with i = 1, since $d_1 = 1$ we add P to Q and store the result in Q and double P;

with i = 2, since $d_2 = 1$ we add P to Q and store the result in Q and double P;

with i = 3, since $d_3 = 0$ we do not execute any sum, but we double P;

with i = 4, since $d_4 = 1$ we add P to Q and store the result in Q and double P;

with i = 5, since $d_5 = 0$ we do not execute any sum, but we double P;

with i = 6, since $d_6 = 0$ we do not execute any sum, but we double P;

with i = 7, since $d_7 = 1$ we add P to Q and store the result in Q and double P;

no binary digits of n are left to be taken into account, then returns Q.

The algorithm gives the result of multiplication by executing 5 sums and 7 multiplications. For each iteration of the loop this algorithm performs a summing operation, or alternatively a summing operation followed by another summing operation (doubling P), the loop is executed as many times as the binary digits of n, this leads us to **estimate a cost of O(logn).** So far we have talked about elliptic curves in which the variables and the coefficients belong to the real numbers, but **in their cryptographic application both the variables and the coefficients are restricted to the elements of a finite field**. In mathematics, a finite field, or Galois field, is a field with a finite number p^n of elements, with p prime number and is often denoted as $Z(p^{n)}$ or $GF(p^n)$.

The security of elliptic curve cryptography depends on the difficulty with which it is possible to perform the inverse operation, i.e. to determine n when n*P and P are given. This problem is called the discrete logarithm of the elliptic curve and it is a problem that is considered hard. **Currently the fastest known technique for calculating the logarithm is called the Pollard rho method. Designed by John Pollard in 1975, it was used in 1981 to factor Fermat's eighth number issue** (a Fermat number, named after Pierre de Fermat who first studied them, is a positive integer of the form $F_n = (2^2)^n + 1$. It was conjectured that all the Fermat number were prime number, conjecture that was proved to be false) **It is a probabilistic algorithm, in the sense that it does not guarantee to produce a result.**

In reality **there are some elliptic curves for which it is possible to find specific algorithms that solve the discrete logarithm in polynomial time**, such curves are not suitable for cryptographic uses and are therefore called weak. The possibility that some curves are intrinsically weak to a cryptographic analysis imposes several questions related to the trust that it is legitimate to place in objects of this type. Suppose, in fact, that someone proposes the use of a curve, how can we be sure that it does not have some kind of mathematical vulnerability not yet discovered that makes the problem of the logarithm solvable in polynomial times ? **To avoid** the eventuality that some attacker can forge a curve so as to include in it some **mathematical back-doors it is used the principle called nothing up my sleeve**, that is it is introduced a random number, called seed, which is used to generate curve parameters and the generator point, using hash functions. A curve generated by the use of a seed is called verifiably random, or randomly verifiable

Quantum Computing: the next (?) revolution in cryptography.

Quantum computing uses quantum-mechanical phenomena, such as **superposition and entanglement.**

We recall that **quantum superposition** is a fundamental principle of quantum mechanics. It states that, much like waves in classical physics, any two (or more) quantum states can be added together ("superposed") and the result will be another valid quantum state; and conversely, that every quantum state can be represented as a sum of two or more other distinct states. Mathematically, it refers to a property of solutions to the Schrödinger equation; since the Schrödinger equation is linear, any linear combination of solutions will also be a solution. An exemplification of this so strange property is the so-called Schrödinger's cat paradox. **Schrödinger's cat** is a thought experiment proposed by Austrian physicist Erwin Schrödinger in 1935 A cat, a flask of poison, and a radioactive source are placed in a sealed box. If an internal monitor detects radioactivity (i.e. a single atom decaying), the flask is shattered, releasing the poison, which kills the cat. The Copenhagen interpretation of quantum mechanics implies that after a while, the cat is simultaneously alive and dead. Yet, when one looks in the box, one sees the cat either alive or dead not both alive and dead. This poses the question of when exactly quantum superposition ends and reality collapses into one possibility or the other.

