FUNDAMENTOS DE BITCOIN Y BLOCKCHAIN

CARLOS RODRIGO GUZMÁN DURÁN & VICENTE MUÑOZ

ABSTRACT. Bitcoin es un conjunto de conceptos y tecnologías que forman la base de un ecosistema de dinero digital. En este curso daremos una pequeña introducción a las características monetarias de Bitcoin para después adentrarnos en la tecnología que da certeza a la red incluyendo los mecanismos de encriptación. Se explicará cómo crear cuentas y cómo hacer transacciones con bitcoins.

El curso estará organizado de la siguiente manera:

1. Introducción y panorama general.
2. Dinero electrónico descentralizado.
   - Problema de los generales Bizantinos.
   - Consenso decentralizado de Nakamoto.
   - ¿Por qué Bitcoin es dinero? ¿Qué es dinero?
3. Protocolo de la red Bitcoin.
   - Árbol de Merkle y Blockchain.
   - Ledger de Bitcoin.
   - Creación de bitcoins.
   - Cuentas y monederos Bitcoin. Creación de cuentas de la manera más segura.
   - Transacciones.
   - Minería y ajuste de dificultad.
   - ¿Cómo comprar y vender bitcoins?
4. Mecanismo criptográfico (protocolo matemático).
   - Funciones Hash.
   - Aritmética modular: cálculo computacional de inversos y potencias.
   - Encriptación con potencias modulares.
   - Criptografía de clave pública.
   - Firma Digital.
   - Curvas elípticas y su aplicación criptográfica. Elliptic Curve Digital Signature Algorithm (ECDSA).
   - Prueba de trabajo.
   - Forks y ataques maliciosos.
   - Lightning
5. Otras aplicaciones de las tecnologías Blockchain: Ethereum, monedas estables y "contratos inteligentes".

Date: 15 junio 2022.

1. Introducción y panorama general

El 1 de noviembre de 2008, un programador informático con seudónimo Satoshi Nakamoto envió el marco teórico (White paper) de Bitcoin a una lista de correo de criptografía títulado "Bitcoin: A Peer to Peer Electronic Cash System".

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1.1. **Bitcoin se basa en:**

- Network decentralizado (red P2P).
- No hay autoridad central. De hecho, no existe autoridad.
- No hay un *único punto de ataque* al sistema.
- Cualquiera puede ser un nodo (participar en el sistema). Unirse o dejarlo (o tratar de sabotearlo).
- Portarse bien tiene un rédito positivo.
- La coordinación de nodos surge de forma espontánea (como las hormigas).

1.2. **Características de la red Bitcoin:**

- Resiliencia. El sistema resiste eventos negativos:
  - Silk road y Deep web.
  - El robo al Exchange Mtgox `https://en.wikipedia.org/wiki/Mt._Gox`
  - Control gubernamental y prohibiciones (Rusia y China).
  - Burbujas especulativas (primavera 2013, invierno 2014, invierno 2017, 2021).
  - Cambios de código (forks).
- Incluso sale reforzado.
- Transparencia: el protocolo Bitcoin es de código abierto (open source).
- De hecho, demasiada transparencia (todo el mundo puede leer los saldos de todas las cuentas).
- Anonimato (o no): no hay un listado de qué personas poseen qué cuentas.
- Sistema descentralizado, no respaldado por gobierno ni un emisor central.
- Elimina los bancos y políticas monetarias (para bien o para mal).
- Cuanta más gente participe, más estable, seguro y resistente es el sistema (en partícular, porque hay más gente interesada en que funcione).
- De hecho, comprometer la red destruye la confianza y no se saca rédito.

1.3. **La moneda Bitcoin:**

- La moneda bitcoin es dinero virtual (OK, el dinero *fiat* también). Pero bitcoin es completamente virtual.
- Siglas Btc y logo:

- El network Bitcoin se puso en marcha el 9 de enero de 2009, cuando se generó el primer bloque (genesis block).
- Se genera lentamente hasta un tope de 21 000 000 Btc.
- No es inflacionaria (de hecho, ligeramente deflacionaria) cuando se estabilice.
- Cada bitcoin se divide en 10^8 partes llamadas Satoshis.
- Valor refugio (el oro del siglo XXI).
- No es práctico para pequeñas compras, las transacciones son lentas (aprox. 1hr.).
- Respaldado por el sistema de nodos (participación abierta), por el mecanismo criptográfico (protocolo matemático).
- Es la moneda perfecta: inútil, divisible, ligera (transportable), infalsificable, no inflacionaria, no controlada por gobiernos.

2. Dinero electrónico descentralizado

2.1. ¿Qué es el dinero?

- RAE: Del latín *denarius*. Medio de cambio o de pago aceptado generalmente.
- Libros de economía: bien o activo aceptado generalmente como medio de pago que sirve de unidad de cuenta y depósito de valor.
- Históricamente se han usado muchos tipos de "dineros": conchas marinas, cuentas de cristal, sal, pieles, piedras ray, cereales, cigarrillos, latas de conserva, metales, monedas, billetes, bits...
- Propiedades comunes:
  - Consenso
  - Durabilidad
  - Escasez

2.2. La naturaleza del dinero:

- El dinero es **confianza**: se acepta dinero como pago si confías en que va a ser aceptado en el futuro.
- Cualquier cosa puede tener la función del dinero si tiene buenas propiedades.
- El dinero, como las nociones abstractas matemáticas, existe por sus propiedades.
- Cualquier objeto o noción, material o inmaterial, puede ser dinero (no es necesario "poder tocarlo").

2.3. Propiedades del buen dinero:

El buen dinero:

- No es fácil de falsificar o producir.
- Es fácilmente autenticable.
- Es fácilmente divisible.
- Es fácil de transportar.
- Nos permite hacer liquidación de pagos facilmente.
- Es "escaso".
- Es internacional.
- Preserva o incrementa su valor en el tiempo.
- No es volátil.
- Es fungible.
- No decae en el tiempo.
- Tiene una amplia base de usuarios.
- Es líquido.
- Es fácil de guardar de manera segura.
- Es anónimo.
- Es descentralizado.
- Es "inservible".
- Es antifrágil.
- Puede ser usado en canales inseguros.

"•" ¡El buen dinero es programable!

Traits of Money	Gold	Fiat (US Dollar)	Crypto (Bitcoin)
Fungible *(Interchangeable)*	High	High	High
Non-Consumable	High	High	High
Portability	Moderate	High	High
Durable	High	Moderate	High
Highly Divisible	Moderate	Moderate	High
Secure *(Cannot be counterfeited)*	Moderate	Moderate	High
Easily Transactable	Low	High	High
Scarce *(Predictable Supply)*	Moderate	Low	High
Sovereign *(Government Issued)*	Low	High	Low
Decentralized	Low	Low	High
Smart *(Programmable)*	Low	Low	High

2.4. **Historia moderna del dinero:**

- El dinero tradicional desde la antigüedad han sido los metales preciosos, hasta el siglo XIX.
- Competición bimetálica oro-plata, hasta final del siglo XIX.
- El dinero papel se generaliza a finales del siglo XIX.
- Patrón oro: dinero papel cambiable por oro.
- 1933: prohibición en USA de la tenencia del oro por particulares.

2.5. **Hegemonía del dinero fiat:**

- Al acabar la II Guerra Mundial, USA se hace con las reservas de oro del mundo.
- Época del patrón dólar. El dólar es cambiable por oro por los bancos centrales.
- 1971 Nixon shock: se acaba la convertibilidad del dólar por oro.
- Patrón Fiat: las monedas nacionales no están respaldadas por oro.
- Masa monetaría controlada y regulada por los Bancos Centrales.

2.6. **Bimetalismo-Ley de Copérnico-Gresham:**

- Bimetalismo: coexistencia del oro y la plata. Tensiones monetarias.
- Ley de Copérnico-Gresham: el dinero malo expulsa al dinero bueno del mercado.
- Se gasta el dinero malo y se atesora el dinero bueno.
- Durante el siglo XIX: arbitraje entre el Banco de Inglaterra y el Banco Central de Francia.

