorizing this complex series of characters. Each IP packet will contain both the IP address of the device or domain sending the packet and the IP address of the intended recipient, much like how both the destination address and the return address are included on a piece of mail.

How does IP routing work?

The Internet is made up of interconnected large networks that are each responsible for certain blocks of IP addresses; these large networks are known as **autonomous systems (AS)**. A variety of routing protocols, including, Border Gateway Protocol (BGP) is the routing protocol for the Internet. Much like the post office processing mail, BGP picks the most efficient routes for delivering Internet traffic. BGP helps route packets across ASes based on their destination IP addresses. Routers have routing tables that indicate which ASes the packets should travel through in order to reach the desired destination as quickly as possible. Packets travel from AS to AS until they reach one that claims responsibility for the targeted IP address. That AS then internally routes the packets to the destination.

ARTICLE 12

WHAT IS A ROUTER AND MODEM?

KEY TAKE AWAY: NEED A MODEM TO COMMUNICATE WITH THE INTERNET.

What is a router?

A router is an electronic device that wirelessly connects other electronic devices (such as connecting a personal computer to a printer or modem or a modem to a smart TV) or technically speaking, connecting two or more packet-switched networks or subnetworks. It serves two primary functions: (1) managing traffic or transmission of data and information between these networks by forwarding data organized in lumps or convenient data packets, to their intended Internet Protocol (IP) addresses – the ‘fingerprint’ identification address or location required to link up the two devices and accomplish the transmission objective, and (2) allowing multiple devices to use the same Internet connection.

There are several types of routers, but most routers pass data between LANs (local area networks) and WANs (wide area networks). A LAN is a group of connected devices restricted to a specific geographic area (typically what is used in private homes). A LAN usually requires a single router.

A WAN, by contrast, is a large network spread out over a vast geographic area. Large organizations and companies that operate in multiple locations across the country, for instance, will need separate LANs for each location, which then connect to the other LANs to form a WAN. Because a WAN is distributed over a large area, it often necessitates multiple routers and switches. *A switch forwards data packets between groups of devices in the same network, whereas a router forwards data between different networks.*

How does a router work?

Think of a router as an air traffic controller and data packets as aircraft headed to different airports (or networks). Just as each plane has a unique destination and follows a unique route, each packet needs to be guided to its destination as efficiently as possible. In the same way that an air traffic controller ensures that planes reach their destinations without getting lost or suffering a major disruption along the way, a router helps direct data packets to their destination IP address.

In order to direct packets effectively, a router uses an internal routing table — a list of paths to various network destinations. The router reads a packet's header to determine where it is going, then consults the routing table to figure out the most efficient path to that destination. It then forwards the packet to the next network in the path.

What is the difference between a router and a modem?

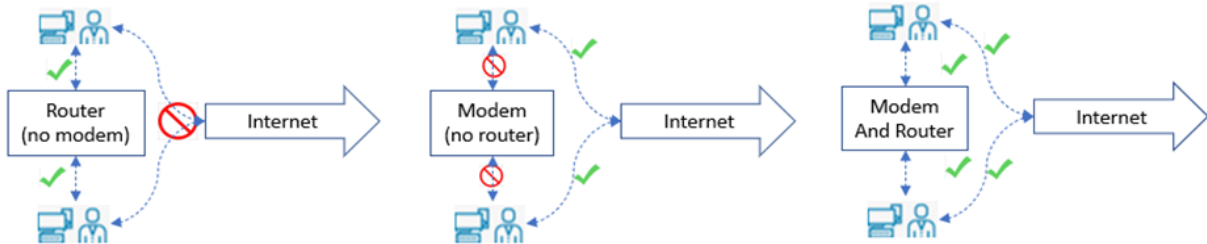
Although some Internet Service Providers (ISPs, such as Comcast or AT&T – service companies that provide an internet service) may combine a router and a modem within a single device, routers and modem are not the same thing. Each plays a different but equally important role in connecting networks to each other and to the Internet.

A router forms networks and manages the flow of data within and between those networks, while a modem connects those networks to the Internet. Modems forge a connection to the Internet by converting analog signals from an ISP into a digital signal that can be interpreted by any connected device. A single device may plug into a modem in order to connect to the Internet; alternately, a router can help

distribute this signal to multiple devices within an established network, allowing all of them to connect to the Internet simultaneously.

Think of it like this:

- **No modem, has router.** If Bob has a router, but no modem, he will be able to create a LAN and send data between the devices on that network. However, he will not be able to connect that network to the Internet.
- **No router, has modem.** Alice, on the other hand, has a modem, but no router. She will be able to connect a single device to the Internet (for example, her work laptop), but cannot distribute that Internet connection to multiple devices (say, her laptop and her smartphone).
- **Modem and Router.** Carol, meanwhile, has a router and a modem. Using both devices, she can form a LAN with her desktop computer, tablet, and smartphone and connect them all to the Internet at the same time.



What are the different types of routers?

In order to connect a LAN to the Internet, a router first needs to communicate with a modem. There are two primary ways to do this:

- **Wireless router:** A wireless router uses an Ethernet cable to connect to a modem. It distributes data by converting packets from binary code into radio signals (signals that can be transmitted through the air or even outer space), then wirelessly broadcasts them using antennae. (These radio signals can be intercepted by anyone, a next door neighbor for instance, and that is why such broadcasts need to be encrypted and secure to maintain secrecy). Wireless routers do not establish LANs; instead, they create WLANs (wireless local area networks), which connect multiple devices using wireless communication.
- **Wired router:** Like a wireless router, a wired router also uses an Ethernet cable to connect to a modem. It then uses separate cables to connect to one or more devices within the network, create a LAN, and link the devices within that network to the Internet.

In addition to wireless and wired routers for small LANs, there are many specialized types of routers that serve specific functions:

- **Core router:** Unlike the routers used within a home or small business LAN, a core router is used by large corporations and businesses that transmit a high volume of data packets within their network. Core routers operate at the "core" of a network and do not communicate with external networks.
- **Edge router:** While a core router exclusively manages data traffic within a large-scale network, an edge router communicates with both core routers and external networks. Edge routers live at the "edge" of a network and use the BGP (Border Gateway Protocol) to send and receive data from other LANs and WANs.

- *Virtual router:* A virtual digital router is a software application that performs the same function as a standard hardware physical router. It may use the Virtual Router Redundancy Protocol (VRRP) to establish primary and backup virtual routers, should one fail.

What are some of the security challenges associated with routers?

Vulnerability exploits: All hardware-based routers come with automatically installed software known as firmware that helps the router perform its functions. Like any other piece of software, router firmware often contains vulnerabilities that cyber attackers can exploit (one [example](#)). Router firmware needs to be updated regularly.

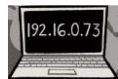
DDoS attacks: Small and large organizations often are the targets of distributed denial-of-service (DDoS) attacks directed at their network infrastructure. (DDoS is an attack where an attacker causes a targets network to be flooded with enormous amounts of non-sensical data and information – often generated by computer programmes – botnets - that generate the nonsense, which clogs and stops up the system from performing legitimate tasks – consumes all of the computer memory and operating capacity - and consequently denying desired internet service to communicate with the target network). Unmitigated (meaning processes are not in place to stop DDoS) network layer DDoS attacks can overwhelm routers or cause them to crash, resulting in network downtime. Such attackers sometimes may request ransom payments as in incentive to stop the attacks.

Administrative credentials: All routers come with a set of admin credentials for performing administrative functions. These credentials are originally set to default values, such as "admin" as the username and "admin" as the password. The username and password must be reset to something more secure: attackers are aware of the common default values for these credentials and can use them to gain control of the router remotely if they are not reset.

See Appendix 1, how modems work.

The router/modem ...

- Wirelessly (or by cable connection) communicates between Client A’s computer (which is identified with a unique one-of-a-kind Internet Protocol – IP – address number (like a fingerprint) – not unlike one’s home address, a number address used to identify the location of the home or the computer) and her modem, and the modem communicates between Client A’s computer and the Internet Service Provider - ISP (such as Comcast or AT&T – a company providing a service of connecting computers to the world wide web internet).



(IP address example)

- The modem converts digital ‘fixed or discrete or lumpy’ electronic signals from the computer (a mixture of “0s” or “1s” in computer binary code, sort of like an on and off switch, the language the computer understands) and



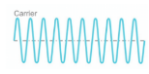
(example discrete digital electronic signal)



- converts that ‘lumpy’ digital language to a continuously ‘variable’ analog signal language understood by the internet (‘non-fixed’ or variable sine wave like oscillating signals that can vary continuously, sort of like gradually turning up the volume on a speaker instead of clicking between fixed discrete individual volume levels),

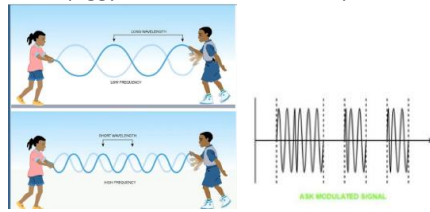


(example analog sine wave signal)



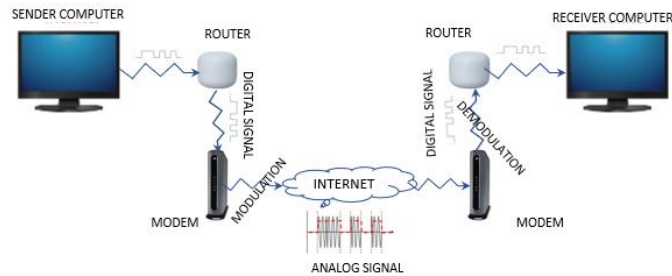
- and the converted or modulated digital signal piggy-backed onto the internet service provider electronic analog normally rhythmic sine wave signals and such conversion/piggyback process referred to as **modulation** (sort of like two

people holding a jump rope, wobbling it up and down in a normal rhythmic sine wave pattern and one of the wobblers adding an extra piggyback wobble that disrupts the normal rhythm of the oscillating rope).



(Example digital signal piggybacked on analog signal)

- The disrupted piggyback burdened analog signals are transported through cables or dish networks over the internet and routed to a destination computer IP (Internet Protocol) address (a one of a kind unique identification number or address that identifies the location and owner of sent and received internet communications), and the disrupted piggyback analog signal converted by the receiver's modem back to the original digital signal and such deconversion process is defined as demodulation (the language of the receiving computer).



- The sender and receiver Internet Protocol (IP) addresses are used to link the sender with the receiver...sort of like mailing a letter from your home address (and the envelope containing the senders return address) and the envelope addressed to the receiving party. The two addresses are needed to link the communication and determine the destination objectives of the electronic signals.

ARTICLE 13

HOW WILL BLOCKCHAIN PROMOTE THE NEW WEB3?

KEY TAKE AWAY: MORE PARADIGM CHANGES ON THEIR WAY.

Reshaping the Internet in the interest of security, privacy, and reliability

A dominant narrative around the contemporary Internet is the global proliferation of user-generated content and applications. Thanks to the rise of social media, online publishing platforms, and other tools — a trend described as ‘Web 2.0’ — it’s easy for individuals and organizations alike to share content and web experiences with a broad audience.

However, this accessible, democratic approach doesn’t extend to every aspect of the Internet. When it comes to hosting web applications, the opposite is often true — because a person or organization who wants to launch an application currently has few realistic choices about where it is stored and run.

These limitations are changing, such as the use of Blockchain technology, observing a slow but steady move towards a more decentralized Internet — with implications for security, privacy, and reliability.

The modern Internet is centralized in many ways

When an individual or organization wants to launch a web application, they have historically had relatively few places to host it. The application may be able to live in a private, on-premise server or data center if it has few users and uses limited bandwidth. When bandwidth requirements increase, however — or if the organization wants to provide the fast, secure experience many users now expect (want it yesterday mentality) — all but the largest, most well-resourced organizations find that their only economical choice is some form of cloud hosting. Cloud hosting provides many performance, security, and flexibility benefits compared with on-premises hosting. But the centralization of data in a select few cloud providers also creates challenges, including:

- **Service outage:** Hosting web application data on third-party servers can introduce a single point of failure unless implemented with appropriate redundancies in the infrastructure. This can become a problem when cloud providers suffer outages or are having connectivity issues to the Internet.
- **Performance risks for global audiences:** Cloud providers operate a relatively limited number of massive data centers, and cloud users often must choose which geographic region their application will live in. If an application’s users are far from its servers, they may experience latency due to the traffic’s long journey.
- **Vendor lock-in:** Migrating from one cloud service to another can be extremely challenging. Should the cloud vendor’s service quality decline — or should it institute unfair pricing policies — organizations may struggle to find a better alternative.

These challenges are not reasons to abandon the cloud. But they may explain a recent trend — the conception, and slow emergence, of a model for a decentralized Internet, driven by technologies like the Blockchain. This model is often referred to as “Web3”.

Web3: What and how?

Web3 is expected to more closely align with the original vision of one of the Internet’s architects, who envisioned a distributed network with no central authorities or single point of failure. Originally called the “Semantic Web”, Web3 could be an intelligent Internet that understands everything a user conveys in

both content and context, processing information with human-like intelligence. This will be achieved through the interconnection and decentralization of data across networks that operate through decentralized protocols.

Blockchain in particular is presumed to be one of the most pivotal technologies necessary for the infrastructure of Web3. Blockchain emerged in 2009 with the creation of Bitcoin a cryptocurrency based network. While cryptocurrency is the most popular use for Blockchain, the technology offers the potential to serve a very wide range of applications. A follow on Blockchain concept is Ethereum, a decentralized, open-source, and distributed computing platform that enables the creation of smart contracts (contracts that are self executing, not requiring performance by individual contracting parties) and decentralized applications. While Bitcoin and Ethereum are both blockchain-based, they have several key differences. Bitcoin is a cryptocurrency and store of value solely intended for transactions. Ethereum, like Bitcoin, *can* be used for transactions, but more important is its enablement of decentralized apps, or **DApps** - computer applications that run on a decentralized computing system.

The Ethereum blockchain is thought to be the ideal open and trustless platform to serve as the infrastructure of a decentralized Internet, that can provide useful applications disconnected from cryptocurrency activities, such as Interplanetary File System (IPFS) technology. One of the key features of Ethereum, unlike Bitcoin, Ethereum open source code allows Developers to piggy back on the Ethereum platform and produce applications that have nothing to do with cryptocurrency. (IPFS can replace the location based hypermedia server protocols http and https to distribute the World Wide Web). This will result in an online Internet experience that is vastly more powerful and tailored to the user and will revolutionize the interconnection of the Internet, applications, and the physical world. And with quantum computing driving the system, who knows what boundaries can be abolished. Through this, there will be huge improvements in terms of privacy and security thanks to the decentralization of data and privacy-preserving cryptographic and computational techniques. (Of course, too much privacy may be a bridge too far if IPFS just becomes a safe haven and port for malicious folk to successfully hide and conduct their malfeasance business).

Challenges to Web3

While Web3 promises to radically change and disrupt (not destruct) the Internet and its ability to provide value to users around the world, key hurdles must be overcome before it can be adopted en masse. Today, there are several issues with decentralized networks that are impeding the rise of Web3, including speed and scale (ability to take a laboratory scale experiment and convert it, scale it up, to commercial use. (What might work in an easily controlled test tube environment may not work in an industrial size tank exposed to the elements).

Despite providing better security, the decentralized web is currently much slower than the centralized web due to the need for *authentication nodes* (special participants in a Blockchain network that provide a verification service and paid a cryptocurrency fee for doing so, that Blockchain transactions and their participants are authentic). Whereas a centralized app can process an incredible amount of requests at one time (think credit card transactions managed by MasterCard, visa or others), a decentralized Blockchain app pales in comparison by orders of magnitude less.

Scalability is also an ongoing issue. Because the Ethereum Network is made up and secured by over 8,000 verification nodes, every transaction must be processed by all such nodes (and this redundant processing and consensus voting, being part of the miracle of why Blockchain claims best in class security policeman).

This can lead to network congestion and is a significant limiting factor in Ethereum's ability to handle the huge volume of enterprise-class applications of tomorrow.

The benefits of Web3

Once the challenges that currently stand in the way of Web3 are solved, it will provide impactful (disruptive) solutions to some of the Internet's most persistent problems. For example, while the centralized apps of today can experience downtime for any number of reasons, Distributed Apps or DApps and Web3 servers promise to be more resilient with a much-reduced risk of downtime as they will be run on Ethereum's decentralized network of tens of thousands of computers. And with greater adoption and increasing network effects, the reliability of the Web3 Internet will continue to improve. That's what makes Blockchain successful...it demands a large number of participants. One for all and all for one.

Similarly, Web3 will reduce if not eliminate the volume and efficacy of Distributed Denial of Service (DDoS) attacks (a malicious party flooding a target computer site with meaningless data and information that consumes all of the computing capacity, effectively shutting down such network, resulting in denial of service attack), further improving reliability. With the Peer-to-Peer networks (users dealing directly with one another and not having to go through a centralized controlling middleperson such as a bank) that secure the Ethereum Blockchain rather than centralized servers, bad actors will not have the same ability to disrupt Internet services as easily as they can now. There will no longer be single points of failure, allowing the network to function as normal regardless of participants being attacked or taken out.

What's next

While the issues of latency (the time it takes for information to be transmitted), scale (volume of transactions handled), and reliability (performing consistently well) are still challenges in the transition to Web3.

Currently, interacting with the Ethereum Network is difficult and requires running complex software, including downloading and cryptographically verifying massive amounts of data, which creates technical barriers and can preclude those with low-power devices.

ARTICLE 14

WHAT IS QUANTUM COMPUTERS?

KEY TAKE AWAY: WILL QUANTUM COMPUTING MAKE CRYPTOGRAPHY USELESS? NO MORE SECRETS

What is quantum computing?

A quantum computer uses the properties of quantum mechanics to perform calculations. Quantum computers are much faster at certain types of calculations than classical computers (meaning any computing device in wide use today, like smartphones, servers, and desktop computers). Most importantly, quantum computing may be able to solve certain extremely difficult math problems that classic computing cannot efficiently solve at all, which would put current encryption methods at risk and expose sensitive data.

Imagine finding a chapter in a book by turning page by page until arriving at the desired place. Now imagine instead consulting the table of contents first, and almost instantly turning to the correct chapter. Quantum computing is more like the experience of using a table of contents: it examines all possible solutions to a calculation quickly and simultaneously, instead of trying different solutions until arriving at the correct one.

Technically, a classical computer can do any calculation that a quantum computer can do — given enough time. But a classical computer might need centuries or millennia to solve a problem a quantum computer could theoretically solve in minutes.

In practice, researchers have produced just a handful of cases where a quantum computer solved a problem faster than a classical computer. Quantum computers are difficult to build and unstable once built. But if the challenges of constructing quantum computers are solved, quantum computing might permanently transform technology.

What are bits and qubits?

A classical computer stores information in a series of **bits**, in binary code. A bit is the smallest possible unit of information; its value is either 0 or 1 (in effect an on-off switch with our without voltage – that on-off voltage clicking is what computers use to think and operate).

A quantum computer stores information in **qubits** rather than bits. A qubit can have a value of 0, 1, or a mix of both states – **at the same time** – welcome to the weird world of quantum mechanics – Einstein *spooky at a distance* thinking (the technical term for such a mix is “superposition”). In fact, a qubit’s value is uncertain — unlike a classical bit where a bit is either a 1 or 0 and known. A qubit’s value remains indeterminate until someone **observes** it – just like quantum mechanics, things are weird until you observe it.

As a result, a quantum computer can hold multiple states, or versions, of information at once (until observed). This enables it to process solutions to calculations at an exponentially faster pace compared to a regular computer — just as a team of people performing multiple tasks simultaneously will complete a project faster than one person doing all the tasks on their own.

Imagine a segment of information as a globe. A bit can sit either at the globe's north pole or south pole. A qubit can sit anywhere on the surface of the globe (until it is observed then and only then is its location knowable), vastly increasing the informational possibilities it can contain.

On a mechanical level, of course, bits and qubits are not actually globes. A bit is a tiny section of a computer that either holds an electrical charge (1) or does not hold an electrical charge (0). A qubit is the uncertain, unstable position of an electron within an atom. Its location defined by probabilistic analysis, could be here or could be there, compared to bits are deterministic analysis, cause and effect, it's either a 1, and if it is it is not 0, or the reverse.

What are the challenges of building quantum computers?

Very few quantum computers have been constructed. Those that have been built are small, unstable, lab scale and not usable outside of laboratory conditions.