Quantum entanglement is a still more strange physical phenomenon which occurs when pairs or groups of particles are generated, interact, or share spatial proximity in ways such that the quantum state of each particle cannot be described independently of the state of the other(s), even when the particles are separated by a large distance - instead, a quantum state must be described for the system as a whole. Measurements of physical properties such as position, momentum, spin, and polarization, performed on entangled particles are found to be correlated. **For example, if a pair of particles is generated in such a way that their total spin is known to be zero, and one particle is found to have clockwise spin on a certain axis, the spin of the other particle, measured on the same axis, will be found to be counter-clockwise, as is to be expected due to their entanglement.** However, this behaviour gives rise to seemingly paradoxical effects: any measurement of a property of a particle performs an irreversible collapse on that particle and will change the original quantum state. In the case of entangled particles, such a measurement will be on the entangled system as a whole. Given that the statistics of these measurements cannot be replicated by models in which each particle has its own state independent of the other, it appears that **one particle of an entangled pair "knows" what measurement has been performed on the other, and with what outcome, even though there is no known means for such information to be communicated between the particles, which at the time of measurement may be separated by arbitrarily large distances.**

Such phenomena were the subject of a 1935 paper by Albert Einstein, Boris Podolsky, and Nathan Rosen, and several papers by Erwin Schrödinger shortly thereafter, describing what came to be known as the EPR paradox. **Einstein and others considered such behaviour to be impossible, as it violated the local realist view of causality (Einstein referring to it as "spooky action at a distance")** and argued that the accepted formulation of quantum mechanics must therefore be incomplete. **Later, however, the counterintuitive predictions of quantum mechanics were verified experimentally in tests where the polarization or spin of entangled particles were measured at separate locations, statistically violating Bell's inequality, demonstrating that the classical conception of "local realism" cannot be correct.**

In earlier tests it couldn't be absolutely ruled out that the test result at one point (or which test was being performed) could have been subtly transmitted to the remote point, affecting the outcome at the second location. However so-called "loophole-free" Bell tests have been performed in which the locations were separated such that communications at the speed of light would have taken longer—in one case 10,000 times longer—than the interval between the measurements. Since faster-than-light signaling is impossible according to the special theory of relativity, any doubts about entanglement due to such a loophole have thereby been quashed. **So there is something that is going faster than the light? Or we live in a Multiverse instead of a single Universe? The only answer, for the moment, is that the entanglement is a real phenomenon, that Schrödinger's cat can be simultaneously dead and alive and that the quantum computing should be possible.**

More in detail a quantum computer is a device that performs quantum computing. Such a computer is different from binary digital electronic computers based on transistors. **Whereas common digital computing requires that the data be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses quantum bits or qubits, which can be in superpositions of states.** A quantum Turing machine is a theoretical model of such a computer and is also known as the universal quantum computer. The field of quantum computing was initiated by the work of Richard Feynman in 1982.As of 2018, the development of actual quantum computers is still in its infancy, but experiments have been carried out in which quantum computational operations were executed on a very small number of quantum bits. Both practical and theoretical research continues, and many national governments and military agencies are funding quantum computing research in additional effort to develop quantum computers for civilian, business, trade, environmental and national security purposes, such as cryptanalysis. **A small 72-qubit quantum computer exists and is available for experiments via the IBM Quantum Experience project**

Large-scale quantum computers would theoretically be able to solve certain problems much more quickly than any classical computers that use even the best currently known algorithms, like integer factorization using Shor's algorithm (which is a quantum algorithm) and the simulation of quantum many-body systems. A quantum computers should be able to efficiently solve problems which are not practically feasible on classical computers. A classical computer has a memory made up of bits, where each bit is represented by either a one or a zero. A quantum computer, on the other hand, maintains a sequence of qubits, which can represent a one, a zero, or any quantum superposition of those two qubit states; a pair of qubits can be in any quantum superposition of 4 states, and three qubits in any superposition of 8 states. **In general, a quantum computer with n qubits can be in an arbitrary superposition of up to 2^n different states simultaneously.** (This compares to a normal computer that can only be in one of these states at any one time). An example of an implementation of qubits of a quantum computer could start with the use of particles with two spin states: "down" and "up". The advantage is clear: representing the state of an *n*-qubit system on a classical computer would requires the storage of 2^n coefficients, while to characterize the state of a classical *n*-bit system it is sufficient to provide the values of then bits.