2.7. **Descentralización:**

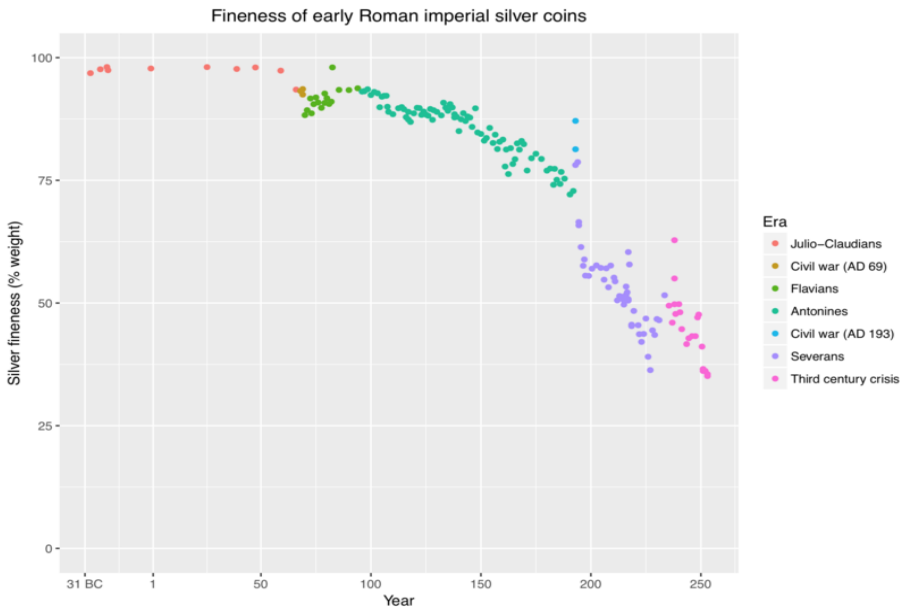
- Moneda descentralizada: moneda libre cuya emisión no está controlada por nadie.
- Ejemplo histórico: el oro físico.
- Ejemplo semi-descentralizado: moneda histórica de oro con valor facial.
- Señal de decadencia de los imperios: envilecimiento de la moneda.

2.8. **Señoreaje:**

- Es el beneficio que se obtiene al acuñar moneda.
- El poder (reyes, Estados) imponen su moneda y se benefician del señoreaje.
- Ejemplo actual: billetes.
- ¿Cuál es la diferencia entre un billete de 500 pesos y un billete de monopoly?

2.9. **Problemas de la centralización:**

- El valor de la moneda depende de la política monetaria.
- El aumento de la masa monetaria provoca inflación y devalúa la moneda.
- El aumento descontrolado de la masa monetaria provoca hiperinflación.



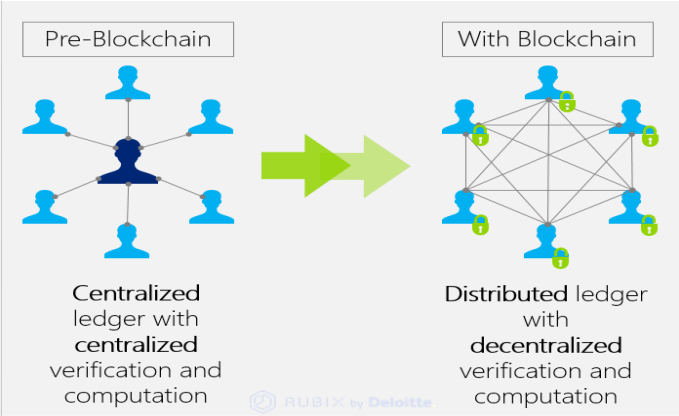
2.10. **Problema de los Generales Bizantinos.** El problema de los generales bizantinos es una analogía de un escenario de guerra; es el problema abstracto de obtener consenso entre entidades que interactuan por un objetivo común. En nuestro contexto es el problema de cómo evitar un doble gasto sin necesidad de un tercero de confianza. La primera solución la dió Satoshi Nakamoto generando un consenso descentralizado, y lo hizo en tres partes (ver [Al]):

1) El registro de los votos debía poder ser consultado por todos los Generales (ledger público).
2) El voto debía fundarse en una "prueba de unicidad". Como no podía tratarse de una simple firma sobre un pergamino (demasiado fácil de falsificar). Cada general debería resolver un criptograma cuya solución sería su propia firma (proof of work). Pero como el criptograma necesita un tiempo determinado para poder resolverse, Satoshi agregó que el proceso de votación debía tener una duración limitada –que precisamente equivaldría al tiempo que se necesita para resolver un enigma– de manera que sería imposible resolver dos y, por lo tanto, "firmar" dos mensajes contradictorios.
3) Para asegurarse de que efectivamente todos los generales poseían el mismo registro que contabilizaba los votos obligó a cada General a "encadenar" su voto con el del siguiente, de modo que toda modificación de un elemento de la cadena modifica el aspecto general del registro.

3. Protocolo de la red Bitcoin

3.1. Ledger.

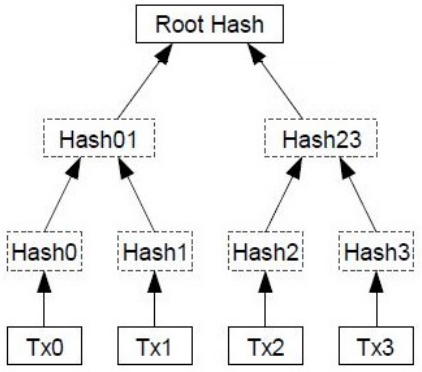
- Todos los usuarios de la red mantienen una copia del registro de todas las transacciones ejecutadas (libro de cuentas global o *ledger*), denominada **blockchain**.



- La puedes bajar en tu computadora.
- Ocupa a día de hoy unos 460GB.
- Al bajártela compruebas la coherencia de la misma. Si está corrompida no la usarías.
- El saldo de una cuenta se mira con el histórico de transacciones.
- Toda la información queda para la eternidad.

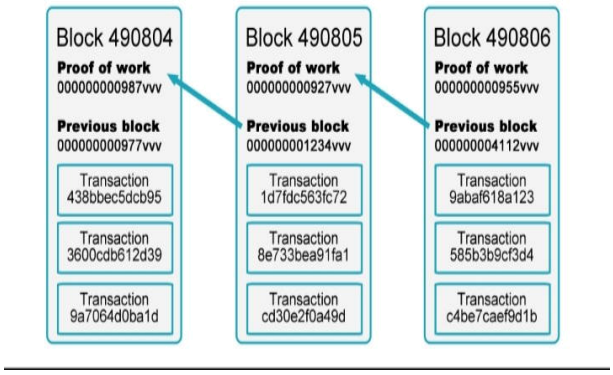
3.2. Árbol de Merkle y Blockchain.

- Es un árbol de bloques (o más bien de hashes), donde cada bloque referencia al anterior.
- Patentado en 1979 por Ralph Merkle.
- El bloque inicial se llama *root* o *génesis*.
- Cada hoja (bloque que sigue a otro) contiene el hash del anterior... y por lo tanto de todos los anteriores.
- El hash es una firma criptográfica.
- Si se cambia un bit de un bloque, cambia su hash, y por lo tanto la información del siguiente y todos los hashes de todos los que le siguen.
- Es decir, se detectaría un cambio de un bit porque hay una discordancia de hashes que no puede ser fácilmente arreglable.



Blockchain:

- El árbol de Bitcoin se llama cadena de bloques (blockchain).
- Cada bloque contiene:
  - el hash del anterior,
  - un número de transacciones,
  - un *nonce*: campo de 32 bits de tal manera que el hash de todo el bloque comienza con muchos ceros.
  - y unos cuantos bitcoins creados de la nada (al día de hoy 6.25 Btc.): que se reparten proporcionalmente (a su inversión de computo) entre los nodos que han conseguido hallar el nonce.