This is because quantum computing faces a some major challenges:

- Interference from the outside environment
 - Qubits are fragile. Noise, vibration, temperature changes, and electromagnetic waves can all inhibit or destroy the internal state of a qubit. To operate properly, quantum computers need to be in highly controlled environments that lack these and other types of interference. Such environments are difficult to construct and maintain outside of a laboratory.
- Environmental factors impact classical computers as well — for instance, high temperatures or strong magnetic forces can slow or destroy a computer. But the problem is much more severe for quantum computers, to the point that it is uncertain if they can operate in real-world conditions.
- Error correction
 - Quantum computers are less stable in general than their classical counterparts. This makes them more prone to errors. All computers commit errors, which is why classical computers have built-in memory and processors dedicated to error correction. But quantum computers have to devote a lot more resources to error correction than classical computers, relative to their processing ability.
- Temperature
 - To keep qubits stable, quantum computers have to be kept extremely cold — just a few degrees above absolute zero. This makes it hard to operate them outside of highly controlled laboratory environments.

The result of these and other challenges is that very few quantum computers have been constructed with more than a handful of qubits. (A 256-qubit quantum computer was announced in 2021, and one firm hopes to construct a 1,000-qubit quantum computer by 2023.)

What impact would quantum computing have on the world?

The full impact of quantum computing is difficult to determine, as it is still unclear if large-scale quantum computers are feasible, let alone if mass production of such computers is possible. This contrasts with classical computing — in most societies, miniature computers are used in almost all aspects of life, and many people carry the equivalent of a supercomputer in their pockets (as smartphones).

Powerful, stable quantum computers could have major positive impacts on society. But it is also clear that such computers would put privacy and security at risk in new ways.

Potential positive effects

There are many possible applications of quantum computers. With more powerful computers, the financial industry may be able to help more accurately analyze and predict the stock market. Climatologists might be able to analyze and predict weather patterns more precisely. Transportation systems could become more efficient if quantum computers can better predict traffic patterns.

All these outcomes are still theoretical. And even if large-scale, highly stable quantum computers could be constructed, their processing results would still only be as accurate as the data they are fed. Even so, quantum computing could have a major positive impact on these or similar areas.

Current encryption methods would break

Today, sensitive information is often protected through the use of encryption. Encryption is the process of encoding a message using a key, so that no one can read the message except someone who has the key. Encryption protects personal data users enter on websites (through TLS), business data stored on hard disks and in servers, confidential government data, and other sensitive information.

Many types of encryption rely on difficult math problems, such as prime factorization, to protect data. The difficulty (“hardness”) of these problems ensures that the encryption cannot be broken within a feasible amount of time. Although well-known algorithms for breaking encryption exist, it is always possible to use larger encryption keys, requiring exponentially more time (for classical computers) to find the key and break the encryption.

However, quantum computers can theoretically solve the hard problems used in currently deployed encryption methods. In this scenario, increasing key sizes does not strengthen the difficulty of the problem exponentially. Thus, breaking encryption could take significantly less time. This would allow quantum computers to break most current encryption methods, putting any encrypted data at risk of exposure. Because of the complexity of quantum computing, ordinary hackers (what ever that is) will probably not have access to quantum computers... But rogue nation states may...and their objectives for using QC is speculative at best...and humanitarian aide may not be at the top of their to do list.

Techy Stuff

WARNING...PROCEED AT YOUR OWN RISK! The below additional technical discussion regarding quantum computers is exciting stuff...but the reader is free to skip all this as being TMI and MTINTK (too much information and more than I need to know).

Superfluids

Desktop computers use a fan to get cool enough to work (since heat generated by the computer can adversely affect its operation). Quantum processors need to be VERY cold – about a hundredth of a degree above absolute zero. To achieve this, super-cooled superfluids are used to create superconductors.

Superconductors

At those ultra-low temperatures certain materials in quantum computer processors exhibit an important quantum mechanical effect: electrons move through them without resistance. This makes them “superconductors” and the ability for electrons to move very fast...close to the speed of light.

When electrons pass through superconductors they match up, forming “Cooper pairs.” These pairs can carry a charge across barriers, or insulators, through a process known as quantum tunneling. Quantum

tunneling is that quantum mechanics bizarre behaviour where a particle can mysteriously travel through a barrier and appear on the other side (think a person standing in front of a brick wall mysteriously appears on the other side of the wall. What Einstein called 'spooky at a distance').

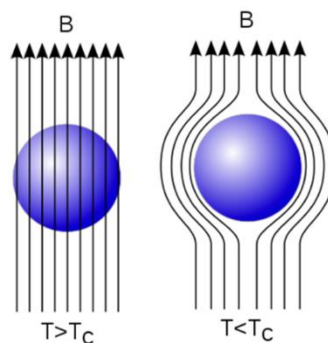
Josephson Junction

Two superconductors placed on either side of an insulator (a barrier) form what is called a Josephson junction (named after its inventor).

Josephson junction is what gives quantum computers their ability to break speed barriers (a particle being able to go through solid barriers and appear on the other side).

A Josephson junction is made by sandwiching a thin layer of a non-superconducting material (the cheese in a cheese sandwich) between two layers of superconducting material (the two slices of sandwich bread).

- A non-superconducting material is a material, an insulator, through which electricity or electrons cannot travel through (examples: dry leather or calcium iron arsenide (Ca_2As_2)).
- Superconductivity which is a property of superconductors, are physical properties observed in certain materials where electrical resistance vanishes and magnetic flux fields are expelled from the material. Expulsion of magnetic flux fields occurs right before a material transitions into superconductivity state (the state of no resistance to the flow of electrons). The magnetic flux expulsion result is illustrated by the following diagram that shows an external magnetic field (say from a nearby magnet) is being rerouted around the superconducting particle – hence the expulsion of the magnetic flux field. (Note: T is temperature and T_c is the temperature at which superconductivity starts to occur. The diagram on the left, the temperature is greater than T_c (its too hot so the magnetic flux field passes through the material) but less on the right (so it is cold enough for the magnetic flux field to be expelled or pushed away from the material).



Any material exhibiting these properties is a superconductor. Unlike an ordinary metallic conductor, whose resistance decreases gradually as its temperature is lowered even down to near absolute zero, a superconductor has a characteristic critical temperature below which the resistance drops abruptly to zero. An electric current through a loop of superconducting wire can persist indefinitely with no power source. Examples of superconductor material includes chemical elements (e.g. mercury or lead), and alloys (such as niobium–titanium, germanium–niobium, and niobium nitride).

The Josephson junction devices are named after Brian Josephson, who predicted that pairs of superconducting electrons could “tunnel” right through the non-superconducting barrier from one superconductor to another as if the barrier was not there. He also predicted the exact form of the current

and voltage relations for the junction. Experimental work proved that he was right, and Josephson was awarded the 1973 Nobel Prize in Physics for his work.

To understand the unique and important features of Josephson junctions, it's necessary to understand the basic concepts and features of superconductivity. If you cool many metals and alloys to very low temperatures (within 20 degrees or less of absolute zero), a phase transition occurs. At this "critical temperature," the metal goes from what is known as the normal state, where it has electrical resistance, to the superconducting state, where there is essentially no resistance to the flow of direct electrical current. Newer high-temperature superconductors, which are made from ceramic materials, exhibit the same behavior but at warmer temperatures, allowing cheaply made liquid nitrogen to act as the super coolant fluid.

What occurs is that the electrons (fundamental quantum particles comparable to quarks, that cannot be further subdivided into other particles that would otherwise make up the electron) in the superconductor metal become paired (which is odd since two negative charges would normally repel each other – like charges repel – but in the quantum world things are not quite as they seem – think Alice in Wonderland). Above the critical temperature, the net interaction between two electrons is repulsive because of the like negative charges. Below the critical temperature, though, the overall interaction between two electrons becomes very slightly attractive, a result of the electrons' interaction with the ionic lattice (the metal's internal molecular ionic structure or architecture sort of like the honeycomb of a beehive) of the superconductor metal. [Molecular compounds are formed when atoms are joined together by sharing of electrons while ionic compounds are formed due to the transfer and not sharing of electrons].

This very slight attraction allows the paired electrons to drop into a lower energy state, opening up an energy "gap." Because of the energy gap and the lower energy state, electrons can move (and therefore current can flow) without being scattered by the ions (electrically charged atoms) of the lattice. When the ions scatter electrons, it causes electrical resistance in metals. There is no electrical resistance in a superconductor, and therefore no energy loss. There is, however, a maximum supercurrent that can flow, called the critical current. Above this critical current the material is normal (meaning the density or electron's per second flow, reach a peak and can't move any more volume of electrons).

There is one other very important property: when a metal goes into the superconducting state, it expels all magnetic fields, as long as the magnetic fields are not too large. (Magnetic fields arise when electron's in superconductor lattice move, since moving electron's generate magnetic fields – and this is how clamp on ammeter's work, measuring the magnetic flux around a wire because current is flowing through the wire and correlating that measurement to current or electron amp flow). At the superconducting low temperature critical state, the movement of electron's is so small that magnetic field generation essentially stops).

In a Josephson junction, the nonsuperconducting barrier separating the two superconductors must be very thin. If the barrier is an insulator, it has to be on the order of 30 angstroms thick or less. If the barrier is another metal (nonsuperconducting), it can be as much as several microns thick. At and below the critical current, a supercurrent can flow across the barrier; electron pairs can tunnel across the barrier without any resistance. But when the critical current is exceeded, another voltage will develop across the junction. That voltage will depend on time—that is, it is an Alternating Current voltage (with wave properties). This in turn causes a lowering of the junction's critical current, causing even more normal current to flow—and a larger Alternating Current voltage.

Like I said...perhaps more than you ever wanted to know...

ARTICLE 15

WHAT IS HACKING ALL ABOUT? (not a cough)

- **UNDERSTANDING: COMMON HACKING TECHNIQUES;**
- **PROTECTING AUTHENTICATION AND TAMPERING;**
- **WHAT IS A BRUTE FORCE ATTACK – WHY DO IT – BRUCE FORCE TECHNIQUES;**
- **HOW TO PROTECT PASSWORDS AND CRYPTOGRAPHIC KEYS**

KEY TAKE AWAY: CYBER THIEFS ARE SMART FOLK...LOCKS KEEP HONEST PEOPLE HONEST...

Common Hacking Techniques

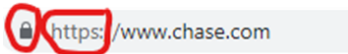
A hacker can learn about confidential unauthorized information. Following are common hacking techniques used to gain access to confidential information in an unauthorized way.

1. **Bait and Switch:** an attacker buys advertising spaces on the websites. Later, when an innocent user clicks on the ad, they are directed to a webpage that's infected with malware. The attacker can then install malware or adware on the innocent's computer. The ads and download links shown in this technique are very attractive and innocent end up clicking on the same. The hacker can run a malicious program that the innocent user believes to be authentic. This way, after installing the malicious program on the innocent user's computer, the hacker gets unprivileged access to their computer.

2. **Cookie theft:** The cookies in browsers store personal and confidential data such as browsing history, username, and passwords for different sites accessed. Once the hacker gets access to cookies, they can authenticate and substitute themselves as the innocent user on a browser. A popular method to carry out this attack is to manipulate or hijack an innocent user's Internet Protocol address (IP address is a computer users digital address or fingerprint used to communicate with other IP addresses through the internet) to pass through attacker's computer. Also known as SideJacking or Session Hijacking, this attack is easy to carry out if the innocent user is not using Transport Layer Security – TLS [**Transport Layer Security (TLS)** is a cryptographic protocol] (procedure) and encryption security computer software coding designed to provide communications security over a computer network. TLS encrypts data sent over the Internet to ensure that eavesdroppers and hackers are unable to see what you transmit which is particularly applicable for private and sensitive information such as passwords, credit card numbers, and personal correspondence. The TLS security protocol is widely used in applications such as email, instant messaging, and voice over Internet Protocol (voice communication – a phone call – over the internet and not through conventional telephone lines), but its use in securing HTTPS (Hypertext Transfer Protocol Secure - it uses cryptography for secure communication over a computer network, such as the internet) remains the most publicly visible. HTTPS is an implementation of TLS encryption – (HTTPS piggybacks HTTP entirely on top of TLS, the entirety of the underlying HTTP/HTTPS protocol can be encrypted. This includes the user's Universal Resource Locator or URL – the web address, query parameters, headers, and cookies (which often contain identifying information about the user)).
 - a. The TLS protocol aims primarily to provide security, including privacy (confidentiality), integrity, and authenticity through the use of cryptography, such as the use

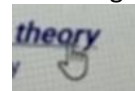
of authentication certificates, between two or more communicating computer applications.

- b. TLS, an upgraded formerly SSL [Secure Sockets Layer – SSL - a digital certificate of authenticity that verifies a website’s identity and allows a secure encrypted connection between servers and browsers].



- c. The familiar padlock icon and HTTPS hypertext wording, located next to the website’s URL (Universal Resource Locator address, the website address, in this example, *https://www.chase.com*) in the address bar indicates TLS/SSL security and HTTPS protection [Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP)].

- d. HTTPS is the foundation of data communication for the World Wide Web, where **hypertext documents** (**hypertext** being a software system that links topics displayed on the computer monitor screen to related information and graphics, which are typically accessed by a point-and-click method, and **hypertext document** is text displayed on the computer monitor screen which contains links - hypertext links, the usual blue bold typed texts - to other texts) include hyperlinks to other website resources that the user can easily access, for example by a mouse click or by tapping the computer monitor screen when a web browser is displayed on the screen. HTTPS uses cryptography for secure communication over the computer network, and is widely used on the Internet. Clicking a hypertext link displayed on the computer monitor screen – aka hyperlink (example [hyperlink](#)) will automatically send the user to another part of the current document or a separate document entirely. The mouse clicking process is visualized by the mouse pointer on the computer monitor screen and when that pointer lands on a hypertext link, the familiar pointed finger icon pops up indicating the link can



be activated by clicking the mouse when the finger icon is shown.

Hypertext links are a fundamental building block of the Internet, allowing users to quickly and efficiently navigate in a web browser from page to page and from site to site. The principal motivations for HTTPS are its communication efficiencies, security authentication of the accessed website and protection of the privacy and integrity of exchanged data while it is in transit through the internet. It protects against eavesdropping and tampering. The authentication aspect of HTTPS requires a trusted third party to sign server-side digital certificates of authenticity (i.e., not a hacker).

- 3. **ClickJacking Attacks:** ClickJacking is also known by a different name, User Interface or UI⁵³ Redress. In this attack, the hacker hides the actual UI (the computer monitor onscreen hot

⁵³⁵³⁵³ **UI** means **User Interface** which is the environment how human’s and computer’s interface with each other when the human is using the computer. Generally, the goal of [user interface design](#) is to produce a user interface that makes it easy, efficient, and enjoyable (user-friendly) for a human to operate a computer in the way which produces the desired result (i.e. maximum [usability](#)). This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the computer minimizes undesired outputs to the user.

User interfaces are composed of one or more layers, including a **human-computer interface (HMI)** that interfaces computers with physical [input hardware](#) operated by a human such as keyboards, mice, or game pads, and output hardware that communicates with human’s such as [computer monitors](#) (visual contact), speakers (audible contact), and [printers](#) (visual and touch contact). A device that implements an HMI is called a [human interface device](#) (HID, such as a computer keyboard device).

button, hypertext link or icon) where the victim is supposed to click. This behavior is very common in-app download, movie streaming, and torrent websites. While they mostly employ this technique to earn advertising dollars, others can use it to steal your personal information. In other words, in this type of hacking, the attacker hijacks the clicks of the victim that aren't meant for the exact page, but for a page where the hacker wants you to be. It works by fooling an internet user into performing an undesired action by clicking on the hidden link.

4. Virus, Trojan. Viruses or Trojans are malicious software programs that get installed into the victim's system and keep sending the victim's data to the hacker. They can also lockup files, server fraud advertisement, divert traffic, sniff your data, or spread on all the computers connected to your network.

5. Phishing. Phishing is a hacking technique using which a hacker replicates (creates a fake copy) the most-accessed sites and traps the victim by sending that spoofed link. Combined with social engineering, it becomes one of the most commonly used and deadliest attack vectors (a path that a hacker takes to exploit cybersecurity vulnerabilities). Once the victim tries to login or enters some data, the hacker gets the private information of the target victim using the trojan running on the fake site.

Additional UI layers may interact with one or more human senses, including: tactile UI ([touch](#)), visual UI ([sight](#)), auditory UI ([sound](#)), olfactory UI ([smell](#)), equilibria UI ([balance](#)), and gustatory UI ([taste](#)).

Composite user interfaces (CUIs) are UIs that interact with two or more senses. The most common CUI is a [graphical user interface](#) (GUI), which is composed of a tactile (touch, such as a mouse) UI and a visual (the computer monitor screen) UI capable of displaying [graphics](#). When sound is added to a GUI, it becomes a *multimedia user interface* (MUI). There are three broad categories of CUI: *standard*, *virtual* and *augmented*. **Standard** CUI use standard human interface devices like keyboards, mice, and computer monitors. When the CUI blocks out the real world to create a [virtual reality](#), (VR) the CUI is wholly **virtual** and uses a *virtual reality interface*. A typical VRI is the virtual reality headset used by human's to enter a virtual world of game playing or



virtual reality experience.

When the CUI does not block out the real world and creates [augmented reality](#) (AR), the CUI is augmented and uses an *augmented reality interface*. Almost any person with a smartphone can get access to augmented reality, making it more efficient than VR as a branding and gaming tool. AR morphs the mundane, physical world into a colorful, visual one by projecting virtual pictures, game activity and characters through a phone's camera or video viewer, thereby simultaneously keeping the human viewer emersed in real life reality environment (whose head is not isolated in a virtual reality headset) yet visually and audibly distracted by the unfolding of the virtual 'gaming' world taking place on the human viewed smartphone screen. Augmented reality is merely adding to the user's real-life experience. Differences between AR and VR are: AR uses a real-world setting while VR is completely virtual; AR users can control their presence in the real world; VR users are controlled by the system; VR requires a headset device, but AR can be accessed with a smartphone; AR enhances both the virtual and real world while VR only enhances a fictional virtual reality. When a UI interacts with all human senses, it is called a qualia interface, named after the theory of [qualia](#) (the qualia knowledge argument aims to establish that human conscious experience involves non-physical properties – its all in the mind sort of thing. It rests on the idea that someone who has complete **physical** knowledge about another human's conscious being, might yet lack knowledge about how it feels to have the experiences of that being (can't stand in the shoes of the observed individual and actually experience what they experience, either physically or emotionally). CUI may also be classified by how many senses they interact with as either an X-sense virtual reality interface or X-sense augmented reality interface, where X is the number of senses interfaced with. For example, a [Smell-O-Vision](#) is a 3-sense (3S) Standard CUI with visual display, sound and smells; when *virtual reality interfaces* interface with smells and touch it is said to be a 4-sense (4S) virtual reality interface; and when *augmented reality interfaces* interface with smells and touch it is said to be a 4-sense (4S) augmented reality interface.

6. Eavesdropping (Passive Attacks) Unlike other attacks that are active in nature, using a passive attack, a hacker can monitor the computer systems and networks to silently gain some unwanted information. The motive behind eavesdropping is not to harm the system but to get some information without being identified. These types of hackers can target email, instant messaging services, phone calls, web browsing, and other methods of communication. Those who indulge in such activities are generally black hat hackers, government agencies, etc.

7. Fake WAP (an abbreviation for Wireless Application Protocol, a system that allows devices such as cellphones to connect to the internet and the connection may be an honest wi-fi connection or fake dishonest one). Just for fun, a hacker can use software to fake a wireless access point. This WAP connects or piggybacks on to the official public place WAP. Once you get connected to the fake WAP, a hacker can access your data. It's one of the easier hacks to accomplish and one needs a simple software and wireless network to execute it. Anyone can name their WAP as some legit name like "Heathrow Airport WiFi" or "Starbucks WiFi" and start spying on you. One of the best ways to protect yourself from such attacks is by using a quality VPN service.

VPN. A VPN or virtual private network, uses encryption to protect communication between to points (such as one's iPhone and the web). What VPN accomplishes is that before any intruder can enter a VPN protected network, the intruder must first prove their identity (and they are authorized to enter the network), information or data to be sent or retrieved from the network be of a certain type and the delivery address of communicated data and information must be verified (and it is an authorized recipient). VPN's are capable of encrypting two different ways:

1. Transport: sets up a secure, encrypted link across the internet's wires (downside is that 'headers' on the sent data are sent in the clear, which is like wrapping a present then putting on a label what is inside; and
2. Tunneling: encrypts data and its headers ("payload") being sent.