However when the final state of the qubits is measured, they will only be found in one of the possible configurations they were in before the measurement. Thanks to this property, for some problems, quantum computers offer a polynomial speedup. The most well-known example of this is quantum database search, which can be solved by Grover's algorithm using quadratically fewer queries to the database than that are required by classical algorithms. **An example (and possible) application of this is a password cracker that attempts to guess the password or secret key for an encrypted file or system. This application of quantum computing is, obviously, a major interest of government agencies**. John Preskill has introduced the term quantum supremacy to refer to the hypothetical speedup advantage that a quantum computer would have over a classical computer in a certain field. Google announced in 2017 that it expected to achieve quantum supremacy by the end of the year, and IBM says that the best classical computers will be beaten on some task within about five years. Quantum supremacy has not been achieved yet, and **skeptics like Gil Kalai doubt that it will ever be.** The reason is that if quantum computing will work in a next future, we will have another proof that the existing world is really counterintuitive, much more of what one can imagine thinking to the famous Schrödinger's cat. In any case, a very promising technology, for the moment, still in the infancy state, that is developing very fast and that could operate a revolution in the cryptography.

VI LESSONS: THE TECHNOLOGY BLOCKCHAIN APPLIED TO BITCOINS. SYBIL ATTACKS AND YAP ISLAND.

Some info about blockchains for bitcoins

Every user who participates in the Bitcoin network **has a portfolio containing an arbitrary number of cryptographic key pairs**. Public keys, or "bitcoin addresses", act as sending or receiving points for all payments. Owning bitcoin implies that **a user can only spend the bitcoins associated with a specific address**.

The corresponding private key is used to affix a digital signature to each transaction, making sure that only the user who owns that currency is authorized to pay. The network verifies the signature using the public key. **If the private key is lost, the Bitcoin network will not be able to recognize in any other way the property of the money: the relative sum of money will be unusable by anyone and, therefore, to be considered irremediably lost .**

The addresses do not contain information about their owners and are generally anonymous. Addresses in readable form are random sequences of characters and digits **with an average length of 33 characters, always beginning with 1 or 3, of the form 1NAfBQUL4d2N7uu1iKxjwF8dESXTT3AKcq. Users can have an arbitrary number of Bitcoin addresses**, and in fact it is possible to generate them at will without any limit as their generation costs little calculation time (equivalent to the generation of a public / private key pair) and does not require any contact with other nodes of the network. **Creating a new key pair for each transaction helps maintain anonymity. Unlike most traditional currencies, Bitcoin does not use a central bank:** it uses a distributed database between network nodes that track transactions, but uses cryptography to manage functional aspects such as generating new currency and the attribution of bitcoin properties. The Bitcoin network allows the anonymous possession and transfer of coins; **the data necessary to use one's own bitcoins can be saved on one or more personal computers in the form of a digital "wallet", or maintained with third parties that perform functions similar to a bank.**

In any case, bitcoins can be transferred over the Internet to anyone with a "bitcoin address". **The peer-to-peer structure of the Bitcoin network and the lack of a central body makes it impossible for any government or non-governmental authority to block transfers, sequestration of bitcoins without the possession of their keys or the devaluation due to introduction of new currency.** Unlike legal tender currencies, bitcoins have the characteristic that no one can control their value due to the decentralized nature of the currency creation method. **In Bitcoin the amount of currency in circulation is limited a priori, moreover it is perfectly predictable and therefore known by all its users in advance**. Currency inflation in circulation cannot therefore be used by a central body to redistribute wealth among users.

Transfers are defined as a change of ownership of the currency, and are made without the need for an external body to act as a supervisor between the parties. This **mode of interchange makes it impossible to cancel the transaction and then re-appropriate the coins that have changed ownership.** The Bitcoin client transmits the transaction to its nearest nodes, which verify the authenticity and availability of the funds and in turn relay them back to the nodes to which they are connected. The total number of bitcoins tends asymptotically to the limit of 21 million. The availability of new coins grows like a geometric series every 4 years; in 2013 half of the coins were generated, by the end of 2017 we reached three quarters, so in less than 32 years all the coins will be generated.