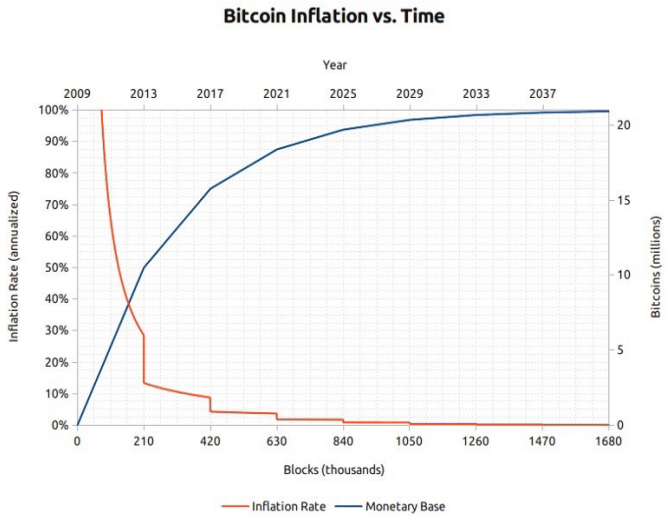
- Cada bloque ocupa 1 MB y contiene un máximo de 2000 transacciones.
- Se genera un bloque cada 10 minutos.
- El nonce se ajusta para que el bloque cumpla una propiedad (su hash sea "especial") y es difícil de hallar.
- Los nuevos bitcoins creados se asignan a la cuenta del (de los) minero(s) que construyó el bloque.

Confirmaciones:

- Si un bloque tiene un bloque sucesor, se dice que tiene dos confirmaciones (él mismo es una confirmación).
- Si hay varios bloques, tiene varias confirmaciones.
- Un bloque con confirmaciones es difícilmente modificable. Esto supone modificar su nonce, y por tanto el hash del siguiente, y por tanto también su nonce y así sucesivamente. Los hashes son fáciles de hacer, pero los nonces son difíciles de encontrar.
- Por tanto, comprobar la coherencia de un bloque (o de toda la blockchain) es fácil, pero modificar un bloque es muy difícil. Y si está confirmado muchas veces es *imposible*.
- Se considera que 3 confirmaciones dan garantía, y 6 confirmaciones son prácticamente seguridad total.

Creación de bitcoins:

- El primer bloque generó los primeros 50 bitcoins.
- Los siguientes bitcoins se generan paulatinamente:
  - 210,000 bloques (2009-2012). 50 Btc por bloque. Total 10,5 MBtc.
  - 210,000 bloques (2013-2016). 25 Btc por bloque. Total 5,25 MBtc.
  - 210,000 bloques (2017-2020). 12,5 Btc por bloque. Total 2,625 MBtc.
  - 210,000 bloques (2021-2024). 6,25 Btc por bloque. Total 1,3125 MBtc.
  - etc.



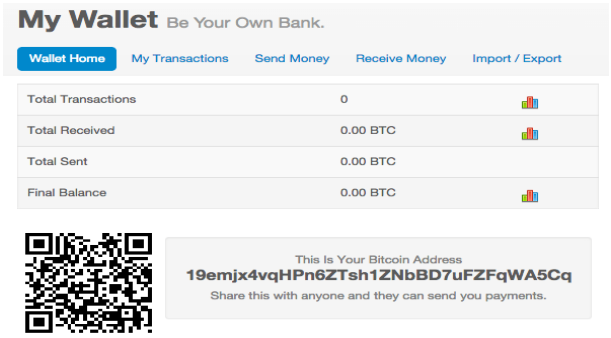
- Es una progresión geométrica:

$$210,000\left(50+\frac{50}{2}+\frac{50}{2^2}+\cdots+\frac{50}{2^{n-1}}+\cdots\right)=10,500,000\sum\frac{1}{2^n}$$

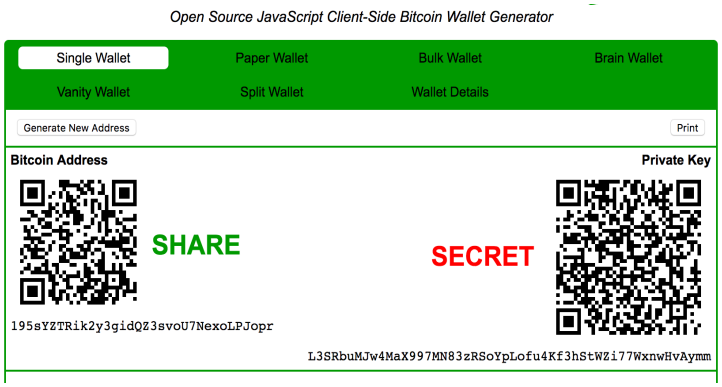
  con límite 21 MBtc. Hoy en día hay 19,068,362.50 Btc, 90.9% del total.
- Obsérvese que 1 bloque cada 10 minutos da $24\times 6\times 365=52,560$ bloques por año, y por lo tanto en 4 años hay 210,000 bloques.
- Desde mayo 2020 la producción de Btc (inflación) es menor al 2%, menor que la de las monedas fíat e incluso que el oro.
- Al cambio de recompensa en Btc se le llama *halving*.
- La recompensa de Btc baja, pero su capitalización sube, luego puede salir ventajoso.
- Los bitcoins generados van a parar a los mineros (que sustituyen a los bancos).

Cuentas Bitcoin:

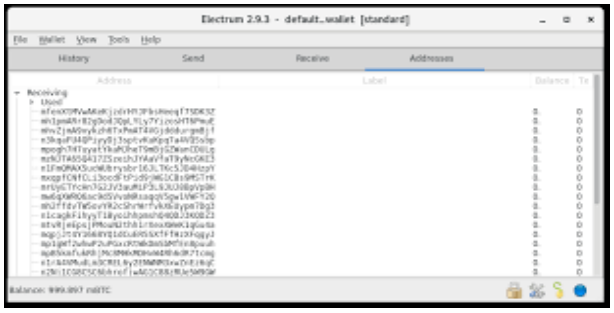
- Un usuario de bitcoin genera una clave pública y una clave privada con un protocolo criptográfico. Esto es una cuenta. `https://www.bitaddress.org`
- La clave pública (dirección bitcoin) es del tipo: 18Ry3k3K6NiTCGNxVNFi6zmmSjvZU4uqCx



- Se recibe dinero con la clave pública pero sólo se puede enviar con la privada.
- Quien posea la clave privada, posee el dinero.
- La clave privada es de la forma:
- Hay 2^{256} posibles direcciones (hay 2^{270} átomos en el universo). Cada usuario puede usar varias, para aumentar la seguridad (Nota: mejor no reusar!)



  - Si se pierde una clave privada, se pierden los fondos (desaparecen para siempre).
  - Si te roban la clave privada, se quedan con los Btc.
  - Recuerda: ¡no está respaldado por bancos o gobiernos!
- Un monedero (wallet) es un pequeño programa (Electrum, Milibit, ...) que genera una colección de cuentas (claves privadas y públicas). Va protegido por un password personal.



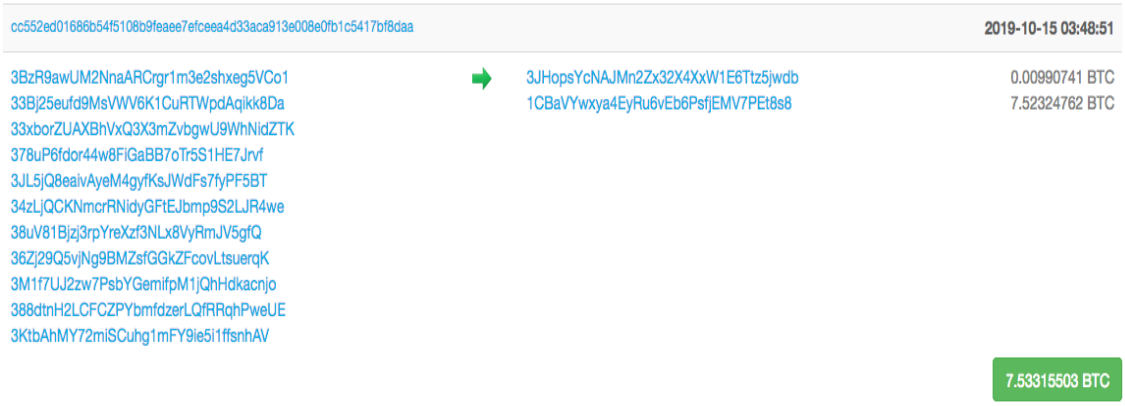
- El wallet puede tener una regla mnemotécnica que es un código generador. Ejemplo: **scissors motor tree car sky supreme cabin mixed outside real able**.