8. Waterhole attacks. If you are a big fan of Discovery or National Geographic channels, you could relate easily with the waterhole attacks. To poison a place, such as a river, it will hit the entire stretch of animals during summer. Comparably, a hacker hits the most accessible physical point of the victim and that point could be a coffee shop, a cafeteria, etc. Once the hacker is aware of your timings, they can use this type of attack to create a fake Wi-Fi access point. Using this they can modify your most visited website to redirect them to you to get your personal information. As this attack collects information on a user from a specific place, detecting the attacker is even harder. One of the best ways to protect against such types of hacking attacks is to follow basic security practices and keep your software/OS updated.

9. Denial of Service (DoS\DDoS) A Denial of Service attack is a hacking technique of taking down a site or server by flooding that site or server with a huge amount of traffic so that the server is unable to process all the requests in real-time and finally crashes down. In this popular technique, the attacker floods the targeted machine with tons of requests to overwhelm the resources, which, in turn, restricts the actual requests from being fulfilled. For DDoS attacks, hackers often deploy botnets or zombie computers that have only one task, that is, to flood your system with request packets. With each passing year, as the malware and types of hackers keep getting advanced, the size of DDoS attacks keeps increasing.

10. **Keylogger** A keylogger is a simple software that records the key sequence and strokes of your keyboard into a log file on your machine. These log files might contain your personal email IDs and passwords. Also known as keyboard capturing, it can be either software or hardware. While software-based keyloggers target the programs installed on a computer, hardware devices target keyboards, electromagnetic emissions, smartphone sensors, etc. Keylogger is one of the main reasons why online banking sites give you an option to use their virtual keyboards. So, whenever you're operating a computer in a public setting, try to take extra caution.

11. **Key Stealing Attacks.** Digital keys should be kept secret and in safe places (don't store copies on one's computer, don't leave on sticky notes, watch out for shoulder grazers who peer over your shoulder as you enter keys on your computer). If an attacker gains access to the key they then have access to confidential information and data. Equally, key-length should be as long as possible – the longer it is, the harder it is for an attacker to determine or duplicate.

12. **Plaintext Attacks.** If a hacker already has access to plaintext message and an encrypted copy, they can compare the two and determine the encryption process and methodically assess each character in an attempt to determine the encryption/decryption keys...sort of like playing Wheel of Fortune, play around long enough with the plaintext/encrypted text character variations, a hacker might discover the key to the entire message.

13. **Pattern Recognition.** When trying to break an encrypted code, searching for a pattern is very useful. "E" is the most commonly used letter in the English alphabet, so looking for a repeated character in an encrypted message *may* mean that character is in plaintext an "e". The second most common letter is "T".

14. **Brute Force Attack.** Brute force attack is a trial and error guess method of trying every possible combinations of characters (often using coordinated and linked computers) against the encrypted data in an attempt to determine the encryption key.

15. **MAN IN THE MIDDLE ATTACK (MITM) – A SPECIAL HACKER,** In cryptography and computer security, a **man-in-the-middle, monster-in-the-middle, machine-in-the-middle, monkey-in-the-middle, meddler-in-the-middle, manipulator-in-the-middle (MITM), person-in-the-middle (PITM) or adversary-in-the-middle (AiTM) attack** is a cyberattack where the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other, as the attacker has inserted themselves through the Internet between the two parties. One example of a MITM attack is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages independently between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is routed through and controlled by the attacker in the middle. The attacker must be able to intercept all relevant messages passing between the two victims and inject new ones. This is straightforward in many circumstances; for example, an attacker within the reception range of an unencrypted Wi-Fi access point could insert themselves as a man-in-the-middle. As it aims to circumvent mutual authentication, a MITM attack can succeed only when the attacker impersonates each party endpoint sufficiently well to satisfy their expectations that each are dealing directly with the other (and not dealing with a MITM). Most cryptographic protocols include some form of endpoint authentication specifically to prevent MITM attacks. For example, TLS can authenticate one or both parties using a mutually trusted certificate authority. (TLS, or **Transport Layer Security** encrypts data sent over the

Internet to ensure that eavesdroppers and hackers are unable to see what you transmit which is particularly useful for private and sensitive information such as passwords, credit card numbers, and personal correspondence.

MITM attacks can be prevented or detected by two means: **authentication** and **tamper detection**.

- **Authentication** provides some degree of certainty that a given message has come from a legitimate source.
- **Tamper detection** merely shows evidence that a message may have been altered.

Authentication

Big problems with access to all networks (such as the internet, or even a local in-house computer network) is that of trust. A UserID and password are vehicles for controlling access to a network, but they really do not do anything about verifying that the User is who he or she claims to be, especially where the ID and password are stolen. Digital certificates and public/private encryption keys are methods used by computer security staff to help give the network an added level of ID to check and verify the authenticity of the User.

All cryptographic systems that are secure against Men In The Middle (a bogus person claiming to be someone they are not, and Users innocently communicating with such bogus person) attacks provide some method of authentication for messages. Most require an exchange of information (such as public keys) in addition to the message over a secure channel. Such protocols, often using key-agreement protocols, have been developed with different security requirements for the secure channel, though some have attempted to remove the requirement for any secure channel at all.

A public key infrastructure, such as Transport Layer Security, may harden Transmission Control Protocol against MITM attacks. In such structures, clients and servers exchange **digital certificates** which are issued and verified by a trusted third party called a certificate authority (CA). If the original key to authenticate this CA has not been itself the subject of a MITM attack, then the certificates issued by the CA may be used to authenticate the messages sent by the owner of that certificate. Use of mutual authentication, in which both the server and the client validate the other's communication, covers both ends of a MITM attack. If the server or client's identity is not verified or deemed as invalid, the session will end. However, the default behavior of most connections is to only authenticate the server, which means mutual authentication is not always employed and MITM attacks can still occur.

Attestments, such as verbal communications of a shared value, or recorded attestments such as audio/visual recordings of a public key hash, are used to ward off MITM attacks, as visual media is much more difficult and time-consuming to imitate than simple data packet communication. However, these methods require a human in the loop in order to successfully initiate the transaction.

In a corporate environment, successful authentication (as indicated by the browser's green padlock) does not always imply secure connection with the remote server. Corporate security policies might contemplate the addition of custom certificates in workstations' web browsers in order to be able to inspect encrypted traffic. As a consequence, a green padlock does not indicate that the client has successfully authenticated with the remote server but just with the corporate server/proxy used for SSL/TLS inspection.

HTTP Public Key Pinning (HPKP), sometimes called "certificate pinning," helps prevent a MITM attack in which the certificate authority itself is compromised, by having the server provide a list of "pinned" public

key hashes during the first transaction. Subsequent transactions then require one or more of the keys in the list must be used by the server in order to authenticate that transaction.

Tamper detection

Latency examination can potentially detect the attack in certain situations,^[19] such as with long calculations that lead into tens of seconds like hash functions. To detect potential attacks, parties check for discrepancies in response times. For example: Say that two parties normally take a certain amount of time to perform a particular transaction. If one transaction, however, were to take an abnormal length of time to reach the other party, this could be indicative of a third party's interference inserting additional latency in the transaction.

Quantum cryptography, in theory, provides tamper-evidence for transactions through the no-cloning theorem. Protocols based on quantum cryptography typically authenticate part or all of their classical communication with an unconditionally secure authentication scheme. As an example Wegman-Carter authentication.^[20]

What's a Brute Force Attack? (source: <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>)

A hacker brute force attack uses trial-and-error to guess login info, encryption keys, or find a hidden web page; they use excessive forceful attempts to try and 'force' their way into private account(s). Hackers work through all possible combinations hoping to guess correctly. Computers (many linked together in a coordinated fashion) are used to facilitate brute force attacks, since a computer can execute trial and error guesses much faster and with less strain than doing it by hand.

Depending on the length and complexity of encrypted data and information such as passwords, cracking it can take anywhere from a few seconds to many years.

What do hackers gain from Brute Force Attacks?

Brute force attackers have to put in an effort to make these schemes pay off. Here's how hackers benefit from brute force attacks:

- Profiting from ads or collecting activity data
- Stealing personal or confidential data and valuables
- Spreading malware to cause disruptions
- Hijacking a computer network system for malicious activity
- Ruining a website's reputation

Profiting from ads or collecting activity data.

Hackers can exploit a website alongside others to earn advertising commissions. Popular ways to do this include:

- Putting spam ads on a well-traveled site to make money each time an ad is clicked or viewed by visitors.
- Rerouting a website's traffic to commissioned ad sites.
- Infecting a site or its visitors with activity-tracking malware — commonly spyware. Data is sold to advertisers without your consent to help them improve their marketing.

Stealing personal data and valuables.

Breaking into online accounts can be like cracking open a bank vault: everything from bank accounts to tax information can be found online. All it takes is the right break-in for a criminal to steal one's identity, money/cryptocurrency, or sell private credentials for profit.

Spreading malware to cause disruptions for the sake of it.

If a hacker wants to cause trouble or practice their skills, they might redirect a website's traffic to malicious sites. Alternatively, they may directly infect a site with concealed malware to be installed on visitor's computers.

Hijacking your system for malicious activity.

When one machine isn't enough, hackers enlist an army of unsuspecting devices called a botnet to speed up their efforts. Malware can infiltrate a computer, mobile device, or online accounts for spam phishing, enhanced brute force attacks and more. If security is not in place such as an antivirus system, an innocent computer site may be more at risk of infection.

Ruining a website's reputation.

If a website becomes a target of vandalism, a cybercriminal might decide to infect the site with obscene content. This might include text, images, and audio of a violent, pornographic, or racially offensive nature.

Types of Brute Force Attacks

Each brute force attack can use different methods to uncover confidential or sensitive data. Popular brute force methods includes:

- **Simple brute force attacks:** hackers attempt to logically guess credentials — completely unassisted from software tools or other means. These can reveal extremely simple passwords and PINs. For example, a password that is set as *"guest12345"*.
- **Dictionary attacks:** in a standard attack, a hacker chooses a target and runs possible passwords against that username. These are known as dictionary attacks. Dictionary attacks are the most basic tool in brute force attacks. While not necessarily being brute force attacks in themselves, these are often used as an important component for password cracking. Some hackers run through unabridged dictionaries and augment words with special characters and numerals or use special dictionaries of words, but this type of sequential attack is cumbersome.
- **Hybrid brute force attacks:** these hackers blend outside means with their logical guesses to attempt a break-in. A hybrid attack usually mixes dictionary and brute force attacks. These attacks are used to figure out combo passwords that mix common words with random characters. A brute force attack example of this nature would include passwords such as *NewYork1993* or *Spike1234*.
- **Reverse brute force attacks:** just as the name implies, a reverse brute force attack reverses the attack strategy by starting with a known password. Then hackers search millions of usernames until they find a match. Many of these criminals start with leaked passwords that are available online from existing data breaches.
- **Credential stuffing:** if a hacker has a username-password combo that works for one website, they'll try it in tons of others as well. Since users have been known to reuse login info across many websites, they are the exclusive targets of an attack like this.
- **SYN (synchronous) flood attack:** A SYN (synchronization) flood (half-open attack) is a type of denial-of-service (DDoS) attack by a hacker which aims to make a network server unavailable to legitimate traffic by consuming all available server resources. By repeatedly sending initial connection request (SYN) packets, the attacker is able to overwhelm all available ports on a

targeted server machine, causing the targeted device to respond to legitimate traffic sluggishly or not at all.

- **How does a SYN flood attack work?**

SYN flood attacks work by exploiting the handshake process of a TCP connection. Under normal conditions, TCP connection exhibits three distinct processes in order to make a connection.

1. First, the client sends a SYN packet to the server in order to initiate the connection.
2. The server then responds to that initial packet with a SYN(send)/ACK(acknowledge) packet, in order to acknowledge the communication.
3. Finally, the client returns an ACK packet to acknowledge the receipt of the packet from the server.

After completing this sequence of packet sending and receiving, the TCP connection is open and able to send and receive data.

To create denial-of-service, an attacker exploits the fact that after an initial SYN packet has been received, the server will respond back with one or more SYN/ACK packets and wait for the final step in the handshake. Here's how it works:

- The attacker sends a high volume of SYN packets to the targeted server, often with spoofed (spoofing is a specific type of cyber-attack in which someone attempts to use a computer, device, or network to trick other computer networks by masquerading as a legitimate entity) IP addresses.
- The server then responds to each one of the connection requests and leaves an open port ready to receive the response.
- While the server waits for the final ACK packet, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet causes the server to temporarily maintain a new open port connection for a certain length of time, and once all the available ports have been utilized the server is unable to function normally.

In networking, when a server is leaving a connection open but the machine on the other side of the connection is not, the connection is considered half-open. In this type of DDoS attack, the targeted server is continuously leaving open connections and waiting for each connection to timeout before the ports become available again. The result is that this type of attack can be considered a "half-open attack".

A SYN flood can occur in three different ways:

- **Direct attack:** A SYN flood where the IP address is not spoofed is known as a direct attack. In this attack, the attacker does not mask their IP address at all. As a result of the attacker using a single source device with a real IP address to create the attack, the attacker is highly vulnerable to discovery and mitigation. In order to create the half-open state on the targeted machine, the hacker prevents their machine from responding to the server's SYN-ACK packets. This is often achieved by firewall rules that stop outgoing packets other than SYN packets or by filtering out any incoming SYN-ACK packets before they reach the malicious user's machine. In practice this method is used rarely (if ever), as mitigation is fairly straightforward – just block the IP address of

each malicious system. If the attacker is using a botnet such as the Mirai botnet they won't care about masking the IP of the infected device.

- **Spoofed Attack:** A malicious user can also spoof the IP address on each SYN packet they send in order to inhibit mitigation efforts and make their identity more difficult to discover. While the packets may be spoofed, those packets can potentially be traced back to their source. It's difficult to do this sort of detective work but it's not impossible, especially if Internet service providers (ISPs) are willing to help.
- **Distributed attack (DDoS):** If an attack is created using a botnet the likelihood of tracking the attack back to its source is low. For an added level of obfuscation, an attacker may have each distributed device also spoof the IP addresses from which it sends packets. If the attacker is using a botnet such as the Mirai botnet, they generally won't care about masking the IP of the infected device.

By using a SYN flood attack, a bad actor can attempt to create denial-of-service in a target device or service with substantially less traffic than other DDoS attacks. Instead of volumetric attacks, which aim to saturate the network infrastructure surrounding the target, SYN attacks only need to be larger than the available backlog in the target's operating system. If the attacker is able to determine the size of the backlog and how long each connection will be left open before timing out, the attacker can target the exact parameters needed to disable the system, thereby reducing the total traffic to the minimum necessary amount to create denial-of-service.

Tools Aid Brute Force Attempts

Guessing a password for a particular user or site can take a long time, so hackers have developed tools to do the job faster.

- **Automated tools help with brute force attacks.** These use rapid-fire guessing that is built to create every possible password and attempt to use them. Brute force hacking software can find a single dictionary word password within one second.
 - Tools like these have workarounds programmed in them to:
 - Work against many computer protocols (like FTP, MySQL, SMTP, and Telnet).
 - Allow hackers to crack wireless modems.
 - Identify weak passwords.
 - Decrypt passwords in encrypted storage.
 - Translate words into geekspeak (an informal language or code used on the internet, in which standard letters are often replaced by numerals or special characters that resemble the letters in appearance) — "don't hackme" becomes "d0n7H4cKm3," or "seeme" becomes "Cm3", as examples.
 - Run all possible combinations of characters.
 - Operate dictionary attacks.
 - **Some tools scan pre-compute rainbow tables** for the inputs and outputs of known hash functions. These "hash functions" are the algorithm-based encryption methods used to translate passwords into long, fixed-length series of letters and numerals. In other words, rainbow tables remove the hardest part of brute force attacking to speed up the process.
 - **GPU Speeds Brute Force Attempts** ...tons of computer brainpower is needed to run brute force password software. Unfortunately, hackers have worked out hardware solutions to make this part of the job a lot easier. GPU speeds up brute force attempts because the term graphics processing unit (GPU) refers to a chip or electronic circuit capable of rendering (generating and displaying) graphics for display on an electronic

device (such as a gaming monitor or conventional computer monitor). GPU processing executes much faster than conventional CPU or central processing unit located on a computer's motherboard.

- **Combining the CPU and graphics processing unit (GPU)** accelerates computing power. By adding the thousands of computing cores in the GPU for processing, this enables the system to handle multiple tasks at once. GPU processing is used for analytics, engineering, and other computing-intensive applications. Hackers using this method can crack passwords about 250 times faster than a CPU alone.
 - So, how long would it take to crack a password? To put it in perspective, a six-character password that includes numbers has approximately 2 billion possible combinations. Cracking it with a powerful CPU that tries 30 passwords per second takes more than two years. Adding a single, powerful GPU card lets the same computer test 7,100 passwords per second and crack the password in 3.5 days.

Security Techniques to Protect Passwords and Cryptography Keys

- **High encryption rates:** to make it harder for brute force attacks to succeed, passwords should be encrypted with the highest encryption rates possible, such as 256-bit encryption. The more bits in the encryption scheme, the harder the password is to crack.
- **Salt the hash:** randomize password hashes by adding a random string of letters and numbers (called salt) to the password itself before it is hashed. This string should be stored in a separate database and retrieved and added to the password before it's hashed. By salting the hash, users with the same password have different hashes.
- **Two-factor authentication (2FA):** use two-step authentication and install an intrusion detection system that detects brute force attacks. This requires users to follow-up a login attempt with a second factor, like a physical USB key or fingerprint biometrics scan.
- **Limit number of login re-tries:** limiting the number of attempts reduces susceptibility to brute-force attacks. For example, allowing three attempts to enter the correct password before locking out the user for several minutes can cause significant delays and cause hackers to move on to easier targets.
- **Account lockdown after excessive login attempts:** if a hacker can endlessly keep retrying passwords even after a temporary lockout, they can return to try again. Locking the account and requiring the user to contact IT for an unlock will deter this activity. Short lockout timers are more convenient for users, but convenience can be a vulnerability. To balance this, consider using the long-term lockdown if there are excessive failed logins after the short one.
- **Throttle rate of repeated logins:** slow an attacker's efforts by creating space between each single login attempt. Once a login fails, a timer can deny login until a short amount of time has passed. This will leave lag-time for real-time monitoring team to spot and work on stopping this threat. Some hackers might stop trying if the wait is not worth it.
- **Required Captcha after repeated login attempts:** manual verification does stop robots from brute-forcing their way into data. Captcha comes in many types, including retyping the text in an image, checking a checkbox, or identifying objects in pictures
- **Use an IP denylist to block known attackers.** Keep this list constantly updated by those who manage it.

“How long would a password take to crack,” test passphrase strength at <https://password.kaspersky.com>.

How are protocols used in cyber attacks?

Just as with any aspect of computing, attackers can exploit the way networking protocols function to compromise or overwhelm systems. Many of these protocols are used in distributed denial-of-service (DDoS) attacks. For example, in a SYN flood attack, an attacker takes advantage of the way the TCP protocol works. They send SYN packets to repeatedly initiate a TCP handshake with a server, until the server is unable to provide service to legitimate users because its resources are tied up by all the phony TCP connections.

PART 2

BLOCKCHAIN CONCEPTS

ALGORITHM (METHOD OF ENCRYPTION)

Algorithm is a procedure or set of instructions for solving a [mathematical] problem in a finite (fixed) number of steps that frequently involves repetition of an operation; a step-by-step method of accomplishing some task; a well-ordered collection of unambiguous and effectively computable operations that, when executed (or implemented), produces a desired concluding result and halts the operation in a finite amount of time (all things are not computable, example – trying to divide by ‘0’ or the last five digits of pi).

BITCOIN

Bitcoin (BTC) is the name of a particular brand of cryptocurrency whose value is recognized by the owners and users of such currency in the Bitcoin Public Blockchain. There are many brands of cryptocurrency, each having their own value compared to the U.S. dollar. (This branding is comparable to values and exchange rates assigned, for example, by owners of U.S. dollars, British Pounds, or Canadian dollars). The chart illustrates the historical value of Bitcoin compared to U.S. dollar up to December 7, 2022. As shown, the value has a rather



volatile history ranging from a low of US\$327 per BTC to a high over US\$60,000 per Bitcoin. On December 7, 2022, Bitcoin was worth US\$16,830 per BTC. Investors in Bitcoin need to have a risk tolerance that can weather wide value fluctuations. Yeah to those who bought in at US\$327 and sold at US\$60,000 per BTC and whoa to those who bought in at US\$60,000 and sold at US\$16,830.