The transactions, digitally signed, are stored in a public register shared by all the nodes and updated with the new transactions, using a scheme called Distributed Ledger Technology. The data structure that stores the transaction log is called a blockchain. The problems that blockchain allows to solve in the context of digital payments, are the **consistency of data in a distributed network**, the definition of a temporal order between events or secure intermediation in trustless environments. **The blockchain is structured as a chained sequence of blocks, each containing a certain number of transactions, in which in each block, with the exception of the first said genesis block, a link to the previous element is contained.** It is therefore a shared database on a peer to peer network that, containing the details of all the transactions carried out on the network, constitutes its historical memory. **The order in which the blocks are concatenated represents a sort of transaction timestamp** that can be used to determine if a given transaction occurred before another.

This idea was used also in the past. At least several hundred years ago, **islanders on Yap in western Micronesia were the precursor to Bitcoin and blockchain technologies.** Based on studies of rock sources and dating of sites on Yap and nearby islands**, before European contact in 1783**, inhabitants of Yap sailed about 400 kilometers to other islands in Micronesia to quarry limestone from caves and rock-shelters. Sea voyagers negotiated with local leaders for access to limestone deposits. Stone carvers went along for the ride and formed stone disks on site. A central hole was cut into each circular chunk of rock so men could run a wooden pole through the opening to hoist the rock. **These weighty pieces of currency, called rai, were transported to Yap on rafts.** Arriving back home, travellers presented newly acquired rai to their fellow community members at a public gathering. Everyone heard which individuals or clan groups took ownership of particular disks.

Each rai was assigned a value based on size, evenness of shape, stone quality and risks taken on the journey. After being inspected and verified by a local chief, rai were displayed at communal spots, such as ritual dancing grounds. **Ownership of a disk could be transferred**, for instance, as a wedding gift, to secure political allies or in exchange for food from residents of nearby islands after a severe storm. **These deals also occurred in front of the whole community. No matter who acquired a rai, it stayed in its original location**. Bitcoin and blockchain work in much the same way. The bitcoin units are transported and securely stored across the public blockchain ledger. **Yap islanders pioneered a public, oral system for securely tracking and exchanging rai. Blockchain does the same by maintaining digital histories and updates about units of cryptocurrency**.

The mechanism by which new blocks are chained to the blockchain is called mining. The amount of each transaction is tied to a specific network user by means of some simple instructions of **a non-Turing-complete language called Script. We recall that** a **Turing machine** is a mathematical model of computation that defines an abstract machine. The Turing machine **was invented in 1936** by Alan Turing. With this model, Turing introduced the computability theory. Turing completeness is the ability for a system of instructions to simulate a Turing machine. A programming language that is Turing complete is theoretically capable of expressing all tasks accomplishable by computers**. To do this, the limitations of finite memory are ignored.** This means that the system can end in a loop. In order to avoid this problem, for the bitcoins Satoshi Nakamoto chose a non-complete Turing language.

In a network where all nodes share constantly changing information, there is the problem of finding a common agreement on what the current state of the system is, this problem is a problem of consensus and the blockchain resolves it in a decentralized way. **In the context of digital payments, these properties guarantee that it is possible to receive or transfer digitally the value, or more generally, an asset, without having to trust the counterparty of the transaction or in a regulatory authority. or in a completely trustless environment.** If we abstract the characteristics related to payment systems, we can use this technology in any context in which we want to make a connected system able to regulate in a secure and independent manner any type of intermediation that may occur within.

In a distributed network, it is often necessary to certify the creation or modification date of an information so that no one, not even those who have created the information, can alter it. To implement such a system, it is necessary to publicly make available an infrastructure capable of collecting, processing and renewing time stamps, of digital documents. You can base this type of infrastructure on **a centralized entity called TSS or Timestamping Service** which receives the hash of a document, puts a time stamp on it, generates a new hash starting from the previous one and from the time stamp. his digital signature and sends the signed hash and the timestamp back.