Tipos de monederos:

- Full client. Te descargas toda la blockchain. Actúas como nodo repetidor.
- Light client. Te conectas a un nodo. Descargas un pequeño wallet.
- Web client. Te conectas a una web que ofrece un wallet.
- Monederos Hardware wallet.
- Cold storage wallet. Monedero papel.
- Tarjeta de débito en bitcoins

Transacciones:

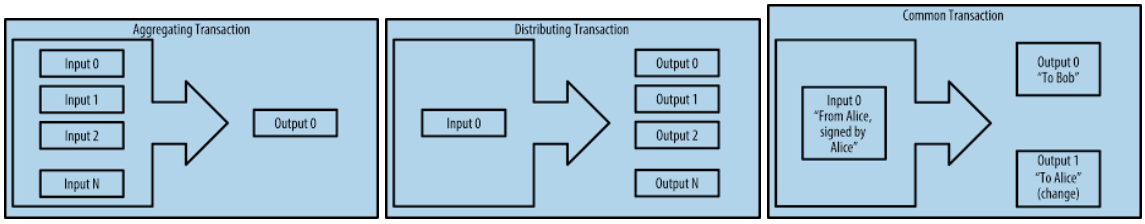
- Las transacciones son de la forma:
  - 1.25 Btc
  13WCbQepQrZr5g5DNRfvsWBwSdcEyhbdGV → 1PH5SniVxd2KchXZzEhjzrjJvHFuuRVkKQ
  - Firma (con clave privada de 13WCbQepQrZr5g5DNRfvsWBwSdcEyhbdGV).
- La unidad de monedas más usada ahora es el milibitcoin (mBtc).
- De hecho las transacciones suele ser:



- Transacciones en

  `https://www.blockchain.com/es/btc/unconfirmed-transactions`

- El saldo de una cuenta se averigua leyendo toda la blockchain y viendo cuántos Btc recaen en ella (recibidos-emitidos).
- Las transacciones tienen un identificador (id), que es de hecho la firma criptográfica.
- Cada cuenta se parece más de hecho a una moneda (coin).
- Técnicamente, las transacciones tienen como input otra (u otras) transacción (de donde vienen los fondos que se usan *íntegramente*).
- Transacción de agregación: como cambiar morralla por un billete.
- Transacción de distribución: se reparten fondos entre varias cuentas.
- Transacción usual: se paga algo y se recoge el cambio.



Broadcasting y comisiones:

- Las transacciones, una vez firmadas se envían a la red (broadcast).
- Son irreversibles (no hay autoridad central, ¿recuerdas?).
- En la red, se transmiten de nodo a nodo hasta que un minero las incluye en un bloque.
- Las confirmaciones pueden tardar en momentos de mucho tráfico (entre 10 minutos y varios días) Si una transacción no entra, desaparece a los varios días (a menos que se haga re-broadcast).

- En la transacción se puede añadir una comisión (fee) voluntatia, que va destinada al minero. Los mineros suelen elegir transacciones con fee. Por tanto, una fee alta mejora las posibilidades de que sea ejecutada rápido.
- La fee depende del tamaño en bits de la transacción, no del importe que se transfiera!
- Tarifas de 20 satoshis/byte son estándar bitcoinfees.earn.com (aprox 0,1 mBtc/transacción, 500 bytes por transacción).
- Ha habido transacciones de $150 millones por 0 Btc fee.

Búsqueda de transacciones en la Blockchain:

- Las transacciones pendientes se ven en `https://www.blockchain.com/es/btc/unconfirmed-transaction`.
- Se pueden buscar transacciones realizadas y saldos de cuenta en `http://www.blockchain.com`

3.3. Minería y ajuste de dificultad.

- Las nuevas transacciones "flotan" en la red. Algunos usuarios, llamados *mineros*, recogen las transacciones y las escriben en un nuevo bloque.
- Cualquiera puede ser minero (open source code).
- El minado requiere gasto en electricidad.
- El minero añade el hash del último bloque junto con las nuevas transacciones, comprobando que todos los gastos están firmados, que tienen fondos y que no están duplicados.
- El minero debe encontrar el *nonce* para que el bloque le sea admitido en la blockchain. El nonce es difícil de encontrar (proof-of-work).
- El minero suertudo que gana el *bingo* de encontrar el nonce puede transmitirlo al resto de los nodos.
- Se premia al minero con la generaión de nuevos bitcoins (hoy en día 6.25 Btc por bloque generado). Además el minero cobra las comisiones de las transacciones (voluntarias).
- Los mineros se suelen agrupar en *pools* de mineros. Si un pool mina un bloque, se distribuye la recompensa entre todos los miembros proporcionalmente al hash-power aportado.
- Los últimos bloques minados en `https://www.blockchain.com/btc/blocks`

Altura	Hora	Resuelto por	Hash	Tamaño (kB)
599419 (Cadena principal)	2019-10-15 04:06:54	Poolin	00000000000000000000ea3399c707eaa3f93632ed08e4ffe95855518a5ebaf0a	1,166.32
599418 (Cadena principal)	2019-10-15 03:55:18	F2Pool	00000000000000000000bb3fd78ff1b63237f563cc81e88ca8698c444eb639646	276.43
599417 (Cadena principal)	2019-10-15 03:52:16	Unknown	00000000000000000000091020000a5aa19c2832cef5fc8948eac2ace1ebfcdc59	1,042.94
599416 (Cadena principal)	2019-10-15 03:46:47	F2Pool	00000000000000000000e12dbc54d742ba7dd6cdba45d5839f10454df4bcb6fa7	1,219.42

- Puede ocurrir que varios mineros minen un bloque casi simultáneamente (obviamente las transacciones elegidas por cada uno pueden ser distintas)
- En ese caso, los demás mineros elegirán uno de los bloques para seguir trabajando. Eventualmente una de las cadenas formadas será más larga y la otra será olvidada.

Ajustando la dificultad:

- Se produce el minado de un bloque aproximadamente cada 10 minutos (aunque muy variable). Ver gráfica anterior
- Cada bloque contiene unas 2000 transacciones, y con 1MB de tamaño máximo (500 bytes por transacción).
- Cada 2106 bloques (unas 2 semanas), se reevalúa la potencia del hash de la red (depende del número de mineros y de la potencia de las CPU y GPU de sus computadoras).
- Se ajusta la dificultad de la proof-of-work (es decir, encontrar el nonce) para que con la potencia global de la red se tarde aproximadamente 10 minutos en resolver el minado.

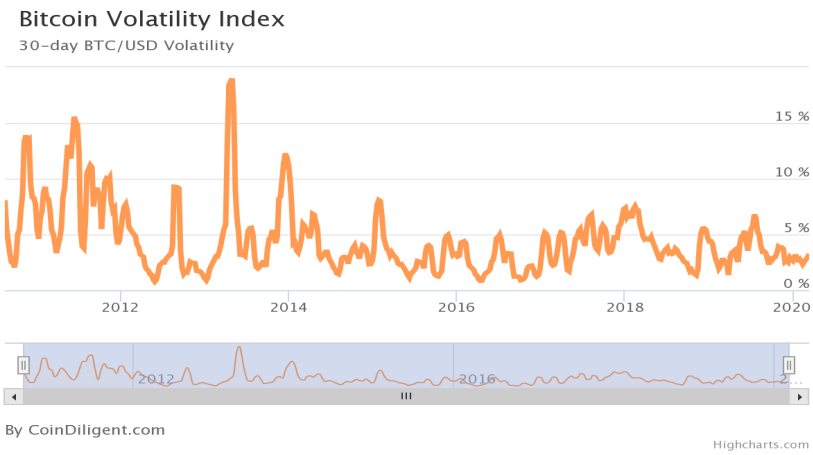
- Fue elegido 10 minutos porque es una estimación de lo que tarda la red en circular por todo el globo en una red P2P.