BLOCK

Block stores data and comprised of a list of **transactions** chronologically grouped together and recorded into a chain of Blocks in a Blockchain ledger over a given period of time. Each Block size has a maximum number of bits (0s and 1s) of computer binary code memory, whose size is limited so that the Blockchain transaction verification process for each Block can take place in bite size chunks in a reasonable period of time. (In the Bitcoin Blockchain, Blocks are limited to 1MB in size, 1000 characters of data). A unique identification number or fingerprint or serial number (one of a kind) is determined for each Block by inputting the Block’s transaction data into a cryptographic algorithm than calculates a unique fixed bit length alphanumeric number (composed of letters and numbers, typically 32 characters long) called a ‘ash’. A hash is a one-way cryptographic number which means it cannot be reversed engineer to recreate the data from which it was determined. What the hash is used for is a way to confirm the data is authentic after saved in a Block, is to run the data into the hash making algorithm and confirm the exact same hash is determined. If even there is the slightest change in the data, an added space, a letter capitalized, etc., hashing it will result in a complete new hash number. If that happens it is a test that the data has been tampered with and its authenticity suspect. Blocks associated with a particular ledger are linked together with other Blocks through their hash (a prior Block hash being associated, linked, with the next Block, and so on).

CHAIN

The term 'chain' in chain of blocks or Blockchain, refers to the hash that links one Block to another, mathematically 'chaining' or linking or associating Blocks together into a common ledger. (The hash of Block1 is associated (linked to) with Block2, the hash of Block2 is associated with Block3, and so on, thus the chain).

CONSENSUS PROTOCOL

In the Blockchain world, consensus is the process or mechanism using an algorithm of (1) developing agreement among a group of mistrusting parties and (2) how a blockchain comes to agreement on new data entered into the system. That agreement is normally related to validating or authenticating the truthfulness of a transaction. The mistrusting parties are the full nodes in the network. The full nodes provide a transaction validation service (which may or may not be compensated as a service fee in cryptocurrency), and provide that service when a transaction is proposed and entered into the network, and when a Block of transactions are validated, the Block and associated transaction data are recorded as part of the network ledger. Each Blockchain will have its own consensus mechanism stated in its protocol (operating agreement). The type of consensus used will vary in part depending on the objectives of the relevant Blockchain: trading value (such as cryptocurrency), storing data or securing systems and contracts. The glossary provides an explanation of various types of consensus mechanisms (such as Proof of Work, Proof of Stake).

Consensus is a set of steps taken by most or all of the nodes in a Blockchain to agree to a proposed state or value. The fundamental requirements of consensus algorithm boil down to safety and liveness (operates even with some faults) conditions. Requirements include:

- Agreement: All honest nodes decide on the same value.
- Integrity: No node can make the decision more than once in a single consensus cycle.
- Validity: The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node.
- Fault tolerant: The consensus algorithm should be able to run correctly in the presence of faulty or malicious nodes (Byzantine nodes).
- Termination: All honest nodes terminate the execution of the consensus process and eventually reach a decision.

CRYPTOCURRENCY AND MONEY OR CONVERTIBLE VIRTUAL CURRENCY

(In God we trust, all others pay cash) Currency and money, represented in the form of 'tokens' (be it printed paper or minted metal coins so called 'cash', or substituted tokens such as a paper check or credit card, gold bullion, tokens, etc.) are what users of such currency and money recognize and accept as having value and exchangeable for goods, services and other valued assets. For example, would you rather possess a US Dollar bill or a block of gold? Probably the gold, but if you were freezing, you might consider the paper dollar bill as being more valuable and burn it for heat. A long time ago, before tokens, folks would exchange tangible objects as an exchange of value, such as an animal pelt used as a coat, traded (exchanged) for a basket of berries to fend off hunger. Or a serf in old time England being allowed to farm land owned by a Duke and in exchange for that farming right, give the Duke part of the year's harvest, the serf being allowed to sell the rest for his benefit.

In modern times, it's easier to carry a token dollar bill or 5 pound British note in your wallet that represents value than an animal pelt, or store digital cryptocurrency⁵⁴ in your computer electronic wallet than carrying around a real wallet full of cash that is subject to pick pocket risks.

Cryptocurrency is digital currency that is secured by cryptography, that provides for accountability (such currency is spendable only once and spent by its rightful owner) and anonymity (protect users privacy). In contrast, money is normally thought of as *fiat* money, **a government-issued currency that is not backed by a commodity such as gold and typically managed by a central bank**. Its value is accepted and recognized by users of the currency based on the full faith and credit trust instilled in the issuing government. (For example, I suspect most folks would accept certain *hard currency*⁵⁵ such as the U.S. Dollar or British Pound, which is generally exchanged world-wide for other value, in contrast to not readily accepted less recognized soft currency such as the Albania Lek, which may really only have value if used within the nation of Albania).

Fiat money gives central banks greater control over the economy because they can control how much money is printed. Most modern paper currencies, such as the U.S. dollar, are fiat currencies.

Just like fiat money, cryptocurrency has value because the people who use it, accept and recognize it as having value and readily exchange among themselves for goods and services or other valued assets

In order to prevent others from printing fake fiat money (counterfeiting the coins or printed paper issued by governments money), the physical token (paper or coin) must be designed in such a way to prevent or make it extremely difficult to create fake or counterfeit copies. For all practical purposes this anti-counterfeiting process is another form of 'coding' the tangible money to make it secure and not easily duplicated. Thus, cryptography is in a sense used to make printed fiat money safe and secure to keep it from being counterfeited.

Generally a sovereign nation has laws that only it can print fiat 'money'. So the question arises, is cryptocurrency in violation of a nation's money printing laws? An interesting and complex question.

Some nation's have passed laws prohibiting the use of cryptocurrency, others have adopted it as the nation's official money, other's have left the question unanswered – the grey zone – spend at your own risk.

It is not uncommon for 'buyers' and 'sellers', and traders (not to be confused with traitors), to substitute for money, tradeable things that they associate with having value, for other things. For example, trading an animal pelt coat for a bucket of berries (and not using money), or trading a lawnmower for a food

⁵⁴ Cryptocurrency is a digital currency that is secured by cryptography. Digital means the currency is recorded in electronic computer ledgers (similar to one's online bank account showing a positive balance) and recognized by users of that cryptocurrency as having value, such value typically measured against fiat money such as so many crypto's equal so many U.S. Dollars. It is interesting that cryptocurrency advocates laud the decentralized bank free status of crypto, yet depend on centralized bank controlled fiat money, U.S. dollar for instance, as the measure of the value of cryptocurrency. **Can decentralized crypto exist without centralized fiat money?**

⁵⁵ Money that is readily acceptable in most nations as exchangeable value as opposed to soft currency which is not readily acceptable (try exchanging in your local U.S. bank an Albanian Lek note for a U.S. dollar bill. You'll have more luck if you try and exchange a British Pound).

processor. Trading one form of cryptocurrency for another or for goods and services. In regard to sovereign nation's rules, if a trade results in one of the parties trading for a higher valued asset – the excess value being considered a gain or profit, that higher value is deemed as a taxable 'profit' and a taxable event. Obviously such taxable event concept (think garage sales or flea market or parking lot swap meets⁵⁶) is difficult if not impossible to enforce, but in larger traceable transactions, the income reporting and tax issue may become an enforceable event.

Is cryptocurrency, not 'money'? – since its not minted to mimic a sovereign nation's fiat money – its not counterfeit money. Is cryptocurrency just another tradeable asset viewed as having value by the trading participants, just like an animal pelt or basket of berries, that users recognize as having value?

Some challenges with cryptocurrency, even though viewed as a tradeable currency asset and not fiat money, include...

- How can any cryptocurrency involved 'private' trade – a fundamental objective of Blockchain technology – protecting privacy (trade cryptocurrency for other cryptocurrency or even fiat money or traded for things), be traced and taxed by a relevant sovereign nation, if there is a trade of value that creates a taxable 'profit' to one of the traders who is subject to the taxing authority of the taxing nation?
- How can anti-laundering rules be enforced where bad folk use 'private' cryptocurrency as a mechanism to hide or conceal the illegal transfer of wealth?
- How can sovereign sanction rules be enforced regarding the restriction of a sanctioned party holding value or trading in U.S. linked 'private' cryptocurrency transactions?

CRYPTOANALYSIS

Cryptanalysis is the term used for the study of methods for obtaining the meaning in plaintext of encrypted information **without access to the key**, otherwise normally required to decrypt the message to plaintext; i.e., it is the study of how to "crack" or break encryption ciphers or encrypted information (either for legitimate or malicious purposes). Learning how to crack codes involves first understanding how messages are encrypted. Breaking a code may be as simple as guessing the decryption method (seeing a pattern where someone uses certain words or phrases or characters) or more advanced sophisticated code cracking techniques such as the development of quantum computers (super-super fast computer machines) breaking cryptographic security systems gets easier (especially brute force attackers), thus an on-going battle to develop and upgrade cryptographic security technology that can compete against quantum computing. Quantum algorithms are used with quantum computing and take a new approach to solving complex problems -- creating multidimensional spaces where the patterns linking individual data points emerge. Classical computers cannot create these computational spaces, so they cannot find patterns.

CRYPTOGRAPHY

Cryptography, or cryptology (from Ancient Greek: "hidden, secret"), is the practice and study of techniques for secure communication in the presence of adversarial behavior (keeping things secret from unauthorized entities trying to break-in and unlock the secrecy and divert other's wealth to the adversary

⁵⁶ Rare is it to find a cash register or paper purchase receipt book being used at such meets, which would otherwise record a sales transaction (for income and tax reporting purposes).

benefit). More generally, cryptography is about constructing and analyzing rules (protocols) of communication between parties that prevent unauthorized third parties or the public from reading private messages or gaining access to confidential data and information.

Modern cryptography crosses many boundaries such as disciplines of mathematics, computer science, information security, electrical engineering, digital signal processing, physics, and others. Core concepts related to information security (data confidentiality, data integrity, authentication, and non-repudiation – immutable/can't be modified or changed once published) are also central to cryptography. Practical applications of cryptography include electronic commerce (online banking transactions, trading stock online, chip-based payment cards, digital currencies, computer passwords, and military communications).

Cryptography historically referred almost exclusively to "encryption", which is the process of converting ordinary information (plaintext) into an unintelligible form (ciphertext). Decryption is the reverse, moving from the unintelligible ciphertext back to plaintext. A cipher (or cypher) is a pair of **algorithms** (the calculating 'machines' - computer software that executes mathematical functions - that performs the coding (encryption) and decoding (decryption) calculations. In an elementary sense, cryptography, or secret code messaging, is about coding and scrambling information so that it looks like babble to anyone except those who know the trick to decoding or unlocking it, then the babble is converted to plaintext language, such as English.

In everyday English language, the term "**code**" is often used to mean any method of encryption or concealment of meaning. However, in cryptography, code has a more specific meaning: the replacement of a unit of plaintext (i.e., a readily understood word or phrase) with a code word (for example, the code word "*wallaby*" (a challenge phrase) means as a substitute word and replaces the plaintext meaning "*attack at dawn*" (a response phrase). A "**cypher**", in contrast, is a scheme for changing or converting a message of plaintext that is in plain sight into a meaningless message in order to produce a ciphertext ("*attack at dawn*" might be "*cfgz12 45t lom8s91*").

A "**cryptographic system**" or "**cryptosystem**" for short, means a security system made up of three elements:

1. **The Encryption Engine:** the part of the computer software that starts the encryption process and reverses (decrypts) it with the selected algorithms. Encryption is accomplished with encryption software that becomes functional with the use of private and/or public encryption keys. In using encryption software, a **passphrase** is required to protect encryption keys. A passphrase is a combination of a **password** and **phrase** (consisting of more than one word and /or special characters). The passphrase is the combination lock that protects the encryption keys that make it possible to encrypt data. Strength (how hard it is to crack or disclose a passphrase) is determined by:
 - a. **The length of the passphrase (number of characters, the longer the better);**
 - b. **It 'entropy' (a measure of how random it is, not predictable);**
 - c. **Safety measures used to protect the key from prying eyes and unauthorized use.**

There is a constant battle between supremely secure passphrases and how user friendly they are to remember and use (with the downside if they are not user friendly, users using unsafe shortcuts to use them – such as writing them down, saving in a file, using shorter easier to remember phrases, etc.).

The Keying Information: or keying system is the portion of the computer software that creates and manages the digital keys needed to start the algorithm for it to encrypt plaintext messages to encrypted ciphertext and to start the algorithm for it to decrypt ciphertext messages to decrypted plaintext. Keys are the digital tool(s) required to start the coding and decoding calculating machines (sort of like using a key to start/turn-off a car, or a key to lock and unlock one's front door); and

2. **The Operational Procedures:** how the encryption engine and keying information parts interact with one another and how the output, or result, is formatted and what file extension (if any) is used (in regard to storing data and information, keys and other elements).

DIGITAL

(An adjective) usually refers to something using discrete (fixed) numbers (like 0 or 1 digits and nothing in between those numbers such as 0.5, an on/off toggle switch or true/false logic); or it can also refer to electronic form of value – digital coins or Bitcoins – a form of currency that has value recognized by its users but is not minted or printed (though Bitcoin owners can own actual metal Bitcoin tokens like a bus token) identified in electronic computer ledgers (viewed on one's computer or phone screen).

DISTRIBUTED DATABASE

There is no central server or system which keeps the data of Blockchain. The data is distributed over millions of computers around the world which are connected with the Blockchain. This system allows notarization (fraud-deterrent process that assures the parties of a transaction that a document is authentic, and can be trusted) of data as it is present on every node and is publicly verifiable.

ENTROPY

Entropy is the probabilistic measure of randomness, disorderliness, or uncertainty. Probabilistic means the measure of the likelihood that a system state will exist (the likelihood of something happening, for example a 50% chance of such and such happening). In physics, the natural higher probabilistic (most likely) entropy state of existence, is for systems not to seek order and certainty but to seek randomness, disorder or uncertainty (in the extreme, termed *chaos*, which is a system state described as not being predictable. Think after taking a hot shower, the condensed water vapor fog in the shower room, is most likely spread out randomly and evenly throughout the room – its high probable state of entropic disorderliness - and not clumped together in a ball of fog, an unlikely – low entropic probability state - small ordered confined space in one of the corners of the room near the ceiling). If 100 pennies are tossed in the air and when they land on the ground (because of gravity), the likelihood of landed pennies being all heads or all tails, a very orderly state of existence (or low entropy, low probability event), is highly unlikely. The most likely 'disordered' arrangement is close to 50% tails and 50% heads the higher entropy measurement (since either are equally likely to happen since those are the only two events that can occur - assuming a penny does not miraculously land on its edge and stay in that extremely unlikely vertical position : /), in contrast there is the opposite low probability, low entropy (lack of disorder, low probability) that all tosses will end up tails or all tosses will end up heads.

There are three avenues to existence or being: **deterministic, probabilistic or chaotic.**

Deterministic means the state of a system that has an observable cause and effect result, a predictable and determined outcome (a yes or no answer). Input certain information into a circumstance, and the same output will be determined every time. For example, heating ice at the earth's surface at one atmospheric pressure to 33 degrees Fahrenheit, the ice will always melt to liquid water. No matter how many times you do the experiment or where, the same result will occur. Thus a cause (heating the ice) and effect (it will melt to liquid water) or deterministic predictable result.

Probabilistic⁵⁷ means the state of a system that is not governed by cause and effect deterministic predictable (yes or no answer) results, but a system whose outcome (a maybe answer) will vary over a range of possible probabilistic or statistical results, such as a 50% chance of an event occurring. For example, toss a penny in the air and the probability of the penny landing heads up is 50%, since there are only two possibilities, heads or tails. Toss the penny a million times, and it is probable or more likely than not, that half of the tosses will be heads and half will be tails. (For the avoidance of doubt, a million toss of the penny does not mean they will never land all heads or tails, it's just that event is highly unlikely – and, btw, I'm willing to bet my pension plan on that result not happening).

Chaos means the state of a system being neither deterministic (yes or no) or probabilistic (maybe), but wholly unpredictable (only the shadow knows). Cannot assign a cause and effect outcome or probabilistic outcome to a chaos event. Input certain information into a system, and the outcome is unpredictable (neither yes or no or maybe, pure randomness what happens – your guess is as good as mine). Entropy measures how chaotic an event may be. High entropy means highly chaotic or highly unpredictable, low entropy means less chaotic and possibly predictable (approaching deterministic or probabilistic outcomes). As an example, cosmologists believe the Universe is expanding to a state of chaos (high entropy), or total disorderliness. One might ask if that is the case then why do highly 'organized' non-chaotic galaxies exist in the Universe? The answer is that they exist because such events are temporary disruptions in chaos because of local energy interferences causing local conditions of orderliness (stars being formed and deformed), but the overall Universe taken as a whole is seeking chaos or unpredictable, disorder behaviour, and hence the Universe entropy is increasing, as is the Universe is expanding (so I've been informed by a bunch of smart published cosmologist and physicists).

WHY THIS DIVERSION INTO ENTROPY AND CHAOS?

Because the concepts of entropy, or chaos, are essential ingredients used in cryptography to develop security methods and keys that are extremely difficult for attackers to attack, duplicate or break. The more chaotic, unpredictable (high entropy), and long length of a security key, the less likely an attacker (think hacker or cyberterrorist) will be able to duplicate the key or break into secure networks and disrupt them or cause harm. Cryptography is all about hiding something in plain sight and to do that, entropic

⁵⁷ As an aside, most civil lawsuit cases are based on a blend of deterministic cause and effect and probabilistic claims. For example, the injuries sustained by a plaintiff in a (car accident) personal injury (the effect) lawsuit, will be argued to have been proximately caused by the negligent conduct of the defendant driver in the other car who hit the plaintiff. The damages incurred by a plaintiff in a breach of contract lawsuit (the effect) will be argued to have been proximately caused by the breaching conduct of the defendant. The trier of fact (the judge or jury) will need to decide, by a preponderance of the evidence, if the evidence supports the claim. Preponderance of the evidence means to prove that something is more likely than not (a probabilistic standard). Oftentimes, expert witnesses give testimony of their professional opinion why something is more likely than not to have occurred. The trier of fact does not have to believe or accept such testimony as being the truth or relevant, since such testimony only adds weight supporting or refuting the overall claim. Sometimes these experts use complex statistical or probabilistic analysis why it is claimed that something is more likely than not to have occurred. Justice sometimes is not nice.

and chaos principles (and as later discussed, in the form of complex mathematical processes) are needed in the security recipe.

In computer speak, the **difficulty** in duplicating and discovering a security key or cracking a cipher code – a successful attack by a hacker or cyberterrorist - is called its '**hardness**'. A security system that is hard is difficult to breach. A soft security system, is easy to breach. The hardness measurement of a password defined as "PASSWORD" would be easy to guess by an attacker and thus not hard, yet one made up of many 'random' characters ("aB23\$*zE&+@!nNzp45h") would more than likely not be easy to break and thus considered hard. As later discussed, seeking randomness (like seeking the fountain of youth) is not an easy thing to do, and expert cryptologists spend a lot of time developing and experimenting with procedures (mainly mathematical) that are desired to be uniquely random when designing security cryptographic systems, the objective of which is developing cryptographic techniques that are extremely difficult (and ideally impossible) for an attacker to successfully break.

Hardness should not be confused with material hardness tests such as the Mohs Hardness Scale, Brinell Hardness Test, Rockwell Hardness Test, Vickers Hardness Test or Knoop Hardness Test, where hardness of material is defined as *the resistance of a specific material to localized plastic deformation or indentation*, or in plaintext English, how easy it is to dent a material sample by banging it with a hammer. Of course one can also be accused of being hard headed which has more to do with not being a cooperative fello or fella, vs the toughness of one's noggin'.

There are many techniques used by attackers to try and break cryptographic security systems. Computers are used to try and attack security systems protected by cryptography. In isolated circumstances the old James Bond techniques of getting secure information can be accomplished with the threat of a weapon, fist or maybe even eavesdropping such as wiretapping, but computer literate hackers and attackers are more stealth full and savvy and prefer to do their clandestine work in the comfort of their man/women cave dens sitting in front of a computer, wearing a hoodie/jogger slacks and tennies, than risk physical confrontation.