We recall that a hash is any function that can be used to map data of arbitrary size to data of a fixed size. For instance, we can map a number of 3 cyphers in a smaller number summing the cyphers forming the number. I.e. we map 156 in 1+5+6 =12. 344 in 3+4+4 = 11. The values returned by a hash function are called **hash values or simply hashes.** Hash functions accelerate table or database lookup by detecting duplicated records in a large file. They are also useful in cryptography. A cryptographic hash function allows one to easily verify that some input data maps to a given hash value, but if the input data is unknown, it is deliberately difficult to reconstruct it (or any equivalent alternatives) by knowing the stored hash value. This is used for assuring integrity of transmitted data. In cryptographic applications we ask, for example, that the hash function has the following properties:

- **resistance to the pre-image**: it is computationally intractable to search for an input string that gives a hash equal to a given hash;
- **resistance to the second pre-image**: it is computationally intractable to search for an input string that gives a hash equal to that of a given string;
- **collision resistance**: it is computationally intractable the search for a pair of input strings that give the same hash

The length of the hash values varies depending on the algorithms used. **The most commonly adopted value is 128 bits, which offers good reliability in a relatively small space**. However, the possibility of using **hashes of larger size SHA (a family of five different cryptographic hash functions developed since 1993 by the National Security Agency (NSA)), for example, can also produce strings of 224, 256, 384 and 512 bits.** The hash functions play an essential role in cryptography: they are useful for verifying the integrity of a message, since the algorithm execution on a text even minimally modified provides a completely different message digest compared to the calculated one on the original text, revealing the attempted change. **The hash functions can also be used to create digital signatures,** as they allow the rapid creation of the signature even for large files, without requiring long and complex calculations**: it is in fact computationally more convenient to quickly execute a hashing of the text to be signed, and then authenticating only that, thus avoiding the execution of the complex algorithms of asymmetric cryptography on very large amounts of data**. Returning to the blockchain to verify the veracity of the timeline applied to a document **must calculate the hash of the document, calculate a new hash from the latter and the timestamp and compare the result with the message signed by TSS.**

An infrastructure of this type requires that a considerable amount of trust be placed in the certifying authority, which is not always possible. Determining whether a document was created before another implies that it is possible to establish a total order relationship between pairs of documents. In a centralized system complex enough to have multiple TSSs and in which delays or transmission problems may occur, it is possible that this order relation is partial, i.e**. it may not be possible to establish for some documents what was created before. A solution can be to provide a partial order together with a system to solve cases of indecision.** The blockchain represents a decentralized timestamp service, in fact a block is considered more recent than another if it is added after it, for this sort the transitive property holds and furthermore, assuming that the sequence of blocks not present bifurcations, this order is total. **As a result, a transaction is considered more recent than another if it is contained in a more recent block. In case the two transactions**

were contained in the same block, they are sorted according to the order in which they appear inside the block.

How to find the agreement? If we think of a group of individuals who must agree on something, **an idea to solve the problem is to proceed to a simple vote and let the majority decide**. This approach can be valid if they vote, at the most, only those entitled, but in a digital system, where it is possible to create fictitious identities that act as voters, this mechanism could does not work. If it is possible to create a sufficient number of fake identities to subvert the outcome of the vote, then a system to reach consensus by quorum could be controlled at will, this situation is known as Sybil Attack. **A Sybil attack can work if, within a voting system, an attacker can create an arbitrary number of pseudonyms without any particular burden.** Therefore, not being able to prevent the creation of more voters' identity, **one of the solutions consists in requiring that the voting operation has a cost that makes it uneconomical for an attacker to create aliases** with which to subvert the outcome of the vote. vote. This approach is based on the proof-of-work concept, which is an operation that is expensive to calculate, but simple to verify.

VII LESSON: HOW IS POSSIBLE TO AVOID THE DOUBLE SPENDING? HOW DOES THE BLOCKCHAIN TECHNOLOGY WORK IN THE BITCOIN WORLD?