Crecimiento del Bitcoin:

- Valor actual de mercado: US$215,164,257,322 (30 agosto 2020).
- Masa monetaria mayor que Perú, menor que Colombia (equivale al país 56 del mundo).
- Mercado global de divisas: $ 80 billones.
- Precio actual: 1 Btc = US$ 11,646
- Es previsible un crecimiento de la valoración por adopción (la demanda crece y la oferta es fija).

Volatilidad del Bitcoin:

- La volatilidad del Bitcoin es alta porque la oferta es totalmente inflexible y escasa. El precio responde directamente a la demanda, además de ser usada más como reserva a futuro (el futuro es impredecible).
- En el momento que haya una adopción completa, la volatilidad será baja dado que no se pueden crear nuevos bitcoins y por su uso "cotidiano" para intercambiar bienes y deuda.
- La distribución de la riqueza sigue el *principio de Pareto* al estabilizarse (el 80% de la riqueza de una sociedad se concentra en el 20% de su población). Son habituales los vaivenes de precio antes de llegar a ese punto.
- Volatilidad media de Bitcoin y dolar:



Comparativa con otros medios de pago:

- Visa realiza 3,200 operaciones por segundo, 100,000 millones al año (con un máximo de 65.000 transacciones por segundo)
- Bitcoin realiza 4 transacciones por segundo, 350,000 al día, 210 millones al año (con un máximo de 20 por segundo).
- Hay al rededor de 25 millones de monederos Bitcoin, mientras que el número de tarjetas de crédito y débito (Visa y MasterCard) ha alcanzado los 5,300 millones.
- Bitcoin tiene unos 10,000 nodos activos en comparación con los 119 centros de datos de Visa.
- Visa tarda 1-3 días en confirmar los pagos. Bitcoin tarda 1 hora-1 día (en épocas de atasco).
- Visa funciona con un límite de gasto (como un prepago, con un seguro de custodia bancaria).
- Las transacciones Bitcoin son finalistas, sin seguro ni revocabilidad.
- Tarjetas Bitcoin combinan ambas características.

El Bitcoin es lento e ineficaz pero no necesita control ni confianza en un gobierno.

3.4. **¿ Cómo comprar o vender Bitcoin?** Se requiere tener abierto un wallet.

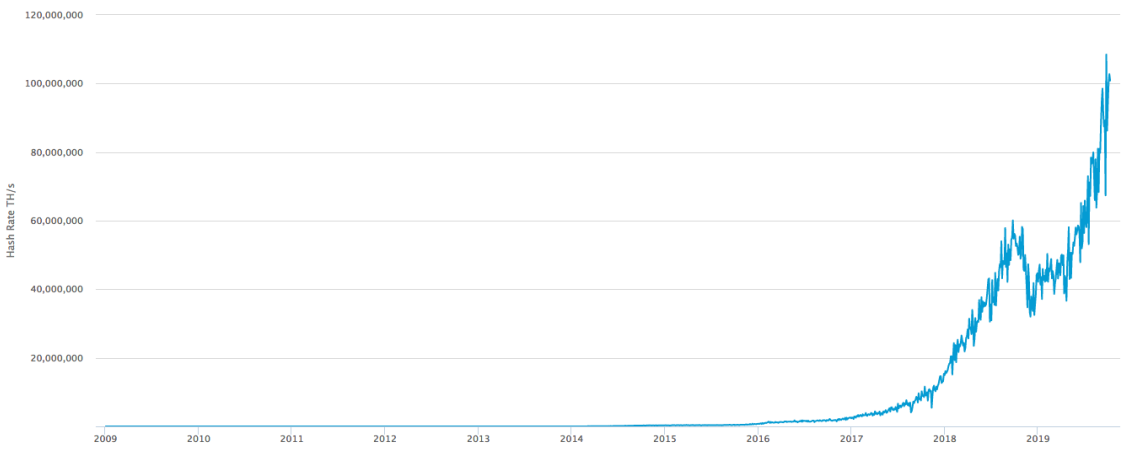
- Vender algo o realizar un trabajo que te paguen en bitcoin.
- Puedes transferir bitcoin entre dos cuentas tuyas.
- Puedes pagar en bitcoin a cambio de un producto (comprar). Hay tiendas físicas y en internet que aceptan bitcoin.
- Puedes pagar en bitcoin a cambio de dinero fíat (o al revés). Es decir, comprar moneda.
- Los exchanges son casas de cambio en internet y se puede comprar o vender bitcoin. Ejemplos: Bitstamp `https://www.bitstamp.net`, Kraken `https://www.kraken.com` o Bisq `https://bisq.network`.
  - Hay que darse de alta, requieren identificación como un banco y siguen regulaciones estatales (KYC: Know Your Costumers). Bisq es descentralizado y no es necesario un ID.
  - Puedes convertirte en un bróker.
  - Hay que pagar a hacienda por lo beneficios.
- Encontrar a alguien a quien comprar bitcoin en `https://localbitcoins.com`.
- ¡Hazte minero!

Gasto de energía

- El gasto de energía por minería Bitcoin mundial es de unos 22 TWh al año.
- Supone el 0,15% de la energía mundial consumida.
- Corresponde al gasto total de Ecuador (país 70 en gasto energético).
- Corresponde con el gasto en lavadoras de EU.
- Gasto en luces navideñas EU = gasto 1 mes de minado Bitcoin.
- La minería de metales preciosos consume 20 veces más que Bitcoin.
- No obstante, hay que pensar en el gasto del sector bancario.

Potencia de hash

- La potencia de hash o *hashrate* actual es de 75.000.000 TH/s.
- Equivalente a 2 billones de PC's (2 millones más que la computadora más superpotente).
- Se puede encontrar en `https://www.blockchain.com/es/charts/hash-rate`



- El hashrate aumentó mucho con la introducción de las GPU y con las tarjetas ASIC (Application-Specific Integrated Circuit)
- Hay granjas hash en países con energía barata (China, Singapur), o junto a fuentes energéticas (Islandia). 75% de la energía Bitcoin es renovable.

Rentabilidad de la minería:

- La minería tiene unos gastos:

  - Adquirir las tarjetas de minado (GPUs)
  - Gasto de electricidad.
  - Pago a hacienda como actividad industrial.
- Hay que adquirir una tarjeta con hashrate alto (más cara), que quedará obsoleta en un tiempo.
- Con el tiempo la electricidad sube de precio.
- El hashrate global de la red también sube (el rédito de minar baja).

Rentabilidad del Bitcoin:

- La rentabilidad media de comprar Bitcoin y esperar es del 70% anual, descontando burbujas.
- Puede ser más rentable comprar y esperar que minar.

Cotización del Bitcoin:

- La primera compra realizada fue de dos pizzas por 10.000 Btc en mayo 2010.
- La primera cotización de Btc en un Exchange fue de 1 Btc= $0,000994 en octubre 2009 calculado mediante el coste de electricidad que se necesitaba para producir un bitcoin.
- Las *órdenes de compra y venta* fijan el valor del cambio.
- El valor del cambio es el de la última compra-venta realizada.

4. Mecanismo criptográfico (protocolo matemático)

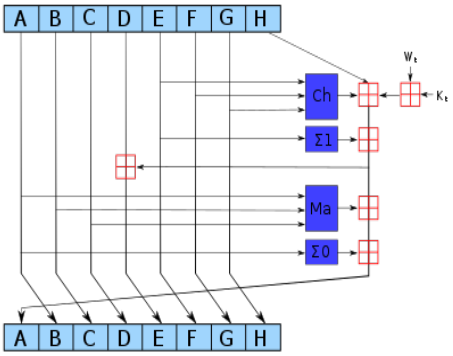
4.1. **Funciones Hash.**

- Una función hash H es una función computable mediante un algoritmo $x \to H(x)$ que tiene como entrada un archivo y lo convierte en una cadena fija y pequeña de bits.
- Barata: se calcula fácilmente.
- Compresión (digest)
- Sirven como identificadores de documentos, difícilmente corruptibles.
- Uniforme (equi-probabilidad).
- Determinista (pseudo-aleatoria).
- No reversible.