When scientists and engineers encounter difficult problems and can't solve a problem using 'normal' computers, they turn to supercomputers. These are very large classical computers, often with thousands of classical CPU (Central Processing Unit is electronic machinery that carries out instructions from computer programs that allows a computer or other device to perform its tasks) and GPU (Graphics Processing Unit, a specialized processor originally designed to accelerate graphics rendering, can process many pieces of data simultaneously, making them useful for machine learning, graphing, video editing, and gaming applications) cores. However, even supercomputers struggle to solve certain kinds of problems.

FORKING

First some background. Blockchain technology has been touted to be secure, immutable (extremely difficult to change) and non-repudiation (each transaction stored in a Blockchain is created by an entity that authorizes and signs the transaction. It is not permitted to remove or change a transaction once it has been added to the Blockchain due to the immutability of Blockchain (its recorded 'permanent' chronological history of transactions). Due to this immutability, the entity that signed and authorized the transaction cannot later deny (repudiate) the existence of the transaction).

With these stringent Blockchain conditions in mind, immutable and non-repudiation, can a Blockchain network be changed or modified because of extreme or unexpected events? (such as a hacker successfully attacking a system, or a major fault bug is found in programming language, or other unusual event).

Answer: Yes, by forking. Here's how...

In Blockchain, a fork⁵⁸ (which means a circumstance in a Blockchain network that can lead to two different paths to follow when only one is permitted), is defined variously as:

- "what happens when a blockchain diverges into two potential paths forward"
- "a change in protocol" (the operating agreement is changed), or
- a situation that "occurs when two or more blocks are validated at the same time, or
- "a hacker successfully attacks the Blockchain network and without authorization, steals cryptocurrency from user accounts."

Forks are related to the fact that parties (Users, nodes, Developers) in a Blockchain network need to use common rules to maintain the (immutable, non-repudiated) history of the Blockchain. When parties are not in agreement, alternative chains of the same Blocks might emerge resulting in different histories of the same transactions in two different chain of blocks. While most forks are short-lived (and any transaction history differences promptly resolved back into one) some are permanent.

Managing forks is not easy since the users of the network have computer programs that only work on one software protocol. Can't mix old software with new software.

Forks can be classified as accidental or intentional.

Accidental fork happens when two or more full node Block transaction validators (miners) find a Block at nearly the same time. The temporary fork (two separate but identical potential valid Block additions to the Blockchain) is resolved when subsequent Block(s) are added and one of the chains becomes longer than the alternative(s). The network abandons the Blocks that are not in the longest chain (they are called orphaned blocks).

Intentional forks that modify the system operating rules (protocol is changed) of a Blockchain can be classified as a hard fork or a soft fork.

Forking means in very rare and certain special circumstances, information in the Blockchain can be changed – which is contrary to the normal immutable (can't change) and non-repudiation (can't deny) rules. Changing information or protocol (operating procedure) in the Blockchain network is known as 'forking' – similar to what happens when one comes to a fork in the road, they can either go left or go right. Some forks are planned, others- results of extreme situations such as malfunctioning bugs in the operating software or unauthorized hacks or attacking by third party intruders.

There are two types of forks:

- (1) a **soft fork** is a change to the Blockchain network operating or protocol procedures that allows new operating rules to be introduced without requiring all users to upgrade their software. In a soft fork, a majority of the network's users (information authenticators called miners) who

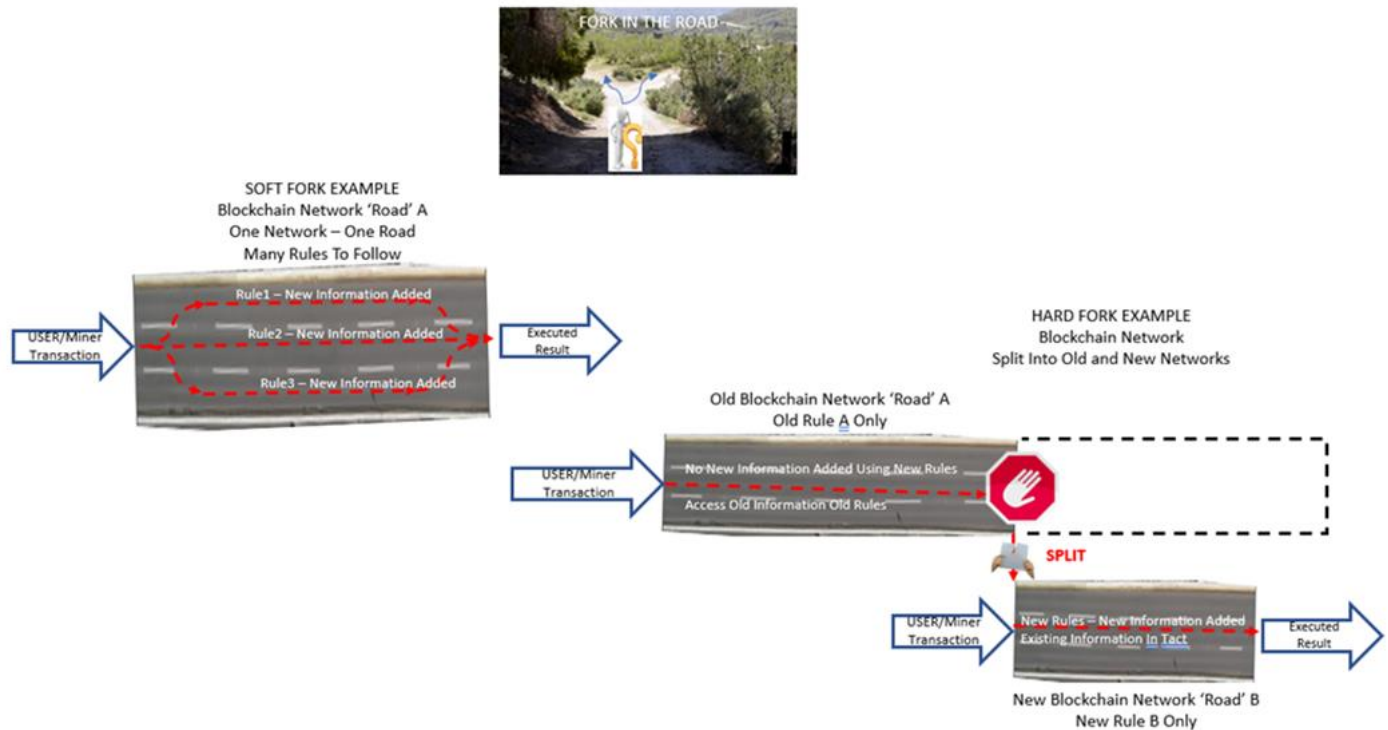
⁵⁸ Recall Yogi Berra's quip: if you come to a fork in the road, take it.

are authorized to authenticate and record new blocks of information in the Blockchain network, implement the new rules and begin following the updated version of the Blockchain. The rest of the network can continue to follow the old version of the Blockchain, but they will be unable to authenticate and add new blocks that follow the updated rules.

Because a soft fork is *backward-compatible* (which means there is only one Blockchain network but that network can operate using separate set of operating rules – thus backward compatibility allowing a user to switch back and forth between two operating rules), it does not result in the creation of a separate and new Blockchain or the splitting of the network into two different Blockchain networks. Instead, it allows the old network rule followers to gradually transition to the new rules, while still maintaining compatibility with the old rules. Examples of soft fork changes to a Blockchain protocol or operating rules include: (1) changing the size (in bits of data) allowed to be saved in bite-size blocks of information recorded in the Blockchain network – a maintenance change, or (2) where there is a temporary duplication of the same information to be recorded in the Blockchain which can happen if two or more authenticators of such information, authenticate at the same time – a natural self-healing operating soft-fork.

- (2) A **hard fork** is a fundamental change to the Blockchain network protocol or operating procedure that is not backward-compatible and requires all users to upgrade their software in order to continue participating in the network, rendering the older versions of that protocol invalid. If there are users desiring to continue to use the old version of the network, then the network splits into two separate Blockchain network versions: one that follows the new rules and one that follows the old rules – the two are not compatible not allowing a user to switch back and forth between them – not backward-compatible. In the extreme case, the old Blockchain network would be shutdown by its Developers if there is no useful or beneficial purpose for keeping it active.

The below diagram illustrates forking events.



HACKER

A hacker is a person skilled in information technology who uses their technical knowledge to achieve a goal or overcome an obstacle, within a computerized system by non-standard means. Though the term hacker has become associated in popular culture with a security hacker – someone who utilizes their technical know-how of bugs or exploits to break into computer systems and access data which would otherwise be inaccessible to them – hacking can also be utilized by legitimate figures in legal situations. For example, law enforcement agencies sometimes use hacking techniques in order to collect evidence on criminals and other malicious actors.

HARDNESS

Hardness is a measure of the difficulty of an adversary breaking an encryption code.

PASSWORDS

We will embark on a discussion of password security to further our understanding of cryptography and its use in Blockchain technology, since most folks are familiar with the use of such security tools, and expand on that knowledge to broader cryptographic devices used in the Blockchain technology.

Passwords have been used since ancient times. Roman sentries would challenge those wishing to enter an area to supply a secret password or *watchword*, and would only allow a person or group to pass if they knew and disclosed the secret password. Similarly (not much has changed in modern times), a computer user uses secret passwords for many purposes: logging into accounts, retrieving e-mail, accessing applications, databases, networks, web sites.

Passwords, are low level security devices – a secret special *string* or combination of characters (composed of letters such as ‘A’, numbers such as ‘9’, and/or characters such as ‘#’, and the combination of characters linked or stringed together, defined as an *alphanumeric* password). Example alphanumeric passwords could be: “PASSWORD” or “Abc123#%*”. When a password is entered into a computer system requiring a password, normally by typing its characters on a keyboard and entered into a login computer monitor screen designated for sign-in, the computer system will compare the secrete password against its database and if the password matches, entry into the system is allowed, and if not, denied.

The easier a password is for the owner to remember generally means it will be easier for an attacker to guess what the password is. Passwords that are difficult to remember can, unfortunately, also reduce the security of a system because ...

- (a) users might write down or electronically store the password thus opening up a potential avenue for attackers to infiltrate and discover the password,
- (b) users may need frequent password resets (an inconvenience factor that often times forces users to revert to easier passwords),
- (c) users being more likely to re-use the same password across different accounts,
- (d) similarly, the more stringent or complex the password requirements, such as "have a mix of uppercase and lowercase letters and digits and characters" or "change it monthly", the greater the degree to which users will subvert the system, because the nuisance of security is viewed as wasting time and getting in the way of what the user is trying to accomplish.

Security experts argue longer, complex passwords with a wide variety of characters provide more security (this higher security described as its “*entropy*”), than shorter/less complex passwords.

PRIMITIVE

Cryptographic ‘primitive’ means a low-level (less complex) cryptographic algorithm (a mathematical function used to encrypt plaintext information to unintelligible babble or gibberish) used as a basic building block for higher-level (more complex) cryptographic algorithms. By linking many simple or primitive cryptographic algorithms together and build on each other, a more secure cryptographic encryption process is developed.

PROVING THE NEGATIVE

Most database and file systems, when searched, can only tell you what data they have within them. In contrast, Blockchain immutable history Block recordation system allows you to see the age of a file, the complete history of a file, and what it looked like over time as it evolved. You can see if there is anything missing from a file. This concept of searching for or seeing things that are missing is called proving the negative, which is absent with conventional file and database system search process.

SCALABILITY

Scalability deals with the development of systems and computer software used in Blockchain networks, to efficiently and quickly process and handle large numbers of transactions

SMART CONTRACT

Unlike ordinary contracts that are usually written and legally enforceable agreements (by a court if required) among two or more parties and stored in a file cabinet or electronically in pdf form, smart contracts are computer programs with predefined terms deployed across a decentralized network, a form of a computer coded documented agreement between two parties, stored digitally on a Blockchain network. Smart contracts are created to automatically facilitate, verify, or enforce the pre-negotiated terms between two or more parties. These smart contract computer programs are triggered and automatically run and enforced when both parties meet agreed-upon terms and conditions. This automated enforcement, without the need of a central party enforcer (such as a court), ensures the correct execution of a smart contract in real time. Unlike conventional contracts that are enforceable in a court of law, smart contracts are enforced by their code – code is law - Blockchain network – an automated enforcement process, unless a special law is passed that treats smart contracts as conventional contracts and enforceable in a court of law. Comparing smart contracts to parking meters: insert the correct payment into the meter and receive the parking validation corresponding to that payment. There is no need for third-party involvement like a cashier because the parking agreement is fixed and automated. Other example smart contract applications include: (1) "smart bonds" that use the Bitcoin Blockchain in which payment streams (purchase or sale of bonds or payment interest) could hypothetically be fully automated, creating a self-paying instrument; (2) "smart inheritance" inheritance wishes could hypothetically be implemented automatically upon registration of a death certificate; (3) "smart birth" social security numbers could be automatically issued upon registration of a birth certificate; "smart deeds" title records and public registers could be automatically updated when a real estate transaction is registered; (4) "smart employment contracts" recording of temporary employee information in tax or human resource accounting records could occur when a temporary employment contract is registered; (5) "smart villages" checklists of planning permission protocols, green belt restrictions and other regulatory compliance requisites could be automatically published upon the registering of a proposed smart village project (building a road, etc.).

Since smart contracts have the inherent limitation of being unable to access any external data, special external interface protocols are required from outside connections, called 'oracles' who provide the necessary external input to the smart contract. (Examples include trip cancellation insurance implementation and coverage executed if input is made the trip was cancelled per the terms of the policy).

TRANSACTION

At its lowest level, a transaction, from the viewpoint of the Blockchain computer network, is just the recording of data/information into the Blockchain ledger. At a higher practical level from the viewpoint of us humans, a transaction is seen as assigning a value to that recorded data which is used to interpret what that data means (for example, in a financial transaction, Party A transfers \$1 to Party B).

PART 3

BLOCKCHAIN LEGISLATIVE AND REGULATORY MATTERS⁵⁹

Introduction

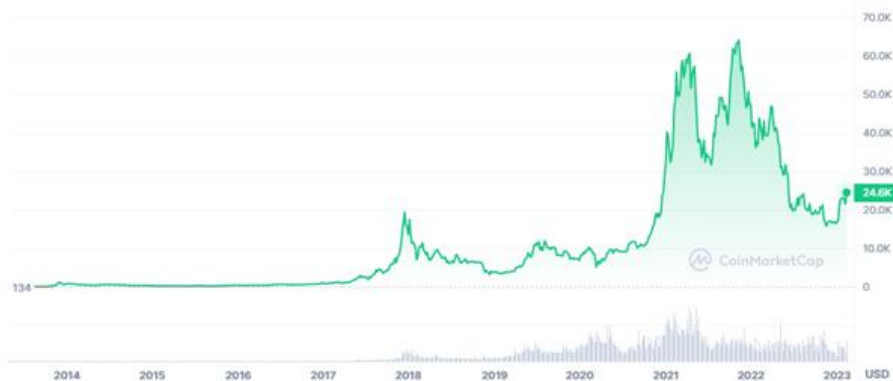
BIG MONEY MEANS A BIG DEAL!

“BUY NOW! THIS CRYPTOCURRENCY COULD BE WORTH 100X!!!; Buy-Low, Sale-High; Get Rich Quick!!! – I’m in...”



On February 16, 2023, the cryptocurrency market was worth over **\$1.1 Trillion dollars**, a 10 year achievement, with over **8,700 cryptocurrency websites**, and more added (and many failures) on a monthly basis – much thanks to the U.S. Jumpstart Our Business Startups (JOBS) Act, that provides for many S.E.C. regulatory registration and security oversight exemptions associated with cryptocurrency Crowdfunding projects! Contrast that to the World’s stock market worth over \$94Trillion (>44,000 listed companies).

On that date, **Bitcoin** was priced at over **\$24,000 per coin**, with a **10 year low of \$119 and high of over \$64,000**. Bitcoin, as most crypto, value has not experienced a typical steady-freddy corporate stock like growth but a volatile lifestyle, with over 2/3rd of its value lost in a one year period.



⁵⁹ Two excellent resources regarding Blockchain regulatory and legislative issues include: *Digital and Digitized Assets: Federal and State Jurisdictional Issues* Prepared By: American Bar Association, Derivatives and Futures Law Committee, Innovative Digital Products and Processes Subcommittee, Jurisdiction Working Group, December 2020, <https://www.americanbar.org>; and. *MoCo Cryptocurrency Litigation Tracker*, MorrisonCohen LLP Law Firm, <https://www.morrisoncohen.com>.

On February 16, 2023 over **\$70 Billion of crypto value was traded** (buying and selling) in a 24 hour period. On the same day corporate stock trades were valued at a little over \$34 billion.

BIG MONEY MEANS A BIG DEAL AND A LOT OF FOLKS DEFENDING THE CRYPTO TREASURY.

Proponents of Blockchain, especially convertible virtual currency, advocate that because of the nature of computer coded network, conventional laws and regulations do not apply or at least are so disconnected, should not apply. That message is accompanied by an express if not implied message that the separatist decentralized structure should remain unregulated and let free market lassie faire principles flourish as they should in a democratic society. That sentiment reminds one of the wild-west, and the old time shoot 'em up cowboy and western movies in which in not one of those movies, accompanied by the compulsory gun slinging town, was there ever the posting of a **STOP SIGN**. Folks knew how to honour



right-of-way yielding or get hurt common sense approach to life, especially for the local bullies who always had the right-of-way. Soon with the advent of the tin lizzie automobile it became mandatory for society to install regulatory stop signs as a logical social thing to do and the populous did not balk at the common sense approach to life and acceptable restriction on their independent life styles. So long as folks followed society's rules and not misuse or abuse stuff...folks were generally happy, or at least safe.

Since human's developed a sense of greed and desire, they have not always acted honorably with their fellow human's. The range of desire and greed have ranged from misuse of stuff to abuse of stuff, that has had adverse impacts on the rights and property of others. There have been bad guys who have tried to disguise themselves as honest people, whether by intent or innocent ignorance. Think of the wild-west days of business monopolies or security fraud misinformation (fake news if you will) that was designed to catapult the wealth status of a few chosen promoters to stratospheric heights, while many other innocent victims that funded that catapult, degraded to pain and despair. Take a lot from everybody is not the only commercial domain of the entertainment industry, be it on the silver screen or the artificial turf of athletic contests or crypto investment websites.

Is unregulated cryptocurrency a wild-west adventure in need of a regulatory stop sign or at least a yield and blinking yellow warning sign, of some sort?

When society is first stepped on and can't defend itself, the governing bodies, in a civilized society, will take the next step, in which relevant laws and regulations are passed to make things right – right being the protection of innocent folks and their property. (One could debate what 'right', 'civilized', 'protection', and 'innocent' means, but lets leave that to common sense and society's rule No. 2, "do unto others...").

Democratically determined laws and regulations are fundamentally invoked and enforced to protect the personal and property rights of society's innocent members, and a civilized mechanism of redress if there has been a breach of those rights – that's justice. Consequently, this principle should/would apply to any

system, including Blockchain and cryptocurrency, and the debate then centers around whether or not an applicable misuse or abuse breach has occurred and not whether a system should be exempt from justice. There is, admittedly, also a countervailing balance not to invoke suppressing laws and regulations that results in unreasonable restrictions on innovation and economic growth risk taking. Need to put the bubble in the middle for a balanced result.

Any system can be used as intended or misused or abused. (Society Rule Number 1, the 2nd is “due unto others...”).

Even in Blockchain, especially cryptocurrency, misuse or abuse events could occur either originating from:

- (i) within the network (there is no ‘perfect’ computer code and always a risk of programming failure, or mischievous manipulations within the system – forking or the 51% Consensus Protocol take over risk, or misuse/abuse of convertible virtual currency transactions; -Anti-Money Laundering - AML/anti-terrorist/Know Your Customer/KYC issues; there is a reason for the constant updates we all get for our mobile phones and computers), or
- (ii) attacks on the network from without (a hacker attack).

These misuse and abuse events may well cause harm to innocent Blockchain participants, and since Blockchain is a uniquely structured separatists, decentralized trustless network, there may not be easily recognizable accountable parties if something goes wrong. An immutable Caveat Emptor, buyer or participant beware, standard of assumed liability does not sound like justice.