Suppose we want to implement a simple election based on proof-of-work, a vote will be a string that expresses the voter's preference. The result of the vote will be determined by the majority of the votes collected in a given time frame and each participant will be able to vote as often as he likes, **the only condition that we will impose is that the hash of a valid vote begins with a particular string, for example a single zero. To generate a valid vote we will have to create a string in which the name of the preference is expressed, plus a nonce, made up for example by the number of the attempt in progress and by a random value.** In this way the last part of the voting string changes until its hash does not respect the established criterion.

Furthermore. By protecting the voting operation with a computational cost, we have somehow discouraged a determined attacker to flood the digital urn with thousands or millions of votes, making it uneconomical to influence the system with a Sybil attack, or to cause disruptions due to high traffic generated. It is not a coincidence that HashCash, one of the first applications of the proof-of-work concept, was aimed at containing the problem of spam and DoS attacks. **An alternative way to reach the majority to award the vote could be to fix a rather high difficulty and count only the first valid vote found. This situation is equivalent to organizing a sort of competition whose winner will receive as a prize the right to express the only valid vote.** To get an idea of the difficulty of requesting such a competition, requesting a digest that starts with 6 zeros would require more than 700,000,000 attempts to calculate a valid vote

It may seem a bit extreme to handle the problem of consensus, but it is not unlike what is being done by the mines to link a new block to the blockchain. Each miner generates a new block including new transactions in it and calculates the proof-of-work, as soon as it is able to calculate a valid hash it concatenates blocking to the blockchain. For each block successfully linked to the miner, a certain amount of new bitcoins is paid to compensate for the computational effort sustained. Because a system of this type functions it is necessary that the output of the selected hash function is completely different every time we modify the input. **Therefore generating a valid vote is not just a difficult problem, but it is also something very similar to trying to win a lottery, since every attempt returns a completely different result from the previous one.** The probability of finding a digest that starts with a number n of zeros is equivalent to determining the probability of a partial collision with the string consisting of n zeros. Each character of a digest generated by SHA-256 is a hexadecimal number, that is, it can assume 16 possible values, so the probability of a collision with a specific string is the inverse of the number of possible hash of length n, that is (1/16) to the power n.

Another problem to be faced is that related to the **limitations given by Theorem CAP**, i.e. that a distributed system can at most guarantee two of the three properties: consistency, availability and partition tolerance. Obviously**, since availability and partition tolerance are indispensable**, the property to be weakened is the one relating to consistency. Consider the three levels of consistency that data can have in a distributed system:

- **no consistency**: the system does not need any consistency and, not having problems in data synchronization, it allows to have a service availability and a tolerance to the maximum partitions;
- **strong consistency**: the system requires that at all times all the individual nodes see exactly the same data, so if a node were to disconnect, the system would be unable to continue its operation, at least until the node reconnects;
- **weak consistency:** the system can continue to operate in the case of partitioning the network, allowing data to be misaligned, at least temporarily.

The most interesting consistency category for a distributed fault tolerant system **is the weak one**, so it is essential to establish a policy that will allow you to resolve any conflicts once the partitioning is over, re-aligning the data between all the nodes**. The blockchain guarantees a weak consistency model of the data in which the availability of the transaction log is guaranteed to all connected nodes**, with some peculiarities:

- **the records of transactions** held by some nodes **may not be synchronized**, at least temporarily;

- **the system must be able to converge** on a global ordering of all transactions;

- **the system must be able to resolve conflicts** in the data in a predictable way;

- **the system must prevent a bitcoin from being spent more than once**

- **the system must be protected against Sybil attacks.**

The blockchain has been designed to cryptographically ensure the integrity of the transaction log, for this purpose the entire infrastructure is built as a communication system based on digital signature with the following characteristics:

- **information** regarding a single transaction **cannot be corrupted** without leaving a trace;

- a **transaction**, once finalized, **cannot be repudiated** by the parties;

- **each transaction is attributable** to the counterparties involved.