SHA (Secure Hash Algorithm):

- Familia de funciones Hash de cifrado publicadas por el NIST (National Institut of Standards and Technology USA).
- SHA-2. Tamaño de salida: 224 a 512 bits.
- Tamaño máximo del mensaje de entrada: 2^{64} o 2^{128} bits
- Operaciones: $+$, and, or, xor, shr, not.
- SHA224 (“The quick brown fox jumps over the lazy dog”)
  0x730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525
- SHA224 (“The quick brown fox jumps over the lazy dog.”)
  0x619cba8e8e05826e9b8c519c0a5c68f4fb653e8a3d8aa04bb2c8cd4c

Mecanismo del SHA-2
Tamaño del bloque: 512 a 1024
Número de iteraciones: 64 a 80
Una iteración en la función de compresión de la familia SHA-2
SHA-256 divide el mensaje en bloques de 8×32 bits.



Los componentes lila representan las siguientes operaciones:
$\text{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$
$\text{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$
$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$
$\boxplus$ significa suma módulo 2^{32}.

Criptografía: La criptografía ha servido, desde la antigüedad, para transmitir mensajes sin que puedan ser leídos, incluso si caen en manos del enemigo.

- Clave del César: desplazar el alfabeto k unidades
  ABCDEFGHIJKLMNNOPQRSTUVWXYZ
  KLMNNOPQRSTUVWXYZABCDEFGHIJ
- Alfabeto: $\mathbf{Z}_n = \{0, ..., n-1\}$.
- Operación: $+$
- Clave: k
- Encriptación: $x \to x + k$
- Desencriptación: $x \to x - k$

FIGURE 3. Enigma

4.2. **Aritmérica modular.**

- Aritmética del reloj: aritmética módulo 12, o módulo 24.
- $7 + 11 = 6 \pmod{12}$.
- Se puede restar, de hecho hay números negativos: $-1 = 11 \pmod{12}$.
- $\mathbf{Z}_{12} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ tiene 12 elementos.
- $\mathbf{Z}_n = \{0, 1, 2, ..., n-1\}$ tiene n elementos.
- También hay producto: $7 \cdot 11 = 77 = 12 \cdot 6 + 5 = 5 \pmod{12}$

División módulo n:

- La ecuación $a \cdot x = b$ tiene solución $x = b/a$
- El inverso satisface $a \cdot a^{-1} = 1$
- Dividir es multiplicar por a^{-1}, es decir, $x = a^{-1} \cdot b$

$$a \cdot x = b \Rightarrow a^{-1} \cdot a \cdot x = a^{-1} \cdot b \Rightarrow x = a^{-1} \cdot b$$

- En aritmética módulo n la situación es la siguiente:
  a) $5^{-1} = 5 \pmod{12}$, porque $5 \cdot 5 = 25 = 1 \pmod{12}$
  b) 4 no tiene inverso: Si $x = 4^{-1}$, entonces $4 \cdot 3 \cdot x = 0 = 4 \cdot 3 \cdot 4^{-1} = 3 \pmod{12}$!!!
  c) De hecho, sólo los números en rojo: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$ tienen inverso.
  d) Los números invertibles a satisfacen que $\text{mcd}(a, n) = 1$, es decir, a, n son coprimos.
- Si p es primo, todos los números $1, 2, ..., p-1$ son invertibles.
- $\mathbf{Z}_p^* = \{1, 2, ..., p-1\}$

Cálculo de inversos modulares:

- De forma práctica, hallar el inverso es resolver $a \cdot x = 1 \pmod{n}$, es decir, $a \cdot x + b \cdot n = 1$.

- Ecuación diofántica para hallar el mcd. Algoritmo de Euclides.
- Ejemplo: Inverso de $5 \pmod{103}$, es decir, $5 \cdot a + 103 \cdot b = 1$
  - División euclídea:
  - $103 = 20 \cdot 5 + 3$
  - $5 = 1 \cdot 3 + 2$
  - $3 = 1 \cdot 2 + 1$
  - $1 = 3 - 1 \cdot 2 = 3 - 1(5 - 1 \cdot 3) = 2 \cdot 3 - 1 \cdot 5 = 2(103 - 20 \cdot 5) - 1 \cdot 5 = 2 \cdot 103 - 41 \cdot 5$
  - Por tanto, $1 = -41 \cdot 5 \pmod{103}$, es decir, $5^{-1} = -41 = 62 \pmod{103}$.

Potencias módulo n: Cálculo eficiente de $a^b \pmod{n}$.

Método 1. Buscar recursividad:

- Ejemplo: $2^{100000} \pmod{13}$. Las potencias sucesivas de 2 son
- $2^0 = 1, 2^1 = 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 1, 2, ...$
- Se repiten cada 12, luego basta calcular el residuo, $2^{100000} = 2^{8333 \cdot 12 + 4} = 2^4 = 3 \pmod{13}$.
- Euler: Si p es primo, entonces $a^{p-1} = 1 \pmod{p}$.

Método 2. Ir haciendo cuadrados:

- Ejemplo: $2^{100} \pmod{103}$. Los cuadrados sucesivos de 2 son
- $2^2 = 4$, $2^4 = 12$, $2^8 = 256 = 50$, $2^{16} = 2500 = 28$, $2^{32} = 784 = 63$, $2^{64} = 3969 = 55 \pmod{103}$
- $2^{100} = 2^{64+32+4} = 55 \cdot 63 \cdot 16 = 55440 = 26 \pmod{103}$.

Encriptación con potencias modulares:

Alfabeto: $\mathbf{Z}_p^* = \{1, 2, ..., p-1\}$

- Operación: potencia
- Clave: k (exponente)
- Encriptación: $M \to m = M^k$
- Desafortunadamente, para p primo, la desencriptación es muy fácil:

- Sea $k \cdot k' = 1 \pmod{p-1}$
- Entonces $m \to M = m^{k'}$ desencripta:

$m^{k'} = (M^k)^{k'} = M^{k \cdot k'} = M^{s(p-1)+1} =$
$= (M^{p-1})^s \cdot M = M$

- Operación: exponenciación
- Clave: g (base)
- Encriptación: $M \to m = g^M$
- Desencriptación. El inverso de hacer potencias es el *logaritmo modular*: dabo b hallar x tal que $g^x = b \pmod{n}$
- Ejemplo: $n = 29$, $g = 5$, $x \to g^x$

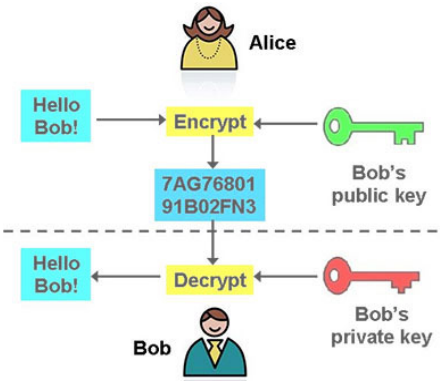
$x = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,$
$= 21, 22, 23, 24, 25, 26, 27, 28.$

$g^x = 5, 25, 9, 16, 22, 23, 28, 24, 4, 20, 13, 7, 6, 1, 5, 25, 9, 16, 22, 23,$
$= 28, 24, 4, 20, 13, 7, 6, 1.$

- No hay forma eficiente de calcular el logaritmo modular.