Misuse and abuse conduct, whether intentional or unintentional, that adversely affects innocent parties, is a significant reason why there are laws and regulations to protect the innocent – that justice thing again.

Legislation and regulatory matters that affect the Blockchain industry are best categorized by first assessing two fundamental Blockchain structures:

1. Blockchain Technology associated with convertible virtual currencies⁶⁰; or
2. Blockchain Technology associated without convertible virtual currencies.

Why those differences? Because Blockchain technology not associated with convertible virtual currencies and otherwise used as an efficient data base management and contract implementation tool, involves non-financial related oversight laws and regulations, if any.

In contrast, Blockchain technology associated with convertible virtual currencies vastly expands the potential regulatory and legislative landscape since it also includes the trading of digital valued assets which naturally invokes financial regulatory concerns.

In addition to Satoshi’s Public Blockchain separatist disruptive incentive to eliminate the central authority high cost bank middleman and conduct currency value trading business directly between transacting

⁶⁰ Two prominent convertible virtual currency or cryptocurrency Public Blockchain systems are Bitcoin and Ethereum. One main difference between the two is that Bitcoin’s source code only allows for cryptocurrency transactions; Ethereum open source code allows developers to write different applications riding on the Ethereum network platform that has nothing to do with cryptocurrency – a logistic supply management application for example, or issuance of independent cryptocurrency (“native”) coins by any one, new minters.

parties (the peer-to-peer thing), anytime money is involved -especially big money, society's behaviour can change from good to bad to ugly (greed -all for one and one for me - has an effect on the human psyche – non-human wild animals instinctively know when to stop eating when they get full). Greed is a 'great' motivator. Since Public Blockchain gave birth to the innovation of convertible virtual currency – another form of currency and trading value through a wallet (leather or electronic kind), measured in the trillions, the greed motivation factor continues to escalate exponentially. And that greed factor can and naturally result in misuse and abuse behaviour of participating parties, and hence need for justice.

The greed factor is even more pronounced since the value of convertible virtual currency is only recognized by the parties participating in the system (as is true of any currency value looked upon as storing wealth). There is no underlying asset (goods or services – brick and mortar) that can be looked to why such currency has value – truly a way to pull one's self up by their own boot straps. Convertible virtual currency startup enterprises (occurring more frequent than receipt of merchandise catalogs appearing before Christmas holidays), whether legit or maybe no so much, have made an industry out of creating phantom convertible virtual currency and associated value (not too different than the class action law suit industry where huge awarded lawyer fees are expanding faster than crypto). Not that's a bad thing...after all, attend an auction and folks will bid an enormous amount of cash to claim the prized ownership of some used athletic artifact (a dirty baseball, a pair of used shoes, a torn jersey) the intrinsic 'worth' of which is measured in tens of dollars, but society's recognized 'value' measured in hundreds of thousands if not millions of dollars. Value is like beauty, its determined in the eyes of the beholder.

The wild volatile swings in values and risk of convertible virtual currencies is of concern with many regulators as they and society **generally** look for order and not chaos (and that's why when we look into the clouds above we instantaneously make out animals or ships or buildings – cause our minds like order not chaos) unless of course its crypto - that form of volatility is ok for some investors – and essential for its promoters - cause it might mean winning the lottery – and especially those who were early entrants and obtained cheap tokens now worth a King's ransom. Volatility is what market makers look for. Huge swings in value to time buying and selling events, and take advantage of those with less analytic skills – generally the investing public – to unfairly achieve an advantage over others. Justice generally likes fairness and openness and honesty, at least in a civilized society.

Funding and Regulations: Cryptocurrency Blockchain startups (and they tend to happen quite frequently as do many fail quite frequently – can be often motivated for get rich quick schemes) require seed capital to fund some alleged project development (which project may not be more obvious than creating a phantom wealth generating machine – at least for some). Typical capital raising schemes include: Crowdfunding (Jumpstart Our Business Startup Act), Initial Coin Offering (ICO), Initial Exchange Offering (IEO), Private Security Offering (PSO), Security Token Offering (STO) and Digital Security Offering (DSO), and Simple Agreement Future Equities (SAFE). Capital raising schemes can run the gambit from legit to scams to well intentioned but not a clue what is going on. Because of this range of risk, regulatory oversight is often need for justice reasons.

The advantages of tokenizing securities in a Blockchain include efficiencies such as tracking dividend payments; smart contract voting management simplified; liquid market for tokens; facilitate peer-to-peer trading and less transaction fees vs a traditional IPO.

Below are short descriptions of many fund raising schemes used by Blockchain developers.

Blockchain Currency The crypto separatist decentralization revolution, in addition to challenging incumbent centralized business models, has dramatically recalibrated the fundraising incentives for entrepreneurs. Instead of housing product development, strategy, decision-making, and — most importantly for investors — profits, under one corporate umbrella, new Blockchain-based projects seek to democratize returns by providing economic incentives to those that facilitate the network rather than those who own it. As a result, distributed networks of workers and adopters earn financial rewards in the network’s “token” based on their participation rather than centralized for-profit structures and passive shareholders. Entrepreneurs creating separatists decentralized networks no longer need to focus on retaining shares while raising capital since such fundraising models eliminate traditional concepts of equity. Likewise, if a founder maintains a sizable portion of the developed protocol’s tokens (antithetical to the democratized Blockchain value proposition), it may stymie user adoption by diminishing the economic incentives for new entrants if the platform remains too centralized. Thus, developers can solicit funding to build platforms, attract new users, and grow networks without surrendering value in a tokenized economy.

Digital assets (convertible virtual currency) are broadly defined as any digital representation of value which is recorded on a cryptographically secured decentralized ledger (Blockchain) or any similar technology (so says the Secretary of the Treasury). Digital assets include (but are not limited to): Convertible virtual currency and cryptocurrency, Stablecoins, Non-fungible tokens⁶¹ (NFTs). Digital assets are not ‘real’ currency (also known as “fiat” money) because they are not the issued coin and paper money of the United States or a foreign country and are not digitally managed by a government’s central bank. A digital asset (cryptocurrency) that has an equivalent value in real currency (US dollars for example), or acts as a substitute for real currency, functions as a medium of exchange, a unit of account and/or store of value, has been referred to as convertible virtual currency. A cryptocurrency is an example of a convertible virtual currency that can be used as payment for goods and services, digitally traded between users, and exchanged for or into real currencies or digital assets and as an investment vehicle hoping to buy low sale high.

Like traditional companies, Blockchain-based entrepreneurs can seek funding at any stage of the network’s development, but will often attract an investment as soon as possible through a pre-sale of

⁶¹ Non-fungible tokens (no other look a likes in contrast to a dollar bill, one looks and trades just like another which what makes it fungible) are unique one of a kind alpha-numeric computer code (generated off a Blockchain) and used to identify and demonstrate ownership in an asset. That asset could be a digital painting or image, picture of a peanut, portions of a section of land (lake side, road side, wooded side, etc) or any other unique asset. Could even be the confirmation of the age of a fine wine – its provenance. (Reminds me of Enron days who could make a market out of most anything it wanted and folks would buy it). Folks have odd ways of creating and recognizing value (as wide and as odd as one’s imagination), and even odder that the investing public participates and churns the value making environment. Probably no worse than trading ancient comic books for big bucks or similar big buck investments in a celebrity athlete used pair of tennies. NFTs are treated like convertible virtual currency and can be sold and traded. The value of an NFT is as earlier noted, like beauty, value is in the eye of the beholder. It can be whatever or however the market and investors recognize value – could be worthless, could be millions of dollars. NFTs are traded similarly as other cryptocurrency. So if you want to own a piece of the rock...literally...and investors want to recognize big buck value in that slice of silica, then get on the band wagon and hope your NFT value continues to aspire (or spiral) to new heights (or new lows). Buy low sale high seems to be the same ole governing investment strategy and hope your not the last sucker investor holding the empty bag.

the platform's tokens called an "Initial Coin Offering," or ICO. An ICO often occurs after a developer team publishes a white paper, which describes the underlying technology that will support the network (i.e., the nuts and bolts of the Blockchain technology and how it will manage the issuance and trading of a cryptocurrency), the team developing the protocol, and the platform's value proposition (i.e., its use case; when the network is fully established, how it will improve on incumbents).

In certain Blockchains, cryptocurrency can be moved from one Blockchain to another and back again, referred to as "Sidechains". Some schemes send cryptocurrency to an address (wallet) and the coin becomes un-spendable and irrevocable, what is called "Burnt" or "burning the coins". Such burning may be motivated because a new alternate currency (referred to as "altcoin") as proof of an adequate stake in such new coin, used to bootstrap a new currency or introduce scarcity in a cryptocurrency, which results in increased value of the coins. This process, while may be initiated for useful purposes, can be abused or misused by those burning the coins, and potentially selfish reasons for artificially manipulating the value of the crypto.

Tokenization is the process of representing an asset digitally on a Blockchain. It can be used to represent commodities, real estate, ownership of art, currency or anything else of value. Tokens even represent fractional ownership in an asset. When cryptocurrency tokens are used to represent other assets, it is relevant that the tokens are regulated so that investors can have the same level of confidence that they have when they invest using traditional financing methods and institutions. Who is responsible if something goes wrong? And instead of well-known and researched traditional financial fraud methods, malicious parties may choose to launch a technically sophisticated attack. For an average user or investor, this type of attack is difficult to spot and understand as they are entirely on a Blockchain and digitized. New forms of front running and market skewing on decentralized finance platforms is increasingly a concern.

Stable coins, a form of tokenization, are cryptocurrency whose value is pegged against fiat money or precious metals. Price stability, or collateralization, is maintained by backing the token up by a stable asset. Other forms of stabilization include using algorithms that keep track of market supply and demand (using a published index or basket of assets) to maintain stability. While risk management is a good thing to practice, there often is a fine line in regard to whether or not there is an interference with free market ingredients that can artificially influence prices, for good, bad or ugly motives. Not unlike oil syndicates citing they manage oil supply and not oil price...but its sure tough to recognize the difference in practice. Price fixing risks tend to raise a lot of alarm bells in a lot of regulatory arenas.

Crowdfunding is the use of small amounts of capital from a large number of individuals to finance a new business venture normally associated with one product or service, a project based venture. Crowdfunding makes use of the easy accessibility of vast networks of people through social media and crowdfunding websites to bring investors and entrepreneurs together, with the potential to increase entrepreneurship by expanding the pool of investors beyond the traditional circle of owners, relatives, and venture capitalists. Crowdfunding is, without application of an exemption, regulated by the Securities and Exchange Commission in the United States.

The Jumpstart Our Business Startups (JOBS) Act, signed into law by President Obama in April 2012, provides entrepreneurs greater access to capital from both "accredited" (typically high income/sophisticated investor) and "unaccredited" investors by relaxing previous securities exemption

requirements and fostering new legal funding avenues. By combining the Securities Act Section 4(a)(2) private placement exemption, the JOBS Act 506(c) moderation of advertising restrictions, and Section 4(a)(7) and SEC Rule 144 governing re-sales, crypto start-ups can craft private sales to accredited investors to bootstrap their networks. It lowers reporting and disclosure requirements. The primary disadvantage comes from the advantage: less regulation. With less regulation and decreased requirements for disclosures, the potential for fraud is greatly increased for investors. This includes purposeful fraud as well as accidental fraud, which means less experienced business owners may inaccurately describe their business opportunities. Likewise, the genesis of crowdfunding provides blockchain-based founders the opportunity to capitalize on unaccredited investor excitement and launch a democratized platform by directly targeting diverse users. Finally, the JOBS Act's revisions to Regulation A and creation of Regulation "A+" offer developers the opportunity to raise up to \$50 million in a public round from a pool of accredited and unaccredited investors. Depending on the entrepreneur's priorities, all three exemptions offer compelling opportunities to elicit funding.

ICO: An initial coin offering (ICO) or initial currency offering is a type of funding using cryptocurrencies. It is often a form of crowdfunding, although a private ICO which does not seek public investment is also possible. In an ICO, a quantity of cryptocurrency is sold in the form of "tokens" ("coins") to speculators or investors, in exchange for legal tender or other (generally established and more stable) cryptocurrencies such as Bitcoin or Ether. The tokens are promoted as future functional units of currency if or when the ICO's funding goal is met and the project successfully launches. An ICO can be a source of capital for startup companies. ICOs can allow startups to avoid regulations that prevent them from seeking investment directly from the public, and intermediaries such as venture capitalists, banks, and stock exchanges, which may demand greater scrutiny and some percentage of future profits or joint ownership.[citation needed] ICOs may fall outside existing regulations, depending on the nature of the project, or be banned altogether in some jurisdictions, such as China and South Korea. Due to the lack of regulation and enforcement of securities law, ICOs have been the vehicle for scams and fraud. Fewer than half of all ICOs survive four months after the offering,

IEO: An IEO, or initial exchange offering is a fundraising method for cryptocurrency projects a lot like an Initial Coin Offering (ICO). The premise is still the same - a token issuer conducts an IEO to raise capital for project expansion. The difference lies in the use of a crypto exchange to host the token sale. The exchange does not necessarily legitimize the IEO project by default. The differentiator between an IEO and an ICO is the safety of the investment process (though some exchanges are registered and some are not, caveat emptor). IEO is everything ICO should have been. Safer, more reliable, better controlled. IEOs are substantially safer than the ICOs, nevertheless, investors are still required to conduct their own due diligence.

PSO: Private security offerings or private placements, typically restricted to certain high net-worth sophisticated investors, are not subject to some of the security laws and rules designed to protect investors, such as the comprehensive disclosure requirements that apply to registered offerings. Under the federal securities laws, a company may not offer or sell securities unless the offering has been registered with the SEC or an exemption from registration is available. **Offerings exempt** from the SEC's registration requirements pursuant to Securities Act Section 4(a)(2) or its safe harbor under Regulation D of the Securities Act are often **referred to as private placements**. Issuers offering securities in private placements are required to provide only limited disclosure to non-accredited investors, or may face no disclosure requirements at all. Therefore, investors in private placements are generally on their own in

obtaining the information they need to make an informed investment decision. Investors need to fully understand what they are investing in and fully appreciate what risks are involved.

STO/DSO: A security token offering (STO) /digital security offering/ tokenized IPO is a type of public offering in which tokenized digital securities, known as security tokens, are sold in security token exchanges. Tokens can be used to trade real financial assets such as equities and fixed income, and use a blockchain virtual ledger system to store and validate token transactions. Due to tokens being classified as securities, STOs are more susceptible to regulation and thus represent a more secure investment alternative than ICOs, which have been subject to numerous fraudulent schemes. Furthermore, since ICOs are not held in traditional exchanges, they can be a less expensive funding source for small and medium-sized companies when compared to an IPO. An STO on a regulated stock exchange (referred to as a tokenized IPO) has the potential to deliver significant efficiencies and cost savings, however.

SAFE: A simple agreement for future equity (SAFE) is an agreement between an investor and a company that provides rights to the investor for future equity in the company similar to a warrant, except without determining a specific price per share at the time of the initial investment. The SAFE investor receives the future shares when a priced round of investment or liquidity event occurs. SAFEs are intended to provide a simpler mechanism for startups to seek initial funding other than convertible notes. The regulatory and risk aspects of SAFEs resulted in them not being used very often.

Structure: Decentralized Autonomous Organizations (DAOs) is a computer program, without ‘legal’ organizational status (which naturally arises questions about ensuring its integrity from a legalistic and compliance perspective), that runs on top of a Blockchain and embedded within its governance and business logic rules. DAOs are autonomous which means they are fully automated and contain artificial intelligent logic. The fact that DAOs are purely decentralized entities (sort of like a digital corporation) enables them to run in any jurisdiction, thus they raise a big question as to how the current legal system could be applied to such a varied mix of jurisdictions and geographics.

Colored Coins are cryptocurrency that is used to represent digital assets (‘smart property’). Such coins represent ownership in non-fungible assets (such as a painting or even stocks and bonds). While Colored Coins have many of the advantages associated with Blockchain (security, transferability, etc.), there is a risk such digital asset could be misused or abused, particularly concealing value or trading in sanctioned assets (military related, etc). Whenever cryptocurrency starts to be tied to other assets, it does pose an issue if the cryptocurrency is just another financial instrument form subject to various compliance requirements. [Looks like a duck, quacks and swims like one, it might just be a duck].

Privacy: Striking a balance between anonymity (privacy) and transparency for compliance purposes (Bank Secrecy Act issues regarding anti-money laundering and terrorist or know your customer regulations) is often times a challenged decision. **Mixing Protocols** or **CoinSwap** is a way for parties to a cryptocurrency transaction to first send the crypto to a third party intermediary, then that intermediary sends, for a fee, from another account, cryptocurrency intended to be routed to another party, thereby concealing the relationship between true sender and receiver. While such act may be use for legitimate purposes, there is scope of risk of such methods being abused or misused.

etherBlockchain Speak Foreign Language. While the two Blockchain distinctions (with and without use of convertible virtual currency) seem somewhat easy to understand, complications arise because of the

new language and new technologies associated with Blockchain. Because the Blockchain industry is still in the early days of its birth, the language and definitions of Blockchain terminology varies from user to user. For example, there are conventional contracts (subject to being modified if justified) and then there are Blockchain unconventional immutable ‘smart contracts’ (can’t be changed, sort of like a vending machine, a product is automatically dispensed and you can’t get your money back or stop the vending process once the coin is inserted; or accidentally put 300 years instead of 30 years in a smart contract trigger day to pay college funds to a daughter – stuck with the 300 years) and much debate if conventional contract law applies to ‘smart contracts’ since there is no performing party – only an autonomous automatic computer code performed smart contract. One proposal for smart contracts is the ‘Ricardian contract’, which fundamental idea is to write a document that is understood and accepted by both a court of law and computer software – a User Interface Agreement.

Trying to insert the newly discovered “convertible virtual currency” asset species (aka cryptocurrency, native token, tokens, digital asset, digital currency, crypto, cryptoasset, etc. – pick your favorite brand) into (1) historical property and financial definitions or (2) existing conventional regulatory regulations and laws, is awkward at best, especially since conventional financial assets can either be:

- (1) touched (us human’s can physically hold a U.S. dollar bill in our hand but can’t clutch a bitcoin),
or
- (2) regulated by central authorities (be it a bank or government Treasury department, cryptocurrency is decentralized and no central authority administration).

Although there has been much debate about whether or not or how much of existing laws and regulations should apply to the new Blockchain cryptocurrency species, there have, in any case, been many legal and regulatory challenges from U.S. Federal, U.S. State and non-U.S. jurisdictions, based on existing laws and regulations against various cryptocurrency related entities. Many of these complaints are against identified internal parties within the Blockchain, and involve familiar claims⁶² associated with:

- The cryptocurrency is viewed as a regulated commodity, security or financial service and breach of security laws; the structure and *activities* of the business determined to be a regulated money services business involving money transmission services and money transmission by money transmitters (operating on a transactional or account basis) denominated in value that substitutes for currency; security fraud pump and dump schemes; unregistered security; fraud; misrepresentation; insider trading; control person liability; conspiracy to defraud; security or commodity regulatory violations; illegal trading; Bank Secrecy Act violations (failure to (i) comply with anti-money laundering - AML or (ii) failing to monitor Counter the Financing of Terrorism

⁶² Often before a complaint is formally filed, notice is given to the target defendant to come forward with preliminary information why a claim should be pursued. An example is a **Wells notice letter** that the U.S. Securities and Exchange Commission (SEC) sends to people or firms when it is planning to bring an enforcement action against them. It is issued at the conclusion of an SEC Investigation notifying the people or firm in question that the SEC has concluded that they should be charged with violation of the securities laws.[1] The notice indicates that the SEC staff has determined it may bring a civil action against a person or firm, and provides the person or firm with the opportunity to provide information as to why the enforcement action should not be brought.[2] The person or firm is generally given 30 days to file this response in the form of a legal brief considering legal and factual arguments as to why no charges should be brought against them. Although investigation is conducted on a confidential basis, this notice, as well as its response, is public information that can be used in later public hearings among other things.