The digital signature serves both to constrain each transaction to its owner, who is the only one who can dispose of the funds contained in it, and **to ensure that the funds transferred from a transaction are paid only to the legitimate recipient**, who will become the new master. Once a transaction has been accepted by the system, the only way to alter its data is to modify the block in which it appears, in addition to all the blocks that have been concatenated to it later, which is computationally impractical. **The algorithm used for the digital signature is based on the arithmetic of elliptic curves** and its strength depends mainly on the type of elliptical curve that is chosen, some curves in fact have some mathematical vulnerabilities and therefore their use is not recommended in cryptography.

The particular elliptical curve used and made popular by Bitcoin is called secp256k1 and, unlike the standard curves recommended by NIST, is a curve whose parameters and constants have been chosen in a predictable and well documented manner. For this reason it is thought to be rather unlikely that its creator has deliberately introduced you some mathematical backdoors. In addition, it is a curve that lends itself particularly to efficient calculations on a computer, allowing an increase in performance up to 30 % compared to other curves if the implementation is quite optimized. In addition to using the digital signature algorithm, the blockchain uses different hash algorithms to ensure the integrity of the transactions contained in each block and the block itself, to achieve consent by proof-of-work and to generate valid bitcoin addresses.

A **Public Key Infrastructure or PKI** is an infrastructure composed of both hardware and software tools as well as **procedures and protocols used to ensure that the electronic transfer of information between a set of subjects can be carried out safely**. When there is a need to send confidential data between different counterparts in distributed and therefore potentially insecure environments, three components are needed:

- **authentication** of participants in the system;

- **encryption** of messages;

- the **non-repudiation** of messages;

The authentication of the participants serves to validate the identity of those who send and receive messages on the infrastructure, be they individuals, websites or software agents, ensuring their identity. This component ensures that on the PKI everyone is who he claims to be.

The encryption of the messages ensures the confidentiality of the data sent, if an unauthorized person were to intercept the message in fact could not decrypt the content if the encryption algorithm used is quite robust. This component ensures that data is readable only by those who are effectively authorized to do so. The non-repudiation of messages is the assurance that the exchanged messages cannot be altered during their transfer and also guarantees the authenticity of the sender's identity. This component ensures both the paternity and the non-alterability of messages sent over the network. **The network of Bitcoin nodes joined to the blockchain can be considered as a PKI in which, in order to guarantee to every node the possibility to control the transaction register, the encryption of the messages is not implemented.**

Bitcoin is born to be free from any form of control or constraint, therefore it is entirely open source. This guarantees all interested parties the possibility of verifying the code or protocol, reporting problems and proposing improvements. A decentralized infrastructure removes the need to have bodies that certify and finalize transactions, with obvious positive repercussions of practical order:

- the single point of failure of the system is eliminated and the load is distributed on all the nodes making the system more robust and resilient;

- **eliminates the risk that a centralized entity**, responsible for the validation of transactions or the issuance of money, betrays this trust due to incompetence or dishonesty;

- **the brokerage costs for operations are reduced**, because it is no longer necessary to maintain a separate infrastructure to validate and make transactions effective;

- **it is not technically possible to confiscate** or steal funds from a user unless he knows his private key.

Bitcoin is a push payment system, i.e. **it does not require anyone who sells, accesses or stores information about their customers and their transactions. In a pull system, instead, this information accumulates in the databases of e-commerce sites and represents sensitive objectives that can be violated**, compromising the security of the data contained, such as credit card credentials or personal information of customers. In Bitcoin all transactions are referred to pseudonyms, so it is not immediate to establish within the system, a link between a real physical identity and a set of transactions and vice versa. **It is not necessary to place trust in the human factor**, which is generally the weak link in a security-oriented structure, in fact there is little to do with firewalls, cryptographic protocols and security devices to stop an individual or an organization that, by acting dishonestly, tries to take advantage of the system it should control. On the contrary, by its nature, **Bitcoin presents some disadvantages that are difficult to solve, for example the funds can only be used by means of a private cryptographic key; if this were to be lost, the related funds would also be lost without any possibility of recovery**.

Since this is a system without authorities and guarantors and financial institutions that provide services to the public**, there is no assistance service to refer to in case of problems such as loss or theft of your wallet. Bitcoin has attracted the attention of financial speculators** and this in the absence of a central authority that regulates the operating parameters **has led to an extreme volatility of its value.** Bitcoin is a peer-to-peer electronic money system that ensures secure brokerage and value transfer in trustless environments. **Unlike traditional payment systems, it allows counterparties to perform direct operations,** delegating their validation and implementation to the entire network of participating nodes.