Criptografía de clave pública:

- Cada computadora (A) genera una clave pública (PU-A) y una clave privada (PR-A).
- La clave privada sirve para encriptar mensajes. La clave pública sirve para desencriptarlos.
- Cada usuario envía su clave pública a la red para que todas las computadoras en red la oigan.
- Alicia quiere enviar un mensaje M a Bob. Toma la clave pública de Bob y encripta m=Enc(M,PU-B).
- Envía el mensaje m a Bob por internet.
- Bob lee el mensaje y lo desencripta Des(m,PR-B)=M usando su clave privada.
- Sólo Bob podrá desencriptar y leer ese mensaje.
- Para evitar ataques (hackers) debe ser casi imposible averiguar PR-B



Encriptación RSA

- Función de Euler: $\Phi(n) =$ cantidad de números coprimos con n (módulo n).
- Para $n = 15$, tenemos $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14$. Luego $\Phi(15) = 8$.
- Si $n = p \cdot q$, con p y q primos, tenemos que $\Phi(n) = p \cdot q - (p + q - 1) = (p - 1)(q - 1)$.
- Euler: Sea a coprimo con n, entonces: $a, a^2, a^3, ...$ son coprimos con n. Al repetirlo $\Phi(n)$ veces, tenemos $a^{\Phi(n)} \equiv 1 \pmod{n}$

Generación de claves:

- Bob: elige dos números primos p y q, grandes y de forma aleatoria.
- $n = p \cdot q$.
- $\Phi(n) = (p - 1)(q - 1)$
- Clave pública: se elige e coprimo con $\Phi(n)$. Se dan a conocer e, n.
- Clave privada: se determina d con $e \cdot d \equiv 1 \pmod{\Phi(n)}$.

Encriptación y desencriptación:

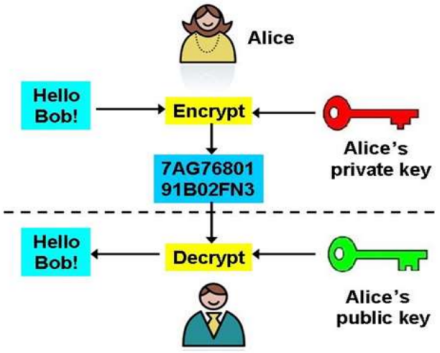
- Alicia: sea M el mensaje. Sean e, n las claves PU-Bob.
- Alicia encripta y envía: $m \equiv M^e \pmod{n}$.
- Bob recibe m y desencripta, usando PR-Bob (es decir, d).
- $m^d \equiv M^{ed} \equiv M^{1+k \cdot \Phi(n)} \equiv M \cdot (M^{\Phi(n)})^k \equiv M \pmod{n}$.

Seguridad:

- Dos números primos p y q grandes.
- Clave pública: $n = p \cdot q$, e.
- Clave privada: d con $e \cdot d \equiv 1 \pmod{\Phi(n)}$ donde $\Phi(n) = (p - 1)(q - 1)$.
- Para encontrar la clave privada, necesitamos conocer p y q, es decir factorizar n. Problema imposible computacionalmente.
- RSA-768 (232 dígitos), el mayor en ser factorizado lleva 15000 años-CPU de cálculo.
- Se suele usar RSA-1024 o RSA-4096.

Firma digital:

- Cada usuario (A) genera una clave pública (PU-A) y una clave privada (PR-A).
- La clave privada sirve para encriptar mensajes. La clave pública sirve para desencriptarlos.
- Cada usuario envía su clave pública a aquellos con quien quiera comunicarse (en Bitcoin a toda la red).
- Alicia quiere enviar un mensaje M a otro usuario, o bien a toda la red. Encripta m=Enc(M,PR-A).
- El receptor puede leer el mensaje. Desencripta M=Des(m,PU-A), además sabe que proviene de A (firma digital).
- Para evitar suplantaciones debe ser imposible adivinar PR-A si conocemos PU-A.



Firma digital (Digital Signature Algorithm DSA):

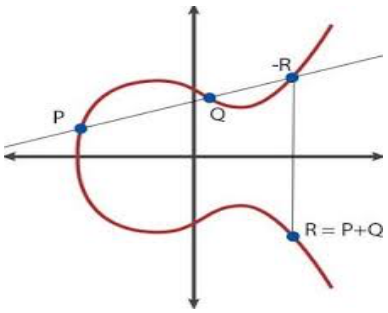
- Generación de claves
  - Se elige un primo q de 160 bits.
  - Se toma g entre 1 y $q - 1$.
  - Se calcula $y = g^x$
  - Clave pública: (q, g, y).
  - Clave privada: x.
- Firma
  - Se elige aleatoriamente k entre 1 y $q - 1$.

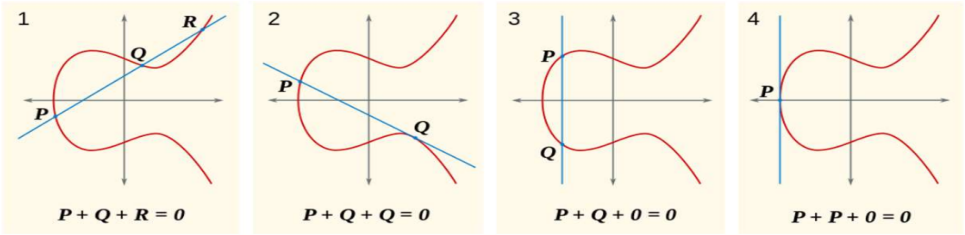
  - $r = g^k \pmod q$.
  - $s = k^{-1} \cdot (H(M) + r \cdot x)$, donde $H(M)$ es el hash SHA1 del mensaje M.
  - Mensaje: M
  - Firma: (r, s).
- Verificación
  - Calculamos $w = s^{-1} \pmod q$
  - Calculamos $u = H(M) \cdot w \pmod q$
  - Calculamos $v = r \cdot w \pmod q$
  - Sea $t = g^u \cdot y^v$
  - Verificado $t = r$
- Demostración:
  - $w \cdot (H(M) + r \cdot x) = w \cdot k \cdot s = k \pmod q$
  - $t = g^u y^v = g^u \cdot g^{x \cdot v} = g^{u+x \cdot v} = g^{H(M) \cdot w + r \cdot x \cdot w} = g^k = r$
- En la práctica
  - Se suele fijar (q, g).
  - Firmar es $x \to g^x$.

4.3. **Curvas elípticas.**

- Curva elíptica: $y^2 = x^3 + ax + b$, con a, b dados y todos los números en aritmética modular $\mathbf{Z}_p$ con p un primo.
- Hay una suma geométrica: Si P, Q son dos puntos en la curva, la recta que pasa por P y Q corta a la curva en un tercer punto $-R$. Sea R el punto simétrico. Se declara

$$P + Q = R$$





Criptografía de curvas elípticas:

- Sean $P = (x_1, y_1)$, $Q = (x_2, y_2)$ y $R = P + Q = (x_3, y_3)$.
  - $x_3 = \lambda^2 - x_1 - x_2$
  - $y_3 = \lambda \cdot (x_1 - x_3) - y_1$
  - $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
- Los puntos (x, y) de la curva con x, y en $\mathbf{Z}_p$ tienen la operación suma de puntos.
- $kP = P + P + \cdots + P$ (k veces).
- Generación de claves:
  - Se elige una curva eliptica y un primo p. Se elige un punto base G en la curva.
  - El orden $n > 0$ es el mínimo que verifica $nG = 0$.
  - Clave privada: un número k entre 1 y $n - 1$
  - Clave pública: el punto $K = kG$.

Elliptic Curve Digital Signature Algorithm (ECDSA):

- En Bitcoin se elige la curva elíptica

$$y^2 = x^3 + 7$$

- Sobre $\mathbf{Z}_p$, con el primo $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ (de 255 bits)
- Punto base $G = (g_x, g_y)$ con
  - $g_x = 55066263022277343669578718895168534326250603453777594175500187360389116729240$
  - $g_y = 32670510020758816978083085130507043184471273380659243275938904335757337482424$
- Clave privada: k, número de 256 bits, 32 bytes, 64 hex.
  - Se elige de forma aleatoria.
  - El número claves privadas es de 1.158×10^{77}, es un poco menor que 2^{256} (hay 2^{270} átomos en el universo).
- Clave pública: se toma $kG = (x, y)$. Se concatena para formar $K = 04xy$, número de 520 bits, 65 bytes, 130 hex.