/Know Your Customer – CFT/KYC or (iii) failing to file suspicious activity reports – SAR or currency transaction reports - CTR); criminal theft; illegal gambling of cryptocurrency; identity theft; theft of one’s virtual digital electronic wallet; violation of sanctions (sanctioned party illegally traded cryptocurrency); breach of fiduciary duty; Class Action Lawsuit; Racketeer Influenced and Corrupt Organizations Act (RICO) violations; violations of the Anti-Cybersquatting Consumer Protection Act; not registered as a money transmitter

- Breach of ‘smart contract’ claims: some allowed some dismissed. Smart contracts are contracts that are automatically performed by computer code (example, a stock sale limit order, timing of sale may be influenced by mechanical issues, and the stock automatically sold, can’t be stopped, at a time when its value is materially and adversely affected because of the delay – even a court order can’t stop the smart contract execution). Are smart contracts subject to conventional breach of contract laws? (Some States have ruled smart contracts are the same as conventional contracts, others have deemed them not to be contracts because of the lack of human performance intervention; some may allow unjust enrichment or court of equity claims to trump black letter law contract rights);

Similar Blockchain external claims,

- Financial currency exchanges charged with failing to properly register as a financial institution; failure to comply with Bank Secrecy Act regulations; income tax evasion; violation of patent law claims since many Public Blockchain developers are patenting their coding and procedures that could intentionally or accidentally be used by other Blockchains.

Other potential claims,

- Regarding the immutable (can’t be changed) Blockchain structure, how will privacy rights involving “Right To Be Forgotten” (RTBF) be managed (particularly regarding European RTBF rules);

The range of potential complainants in a cryptocurrency complaint includes:

- Federal
 - Securities and Exchange Commission
 - Office of Foreign Assets Control
 - Commodities Futures Trading Commission
 - Attorney General
 - Federal Trade Commission
 - Secretary of the Treasury
 - Financial Crimes Enforcement Network
 - Private Law Suits/Class Action Law Suits
- State Security and Banking Administrators

The way forward...

Regulatory and Legislative improvements (Federal and State and ideally collaborating in unison) should answer questions such as:

- Is convertible virtual currency a commodity? a security (the Howey Test)⁶³? a financial instrument – subject to existing or new law?
- Are smart contracts legally enforceable contracts?
- Who is the accountable party in a decentralized trustless Blockchain structure? – nobody is in charge, everybody is in charge.
- Similar to the principle that the polluter pays, should successful inventors of quantum computers be charged with the obligation that before such systems can become available, the inventor(s) are obligated to publish solutions to protecting cryptographic systems and privacy rights? The literature is awash with the risks of both being under threat because of the power of QC.

⁶³ Cryptocurrency offerings are tested by a U.S. Supreme Court four part security test, called a **Howey Test**, to assess if it is a regulated security: —(1) an investment of money; (2) in a common enterprise; (3) with a reasonable expectation of profits; and (4) the expectation of profits is based upon the entrepreneurial or managerial efforts of others. Applying the Howey test necessarily invites questions as to how the particular characteristics of various digital assets fall within each element. Bitcoin has been determined not to be a security since it does not fit all of these criteria.

APPENDIX A

COMPUTER SCIENCE

What is instructive in this computer science overview is that Blockchain technology can be viewed as just another application (app) that computers process – input certain information and data into the Blockchain app and out pops desired outputs and communicated to the user.

That output may be the result of analyzing a database or transfer of value from one party to another (a buyer exchanging value such as their money for purchased good sold by a seller).

Blockchain technology requires access to the internet and connection to user computers. Thus this short refresher on computer science and associated computer hardware and software are relevant to a broader understanding of Blockchain technology. The actual interface between one's computer and Blockchain technology is discussed in Part 1.

OVERVIEW

At a high level (and admittedly an over-simplification), there are three specialized 'career' paths a person may follow in regard to computer science:

- Computer Science or CS which generally deals with software and computer programming (writing computer code that gives instructions to a computer);
- Information Technology or IT which deals with computer networks (optimizing the organization of data and information, linking systems together and providing security); and
- Computer Engineering which deals with hardware (designing and developing powerful components like circuit boards, microchips, video cards, processors, hard drives, and routers).

For the purpose of this guide I have lumped all three paths into one and have selfishly described it as Computer Science (even-so, in practice all three of these paths overlap to some degree regardless of what specialized career path is followed).

There are various definitions of what Computer Science is or is not. To honor my keep it simple aspiration... computer science answers the question ***"What can be automated? utilizing science, technology and mathematics"***. Automation is linked with minimum or practically non-existent human involvement – a computer⁶⁴ 'automatically' solving complex math problems instead of using a human brain, paper and a pencil or a robot controlled by a computer performing an automated service otherwise performed by a human, such as assisting to build a vehicle.

- In other-words,

⁶⁴ Today, computers are usually thought of as the hardware or physical computer appliances us human's use such as our personal computers (Windows based PCs) or iPads or Apple Mac or large main-frame computers run by large corporations (such as IBM, Google, etc). In contrast, at one time *computers* were the descriptive title given to human 'computers' whose job was to solve technical problems using their brains, pencils and paper.

- what can be computed or analyzed (some things cannot be computed – for example what are the last five digits of pi (π) – see glossary of pi definition and why there can't be last 5 digits); and
- what amount of **resources** (such as computer hardware – the physical computer machines and electronics, and software – computer programs written by humans, plus people and energy (electric power) to operate the computers)

are required to perform those computations?

- And what are the time and space costs associated with different approaches to solving a multitude of computational problems (computer scientists spend much time trying to figure out what is the least expensive and most efficient way of performing a computation with a computer)? **Note the recent advances in Blockchain technology in regard to material reduced electric power requirements.**

Computer architecture is a description of the structure of a computer system made from its component parts.

The two main components are...

- (1) hardware – the physical machines and electronics that make up the computer system and how they are linked (networked) together – things that can be touched – and each architectural design, though all having some similarities with all other designs, is structured specifically to accomplish certain automated computations and therefore computer systems cannot interface easily, if at all, with all other computer systems, and
- (2) software – the language of communication, interfacing between humans (or other computers) and the hardware, associated with the instructions (the software) issued to the computer of what the computer is to do with information and data fed into it (the input) and how output results are communicated. As with any language, depending on the computer system architectural design, different computer software languages (described below) are not understood by all computer systems.

Blockchain technology is run by computer hardware and software, just like any other app.

THE HARDWARE

Hardware, located inside or adjacent to the computer protective covering box, generally associated with personal/business computers, encompasses the following main components:

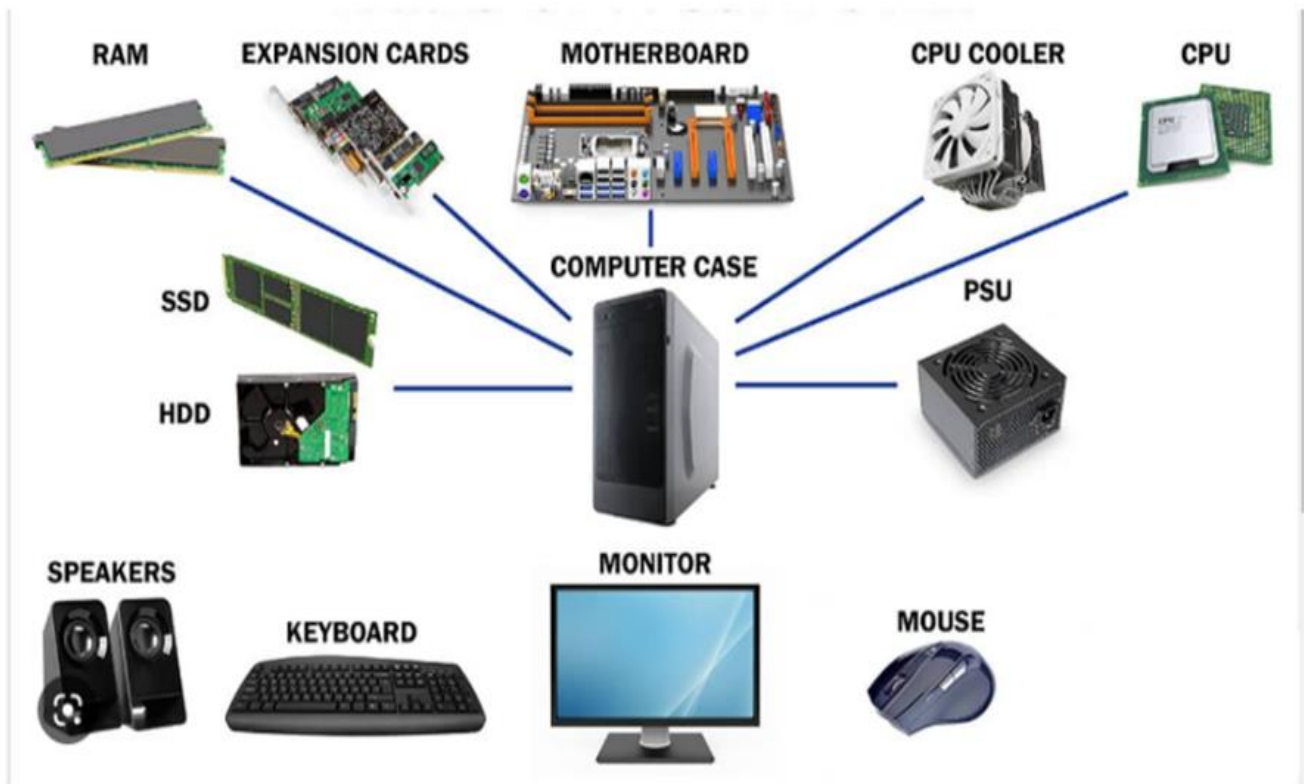
- Motherboard
 - Motherboard (integrated electronic circuit made up of semi-conductor and other electronic devices – such as resistors, capacitors, etc - mounted on a non-conductive 'green' board/card and the various electronic components connected together with wires or metal conductive strips) serves to connect all of the parts of a computer together.
- Central Processing Unit (CPU)
 - The *central processing unit (CPU)*, another integrated electronic circuit, is the computer component that's responsible for interpreting and executing ('processing') most of the commands from the computer's other hardware and software. A software

programmer/developer written computer code is converted (translated) into machine language or object code (binary code) by a 'compiler' that the computer understands.

- Random Access Memory (RAM)
 - Random Access Memory, or RAM, is the physical hardware inside a computer that temporarily stores data, serving as the computer's "working" memory. Such memory (data) is lost ('volatile' – lost if power is lost) when the computer is turned off. (We all have experienced losing our computer work because we forgot to 'save' the work and a power surge or interruption caused the RAM working memory to disappear into the cosmos and gone forever).
- Power Supply Unit (PSU)
 - The power supply unit is the piece of hardware that converts the power provided from the wall outlet (110/220 volt AC) into usable power (typically less than 15 volts DC) for the many parts inside the computer case.
- Video card
 - The video card is an integrated electronic circuit card that allows the computer to send graphical information to a video (visual) display device such as a **monitor**, TV, or projector.
- Hard Disk Drive (HDD)
 - The hard disk drive is the main, and usually most substantial, data storage ('nonvolatile' – not lost if power is lost) hardware device inside a computer. The operating system software, software titles, and most other software files are stored on the hard disk drive. The hard drive is sometimes referred to as the "C drive". This primary hard drive typically stores the root folder (the foundation location from which the computer obtains its software operating instructions to act and operate like a computer) of the operating software system used.
- Solid-State Drive (SSD)
 - Solid state refers to electronic circuitry that is built entirely of semiconductors. A solid state drive uses, as its primary storage medium, semiconductors rather than the magnetic platters of a conventional mechanical hard disc drive. Solid state drives and USB (Universal Service Bus – a type of connection or plug) flash drives use the same type of *non-volatile* memory semiconductor chips, which means they retain information and data when there is no power unlike RAM.
- Optical disk drives retrieve and/or store data on optical discs like CDs, DVDs, and BDs (Blu-ray discs), any of which hold *much* more information than previously available portable media options like the floppy disk⁶⁵.
- Card reader
 - A (**memory**) **card reader** is a device for accessing the data on a memory card such as a CompactFlash (CF), Secure Digital (SD) or MultiMediaCard (MMC). Most card readers also offer write capability, and together with the card, this can function as a pen drive (such as a USB memory stick we all have plugged into computer USB ports).

⁶⁵ Floppy disk is generally a relic of olden times computer science. At one time flexible plastic discs that feel like a flimsy CD, were encoded with magnetically stored data and information, and 'played' on a computer in a floppy disk driver. The disk would spin in the drive like a hard disc drive, and information and data magnetically read off of or written onto the floppy disk. At one time floppy disks contained the computers operating system software and had to be read into the computer each time the computer was used. Early day PCs contained dual disc drives, one for the operating system software floppy disc and the other as a working floppy disk. Now-a-days, such operating system software is stored in the Central Processing Unit.

- Monitor
 - The TV like communication screen us human’s sit in front of when using a computer.
- Keyboard
 - Suspect that high school typing class actually found some use as it’s faster to use a keyboard knowing ten finger typing than two digit typing. But with voice command, who needs fingers?
- Mouse
 - That point and click device that may be connected by wire or wirelessly to the computer. But voice command can make even a mouse redundant, sort of.
- Uninterruptible Power Supply (UPS)
 - A battery backup, or uninterruptible power supply (UPS), is primarily used to provide a backup power source to important computer hardware components in the event the main outlet power is interrupted.
- Printer
- Speakers
- External hard drive
 - An extra hard drive plugged into and located outside the computer typically with a USB plug.



The following hardware, essential to access Blockchain technology on one’s computer, is referred to as *network hardware*, and various pieces are often part of a home or business network:

- Digital modem (e.g., cable modem – data transmitted by a cable), Digital Subscriber Line – DSL - modem – (data transmitted by ordinary telephone lines)
 - A modem, is a hardware communication device which sends and receives data between two computers. It connects home or business computers, usually through a coax cable connection, to an Internet Service Provider (ISP), like Xfinity, through which computers are connected. A modem's purpose is to convert digital⁶⁶ information (machine or binary code language) from a senders computer to the ISPs analog⁶⁷ oscillating sine wave signals (the conversion process defined as the modulation process or transformation), and to reconvert at the receiver computer's modem, modulated analog signals back into useful demodulated original digital information (demodulation) that the receiver computer can use. A device that performs both modulation and demodulation is called a *modem* -- a name created by combining the first letters of MOdulator and DEModulator.
 - Digital Subscriber Line (DSL) technology – data transmitted over ordinary telephone lines instead of a coax cable - offers high-speed internet service for homes and businesses. It competes with coax cable and other forms of broadband internet. The technology behind DSL means the network (internet service) and telephone service share the same phone line without disrupting either voice or network connections.
 - The Internet Service Provider uses analog radio type oscillating sine wave communication signals in its transmission lines (with fixed frequency, amplitude and phase characteristics, unless the signal has been modulated).
 - A sender modem transmits computer generated machine language binary code digital data and information from the sender's computer by modulating (which means changing or modifying or modulating - one or more of the Internet Service Provider radio sine wave characteristics – a change in frequency, amplitude or phase, further illustrated in the below diagrams) –
 - (1) such as changing the wave's frequency – how fast the ISP analog wave oscillates, and/or
 - (2) changing the analog wave's amplitude – how strong the signal is or how high the sine wave oscillates and/or
 - (3) changing the analog wave's phase
 - which results in encoding or embedding the digital information signal from the sender's computer, onto the ISPs analog wave signal, while the receiver modem connected to the receiver's computer, demodulates (reverts) the ISP

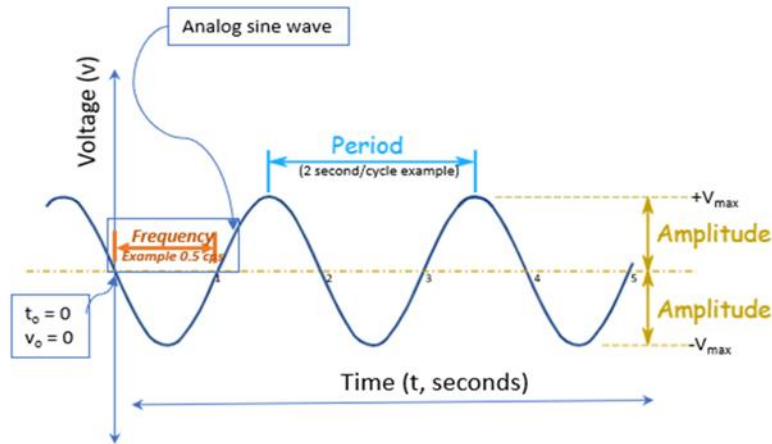
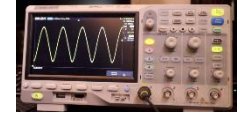
⁶⁶ Digital signals express variation in the system's variable (change in voltage, current, or other variable) in response to a set of discrete (lumpy not continuous) values (more like a light with an "on/off" switch or a three-way bulb with multiple, discrete levels of output). The signal is like a square saw tooth curve whose changes are in lumpy discrete jerky steps instead of continuous smooth transitions.

⁶⁷ An analog signal is a continuous signal in which one time-varying quantity (such as voltage, pressure, etc.) represents another time-varying variable. In other words, one variable is an analog (copy cat) of the other. For example, imagine a dimmer switch tied to a light bulb. The dimmer will have an infinite number of positions between "off" and "full" by varying the resistance in the dimmer from zero resistance (a closed connected circuit) to infinite resistance (open off circuit) which in turn varies the dimmer's output voltage – and a correspondingly infinite number of levels of output brightness (lumens) displayed by the lightbulb – if the dimmer causes the voltage to go up, the lightbulb gets brighter and vice versa – the copy cat effect – one goes up smoothly the other goes up smoothly. The output by the bulb is analogous to the time-dependent variable "position of the dimmer switch" which affects the resistance in the dimmer which effects the dimmer's voltage output.

modulated analog sine wave signal to recreate the original sent digital signal information that the sender and receiver's computers understands.

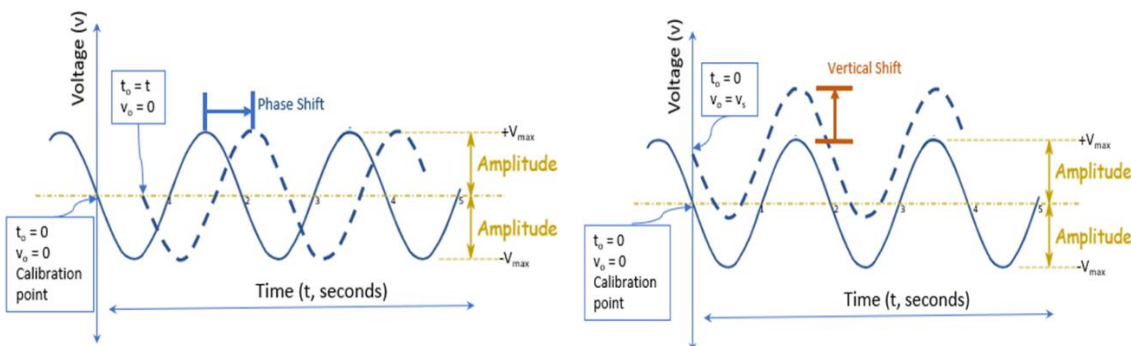
- The goal of a modem is to produce a modulated signal that can be transmitted by the ISP easily and decoded reliably.

The ISP's analog sine wave when measured with an oscilloscope, is represented as a wavy, oscillating signal as illustrated, showing the wave's Period, Frequency and Amplitude.



- ISP analog alternating current oscillating sine wave graph, Voltage (v) – vertical scale, Time (t) – horizontal scale
 - Sine waves are typically 'calibrated' such that time zero ($t_0 = 0$) for the beginning of the wave is established when the Amplitude measures 0 volts ($v_0 = 0$)
- **Amplitude** is the peak (maximum) Voltage (v) at any time (varies from $+V_{max}$ to $-V_{max}$ since the wave is oscillating).
- **Period (or cycle)** is the time (measured in seconds) between two adjacent maximum or minimum Amplitudes (example assume the Period is 2 seconds per cycle).
- **Frequency** is the number of Periods occurring in one second, the inverse of Period, or in this example Frequency = $1/\text{Period} = 1/2 = 0.5$ cycles per second (cps)
 - Frequency is a measure how fast the sine wave is oscillating up and down, the smaller the Period, the faster the Frequency and the faster the wave oscillates.
 - Typical residential home alternating current power Frequency is 60 cycles per second or a Period of 0.017 seconds between peak Amplitudes.

An analog wave Phase or Vertical Shift change are illustrated below:



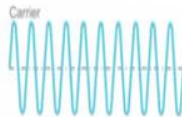
- **Phase Shift** is a measure of the shift in the sine wave horizontally if the wave calibration does not start or 'calibrated' at time zero ($t_0 = 0$) but starts at a later time t when the Amplitude is 0.
- **Vertical Shift** is a measure of the shift in the sine wave vertically at $t_0 = 0$, whereby the Amplitudes changes and the + and – Amplitude values are not equal to and mirror images of each other.