Let's see what happens when two users decide to make a transaction, let's assume that **Bob wants to transfer 1 Bitcoin to Alice:**

- **Bob's wallet creates a new transaction** and sends it to the network;

- **the nodes receive the transaction** and after having validated it they send it to their neighbours;

- **the transaction arrives at a miner** that verifies its correctness;

- **the miner inserts it, together with the other transactions to be confirmed, of which he has knowledge, in a block;**

- **the miner calculates the proof-of-work of the block** and adds it to the Blockchain propagating it on the network;

- **the nodes receive the new block**, verify it and send it to their neighbours;

- **the block arrives at Bob and Alice**, confirming the success of the operation;

The transaction sent by Bob goes a long way in the peer to peer network, is propagated until it reaches the miner that will succeed in inserting it into the new block of the Blockchain and from there, inside a new block, reaches all the nodes of the network, is propagated again throughout the network, this time as a confirmed transaction, i.e. included in the Blockchain, until it reaches the two nodes representing Alice and Bob. **Despite all the steps necessary for validation and implementation, the transfer of value takes place directly from those who pay to whom receives, being Bitcoin a push payment system. The time needed to concatenate a new block to the Blockchain is about 10 minutes which, considering the propagation times of the messages within the network to be negligible, represents the time necessary for each transaction to be verified and included in the Blockchain.**

The Blockchain is a data structure that contains the database of completed transactions, stored within concatenated blocks in a single linked structure. This data structure is shared by all the nodes of the network and is continually updated by concatenating each new block to the last one. To maintain the relationship with the previous element in the chain, **each block stores the blockhash of its own father, this value is calculated as a hash of the block and therefore uniquely identifies it.** The unique back link for which each block has only one parent **but may have, at least temporarily, more than one child, that is, there may be more blocks that refer to him and that thus they generate a fork.** Generally this inconsistency **occurs when two or more blocks are discovered and propagated almost simultaneously**, but once the bifurcation is resolved with appropriate mechanisms, only one of the blocks will be confirmed in the Blockchain remaining an only child.

VIII LESSON: THE MINERS. THE NEW TECHNOLOGIES APPLIED IN MINING.

The blocks are deposited on one another as geological sediments and the sequence in which they are linked is a temporal order for the transactions contained therein. Moreover, like the geological sediments, **the most ancient blocks are even more stable because their value influences the blockhashs of all the blocks that follow it in the chain so to modify a block it is necessary to modify also the whole chain that descends from it so that it is composed of valid blocks.** Mining is the process by which new bitcoins are produced but above all the mechanism by which consensus is reached on the status of transactions within the network. The miners, that is the nodes specialized in this operation, are responsible for keeping the transaction log updated. In the intent of the protocol the task of the miners is essential for the proper functioning of Bitcoin and represents a service of public utility, incentivized by the perspective of individual gain.

The difficulty of the work required by the miners to generate new blocks is designed to automatically adapt to maintain the constant frequency of the constant Blockchain. To allow us to, **every 2016 blocks the network resets the proof of work difficulty based on the speed with which the blocks were added** and therefore based on the computing power of the network. Since a transaction ca not be included in more than one block and since the blocks are added individually to the last element of the Blockchain, **a real race is going on between the miners, said mining competition, to be able to calculate a new block before the others.**

The competition, also due to the increase in bitcoin prices, has increased dramatically in recent years and this has forced the miners to increase its computing power to stay in the game and be competitive in the race to calculate the new blocks. From this upward game there arose a vicious circle whereby the computational power of the miners progressively increased, but this increase was matched by an increase in proof-of-work difficulty. This means a incredible waste of energy. It is estimated that **the current global power consumption** for the servers that run bitcoin's software is a minimum of 2.55 gigawatts, **almost the same as Ireland**. Google, by comparison, used one fifth of this energy in 2015. What's more, bitcoin "miners"