Graphical representation of what the modem is doing in regard to imprinting a computer digital signal (having discrete values of either 0 or 1 – its binary code) on to an ISP carrier analog sine wave...a modulating change, is illustrated below:

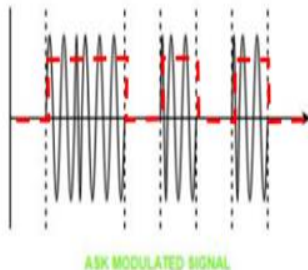
- **Digital Signal** – A digital signal is a signal that represents data or information from a computer, as a sequence of discrete (definite) values (such as either having a value of 0 or 1 in binary code); at any given time it can only take on one of a finite number of values.



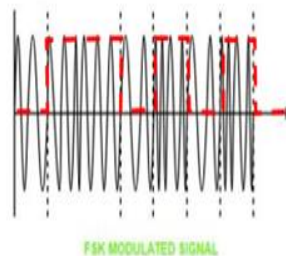
- **Analog Signal** – A carrier analog signal is any continuous signal (a radio frequency sine wave signal from an Internet Service Provider) for which the time varying feature of the signal is a representation of some other time varying quantity.



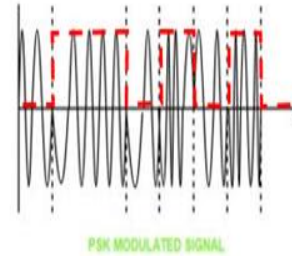
- The following techniques can be used for Digital to Analog (modulation) Conversion (imprinting the Digital Signal from the computer on the Analog Signal carried by the ISP):



Amplitude Shift keying – Amplitude Shift Keying is a technique in which carrier signal is analog and data to be modulated is digital. The amplitude of analog carrier signal is modified to reflect binary data.



Frequency Shift keying – In this modulation the frequency of analog carrier signal is modified to reflect binary data.



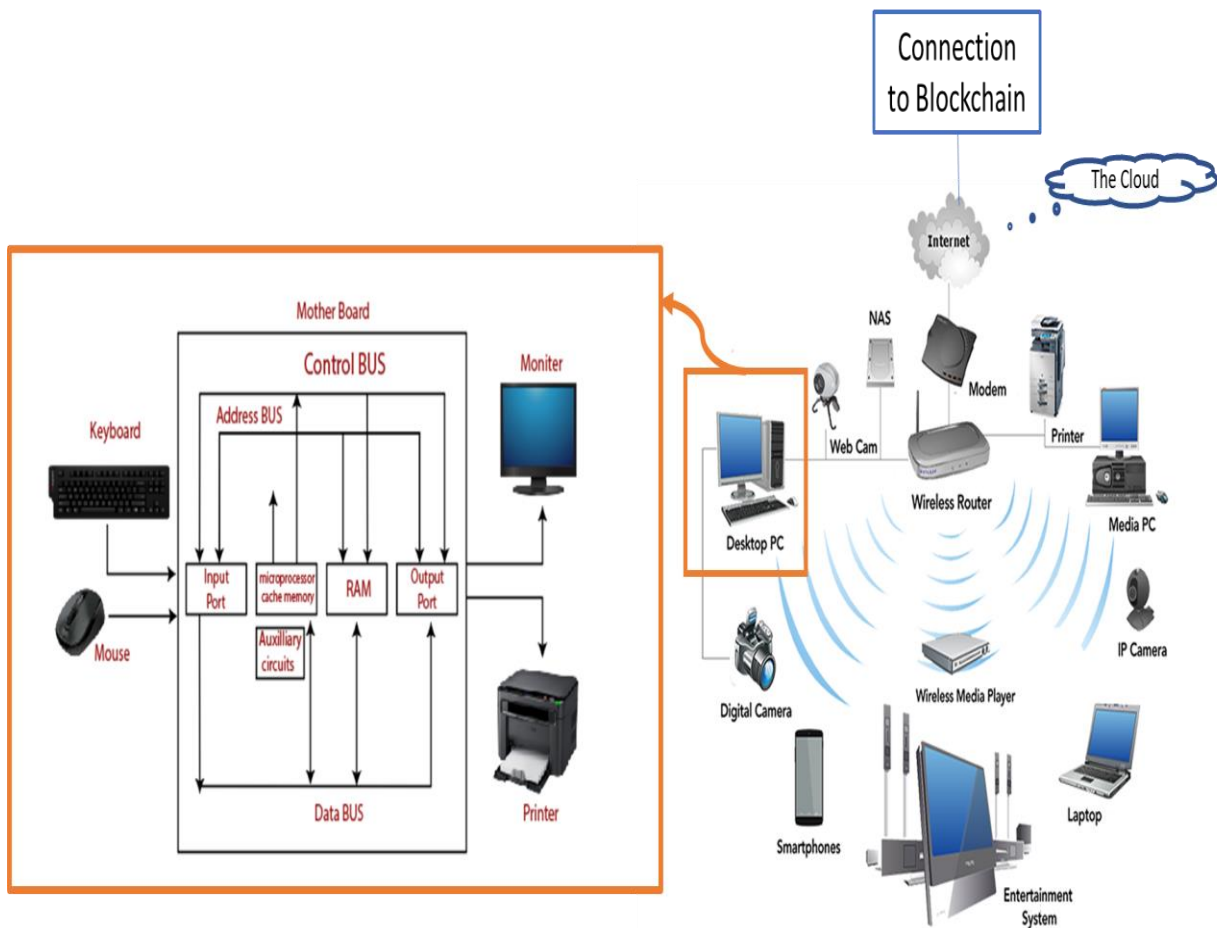
Phase Shift keying – In this modulation the phase of the analog carrier signal is modified to reflect binary data. The amplitude and frequency of the carrier signal remains constant.

When the modulated analog signal is demodulated in a modem, the modulated analog signal reverts to its original pattern and the output of the modem to the computer is the same digital signal that was sent by the sending modem.

What is instructive is that the operation of the modem does not affect access to Blockchain technology.

- Router
 - The router (connected through the modem), is the piece of network hardware that allows communication between the local home network—like personal computers and other connected devices (such as TVs, WiFi, printers, thermostats, security system, etc)—and the internet.
- Firewall
 - A network firewall protects a computer network from unauthorized access. It might take the form of a hardware device, a software program, or a combination of the two. Network firewalls guard an internal computer network against malicious access from the outside, such as malware-infested websites or vulnerable open network ports.

A graphical representation of a typical desktop (PC) computer hardware system architecture connected to the internet is illustrated below. The relevance of this graphic is how Blockchain technology interfaces at a high level to a computer system through the internet web and the modem.



THE SOFTWARE

Software is the language used to communicate between human's and computers (or other computers). That communication or instruction is for the purpose of causing the computer system hardware to

function a certain way or process information and data in a certain automated way with the objective of outputting a desired result in an efficient and effective and secure manner.

The task of a computer scientist is to design and develop algorithms (apps or programs) to solve a range of important problems.

Algorithm is

- a procedure or set of instructions for solving a mathematical] problem in a finite (fixed) number of steps that frequently involves repetition of an operation;
- a step-by-step method of accomplishing some task;
- a well-ordered collection of unambiguous and effectively computable operations that, when executed (or implemented), produces a desired result and halts the operation in a finite amount of time. Example:

Step 1: Do something..

Step 2: Do something...

Step 3: Do something...

Step 4: Display on the Monitor or Print the results of the task

Step Z: Stop, task is completed

Algorithms are used all the time in our day to day lives (though we don't use that term) whenever we follow a set of instructions to accomplish a desired task such as assembling a child's toy (insert tab A into slot B), baking a pie, balancing a checkbook or go through a university registration process.

Blockchain technology can be viewed as being a complex algorithm: a step-by-step method of accomplishing some task in a secure environment.

The design of algorithms (software programming) encompasses:

- Is the behaviour of an algorithm correct and efficient (their mathematical and formal characteristics)?;
- Is there hardware and computer systems available to execute (implement or process) algorithms?
- Are there programming languages available into which algorithms can be translated so they can be executed (implemented or processed) by computer hardware?;
- Are algorithms problem solving processes in place to identify important problems and design correct and efficient software packages to solve these problems?;
- Are there credible cryptographic algorithms in place to secure the operations?

The life cycle of developing algorithms is structured in a Program Development Life Cycle⁶⁸ (PDLC) process, summarized as follows:

Programming is the process of creating a set of instructions that instructs a computer how to perform a task. **Programming** can be accomplished using a variety of computer code "languages".

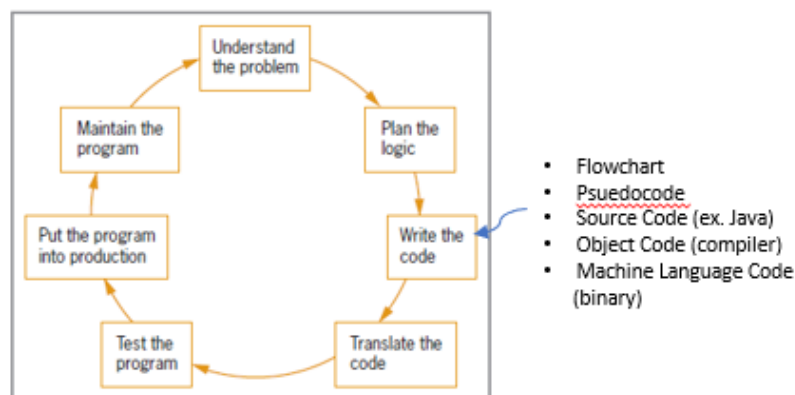
⁶⁸ Content extracted from California State University lecture notes published online.

First Some Relevant Definitions:

- **Syntax** refers to the spelling and grammar of a **programming** language. Computers only understand the **exact** form that the computer expects. Like a Password, any error, as simple as capitalizing the wrong letter, will result in an error. Syntax errors cannot execute in a computer.
- A **logic error** is an error in a program's **source code** (defined as any collection of software code, with or without comments, written using a human-readable programming language, usually as plain text) that results in incorrect or unexpected behavior when the program is executed.

PDLC is the process of systematically creating application **programs**, containing the following six sequential phases of **program development**:

1. Analyze and define the problem. The program developer must define the problem (written out in everyday plain language), then decide how to resolve it - choose a program language (source code such as Python, Java, C+, Basic, etc).
2. Design the program. A flow chart is often helpful, a visual diagram of the flow containing the program. Pseudocode may also be written at this point that converts the flow diagram logic into a simple language (often English) that is not understood by the computer, but helps in the next step of actually writing software source code, which is readable by a computer).
3. Code the program. Using the chosen source code program language to write the lines of code. The code is called the listing or the source code. The computer user will run an object code - which is the source code written by the programmer and run through the compiler and machine language (normally binary code) is produced. Errors in the source code can be detected at this point if the compiler reports object code errors. A source code is not executable or implementable until the object code is saved into an executable file, a library file, or an object file. (described below)
4. Debug the program. The computer user must debug. This is the process of finding the "bugs" on the computer. The bugs are important to find because this is known as errors in a program.
5. Formalize the solution. One must run the program to make sure there are no syntax and logic errors. Syntax are grammatical errors and logic errors are incorrect results. (The object code may run but the program determines incorrect results because of errors in the logic of the source code which the compiler may or may not detect.)
6. Document and maintain the program. This step is the final step of gathering everything together. Internal documentation is involved in this step because it explains the reasoning one might of made a change in the program or how to write a program

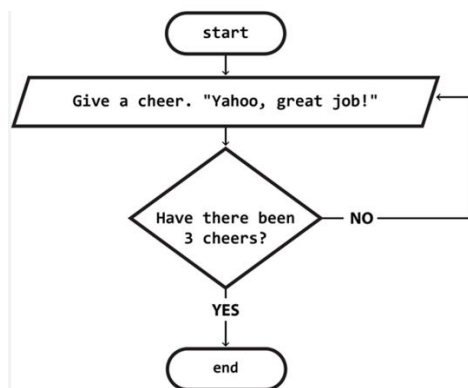


Writing Code

Computer code is a series of statements that have been assigned a function or process by a higher level language (typically referred to as source code such as Python, Java, etc). This language is similar to English and when converted to machine language (object code or binary code) using a type of program or translator needed to convert English language like coding into a computer language, understandable by a compiler, a computer can then understand the instructions in the machine language (object code) that it speaks. A program developer can write a simple code with a basic word processor or text editor, however, using a software application (specifically designed for coding in a particular language) is significantly more effective and efficient. A coding editor provides editing tools to ensure accuracy. A code editor is also known as an integrated development environment (IDE), which is a software application for formatting.

Flowcharts and Pseudocode

During the design process of the PDLC, it is important that programmers (and non-programmers) are able to visualize the way in which the program will work. Certain tools such as flowcharts and pseudocode are used to simplify the design process and allow the developers to see the program before any actual coding is used. A common type of design tool is the flowchart. A flowchart can be either handwritten or created with software such as Visual Logic or Flowgorithm. Flowcharts are also useful for education tools because they focus more on the concept of programming rather than focusing on the syntax of languages.



Another type of design tool is pseudocode (often written after a flow diagram has been prepared). Pseudocode is very similar to a programming language except that it uses non-syntactical 'simple' English words to summarize the processes of a program. Here is an example of pseudocode:

```
If user's age is greater than or equal to 18:  
Print "You can vote"  
Else  
Print "You cannot vote"
```

Compiler

A compiler is a special program that processes source code statements written in a particular programming language (such as Java, Python, C+, etc.) and turns them into machine language or "code" or object code that a computer's processor uses and understands. When executing (running), the compiler first parses (or analyzes) all of the source code language statements syntactically one after the other and then, in one or more successive stages or "passes", builds the output (machine) code, making sure that

statements that refer to other statements are referred to correctly in the final code. The compiler front end analyzes the source code to build an internal representation of the program, called the intermediate representation. The compiler backend includes three main phases, such as analysis, optimization, and code generation. Because compilers translate source code into object code, which is unique for each type of computer, many compilers are available for the same language.

Control Structures

A control structure, special logical operations used in source code, is a diagram used to show how functions, statements, and instructions are performed in a program or module (or algorithm). The diagram shows exactly when an instruction is performed, and how it's performed. Most importantly, a control structure shows the order of the instructions. There are three basic types of control structures: (1) sequential, (2) selection or conditional, and (3) repetition or iterative.

(1) Sequential operations: An operation in which a well defined task is sequentially completed first and when completed the algorithm sequentially moves on to the next task or operation.

- a. Example: Enter the number 1 into the computer (Step 1)
 Enter the number 2 into the computer (Step 2)
 Add the entered numbers: $1 + 2 = 3$ (Step 3)
 Print out the result: "3" (Step 4)
 End the operation, exit computer: Stop (Step 5)

(2) Conditional operations: A logical question-asking instruction.

- a. Examples: A = 1;
 If A is less than 1 then B = 2 else B = 3
 Result: B = 3

A = 1;
B = 2;
C = A + B;
If C is greater than 3 then C = 0 else C
Result: C = 3

(3) Iterative operations: These are looping instructions and depending on the condition, tell us not to move on to the next instruction but go back and repeat execution of the previous block of instructions.

- a. Example: Step 1 n = 0
 Step 2 A = 2
 Step 3 n = n + 1 (so first iteration n = 1)
 Step 4 A = A + 2 (so A = 4 after first iteration)
 Step 5 If n is not equal to 3 [*which it is not after first iteration*],
 then go back to Step 3 (*after first iteration go back to Step 3 and repeat steps 3-5*),
 else go to Step 6 (*if this step executes, then the looping process has stopped because n = 3*)
 Step 6 Print A (prints result of 8) (*there will be three iterations before this step is executed because n = 3 after 3 iterations*)
 Step 7 Stop

Testing Program Design

The program design is very important because it involves the overall step-by-step directions regarding the program. A programmer must test the program design to ensure that it runs correctly and that there are no mistakes. The programmer runs through lines of code to identify potential errors and to check the logic. Advantages to desk checking include the convenience of hands-on "proof-reading" of the programmer's own code.

Debugging

Debugging is basically making sure that a program does not have any bugs (errors) so that it can run properly without any problems. The first step to debugging is done before you can actually debug the program; the program needs to be changed into machine language so that the computer can read it. It is converted using a language translator (the compiler). The first goal of debugging is to get rid of syntax errors and any errors that prevent the program from running. Errors that prevent the program from running are **compiler errors**. Another kind of error is a **runtime error**, which occurs while the program is running and it is not noticed until after all syntax errors are corrected. Many run time errors are because of logic errors, which are errors in the logic of the program.

There are different types of debugging techniques that can be used. One technique called **print debugging**, or also known as the print method, finds errors by watching the print (or trace) statement live on the monitor or recorded to see the execution flow of the process. **Remote debugging** is the method of finding errors using a remote system or network, and using that different system to run the program and collect information to find the error in the code. **Wolf-fence debugging** method finds the error by zeroing in on the problem by continuous divisions or sectioning until the bug is found. (Is error in first half of the program? If not, break the last half into two parts and test the first part, if no error, then repeat the wolf-fencing sequential division process until the error is found.)

Testing/Implementation and Maintenance

After the debugging process occurs, another programmer needs to test the program for any additional errors that could be involved in the background of the program. An Alpha test is first conducted, which is on-site at the company, and Beta tests are sent out to different offsite third parties, states or countries to ensure the program is 100% ready for use. The Alpha test occurs before the Beta test. Once the debugging and testing are finished, the program is now in the system and the program implementation and maintenance phase are completed. This is the most costly to organizations because the programmers need to keep improving and fixing issues within the program.

Alpha testing is done "in house". Beta testing is done "out of house" and gives real customers a chance to try the program with the set intention of catching any bugs or errors prior to it being fully released. Alpha testing is typically performed by engineers or other employees of the company while Beta testing occurs in the "real world", temporarily being released to the public to get the widest range of feedback possible. If all goes well in both phases, the product is ready to be released.

BIBLIOGRAPHY

Bashit, Imran, *"Mastering Blockchain"*, Third Edition, (2020).

Cobb, C, *"Cryptocurrency for Dummies"*, (2004)

Huang, K, Mu, Y, Rezaeibagha, F, Zhang, X, *"Design and Analysis of Cryptographic Algorithms in Blockchain"*, (2021).

Kleppmann, M, *"Implementing Curve25519/X25519: A Tutorial on Elliptic Curve Cryptography"*, University of Cambridge, (2021)

James, L., *"Attorney General James Warns Investors About 'Extreme Risk' When Investing in Cryptocurrency, Issues Additional Warning to Those Facilitating Trading Virtual Currencies"*, <http://ag.ny.gov/press-release/2021>.

LabCFTC, *"A CFTC Primer on Virtual Currencies"*, Commodity Futures Trading Commission, (2017).

Lamport, L., Shostak, R, Pease, M, *"The Byzantine Generals Problem"*, SRI International, (1982).

Laurence, T, *"Blockchain for Dummies"*, 2nd Ed., (2019).

Nastase, R., *"Hacking For Beginners"*, (2018).

Nougayar, W., *"The Business of Blockchain"*, (2016).

Office of Foreign Assets Control, *"Sanctions Compliance Guidance For The Virtual Currency Industry"*, Department of the Treasury, (2021)

Preveil, *"Preveil Security and Design"*, (2022).

Sullivan, N., *"A (Relatively Easy to Understand) Primer on Elliptic Curve Cryptography"*, Cloudflare, (2013).

Tarbert, H, Blanco, K, Clayton, J, *"Leaders of CFTC, FinCEN, and SEC Issue Joint Statement on Activities Involving Digital Assets"*, <https://www.sec.gov/news/public-statements/cftc-fincen-secjointstatementdigitalassets> (2019).

U.S. Treasury, *"Application of FinCEN's Regulations to Certain Business Models Involving Convertible Virtual Currencies"*, FinCEN Guidance, Financial Crimes Enforcement Network, (2019).

White House, *"Executive Order on Ensuring Responsible Development of Digital Assets"*, <https://www.whitehouse.gov/briefing-room/statements-releases/2022/03/09>.

White House, *"Fact Sheet: White House Releases First-Ever Comprehensive Framework for Responsible Development of Digital Assets"*, <https://www.whitehouse.gov/briefing-room/statements-releases/2022/09/16>.