Clave pública y clave privada:

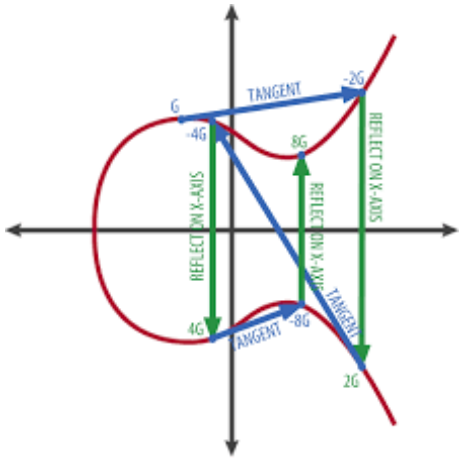
- Ejemplo: clave privada

$$k = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD$$

- Clave pública es $K = kG$

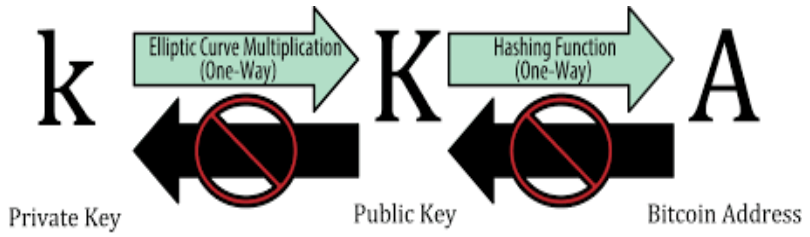
$$K = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD * G$$

- $K = (x, y)$
  $x = F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A$
  $y = 07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB$
- $K = 04F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC3$
  $41A07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB$
- Para hacer el cálculo de $K = kG$, el algoritmo de exponenciación se traslada al algoritmo de duplicación (calcular $2G, 4G, 8G, 16G$, etc.)

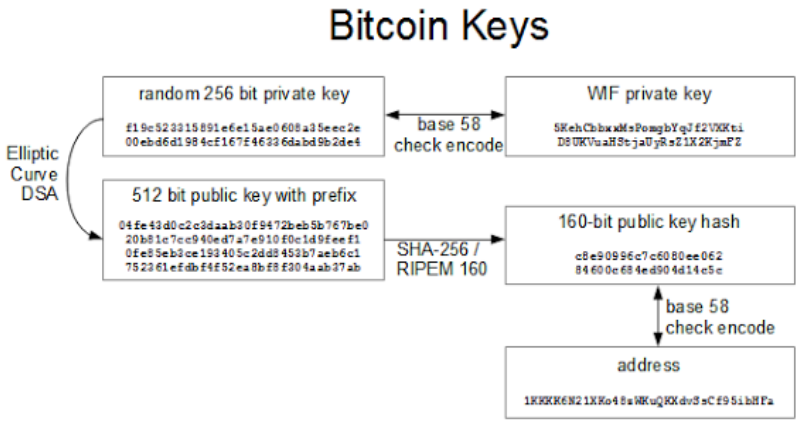


Bitcoin address:

- La dirección Bitcoin de hecho es una forma comprimida de hasheado K.
- Dirección Bitcoin = RIPEMD160(SHA256(K)).
- RIPEMD160 (RACE Integrity Primitives Evaluation Message Digest) es una función HASH similar a SHA1 pero desarrollada por la comunidad académica. Tiene una salida de 160 bits (20 bytes, 40 hex).



- Para la mejor lectura humana, se usa el alfabeto (codificación) Base58Check, de 58 caracteres y con un código detector de errores.
- El alfabeto es 123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz (números, mayúsculas y minúsculas, quitando 0,O, I, l)
- Se añade 1 byte de prefijo (0x00 para clave pública).
- Se calcula checksum=SHA256(SHA256(prefix+data)), que son 32 bytes, se toman los 4 primeros bytes, y se añade al final.
- Bitcoin address = Base58Check(prefix+data+suffix), que son 25 bytes.
- Ejemplo:
  - Public key: 0202a406624211f2abbdc68da3df929f938c3399dd79fac1b51b0e4ad1d26a47aa
  - Address: 1PRTTaJesdNovgne6Ehcdu1fpEdX7913CK



Claves privadas:

- Las claves privadas empiezan con 04, $K = 04xy$, con 130 hex.
- En formato Base58Check, empiezan con 5 o K.
- y está determinada por x salvo signo, dado que $y^2 = x^3 + 7$.
- En $\mathbf{Z}_p$ y tiene dos posibilidades: par o impar.
- Una versión comprimida se puede tomar $K = 02x$, si y par; $K = 03x$, si y impar.

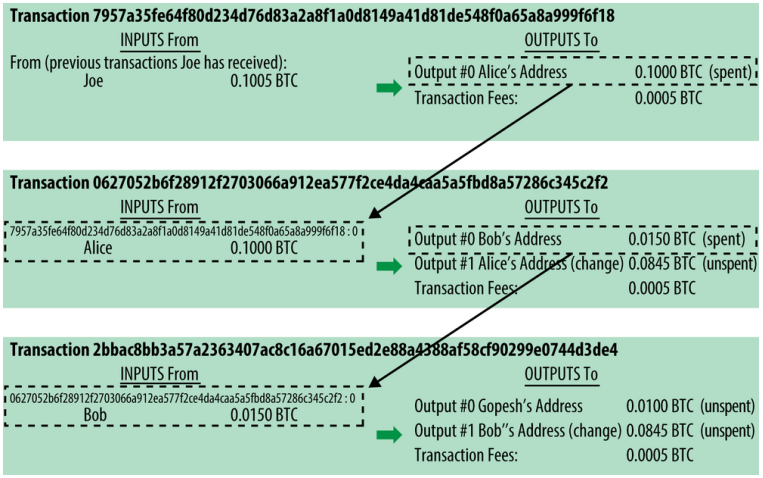
Wallets:

- Paper wallet. Para neuróticos: asegurarse de usar un generador aleatorio adecuado. Mejor generarlo offline. Copiarlo en papel o encriptarlo.
- Non-deterministic (random) wallet. El monedero genera pares de claves privadas y públicas independientes.
- Deterministic (seeded) wallet:
  - Se genera una secuencia aleatoria (entropía) de 128 a 256 bits.
  - Checksum tomando los primeros bits del SHA256 de la entropía.
  - Se añade como sufijo y se divide en secciones de 11 bits.

    - Se usan 2048 palabras para cada sección.
    - Se obtienen entre 12 y 24 palabras.
    - Se expande a 512 bits para formar la semilla.
- Con la semilla se generan las claves privadas y públicas con un sistema determinístico jerarquizado (HD):
    - Cada ancestro (comenzando con la semilla) genera una colección de hijos con un algoritmo que envuelve funciones hash.
    - Con un hijo no se puede tener información de los anteriores y posteriores.
    - La jerarquía se puede hacer a nivel de claves públicas o privadas independientemente.

Transacciones:

- Se pueden hacer online u offline. Para neuróticos: firmarla offline.
- Puede ser enviada por cualquiera (no hace falta ser el poseedor de los fondos)
- No contiene información confidencial (no se encritpta). Comparar con la conexión online a tu banco.
- Se envía a la red desde cualquier punto, y circula de nodo a nodo.
- Se comprueba su validez en cada nodo. Si es inválida (mal formada, que se gasta un dinero que no se tiene) se devuelve.
- Cada transacción incluye entre 1 y 9 inputs, y entre 1 y 9 outputs.
- Cada input refiere a una transacción anterior no gastada (denominada UTXO: *unspent transaction output*). Va firmada por el dueño de los fondos.
- Cada output indica una dirección bitcoin (será el dueño de los fondos del output).



- No se incluyen las fees. Simplemente son la diferencia input-output.
- Puede referirse a una transacción en el mismo bloque.
- La clave privada nunca aparece en la transacción.
- No obstante, se recomienda no volver a usar una dirección bitcoin una vez se haya firmado con su clave privada.
- Las fees de la transacción son calculadas estadísticamente por los wallets para que la transacción se ejecute en un tiempo razonable.
- Las transacciones se pueden emitir y firmar con *scripts*. Un script es un pequeño programa en lenguaje muy básico. Cuando aparece en el output se pide que aquel que quiera disponer de los fondos, ha de firmar con un input que sea un script que al antecederlo al script output nos dé TRUE.
- La transacción básica es:
    - Output: Bitcoin address (Public key hash)
    - Input (redeem): Public key + firma.

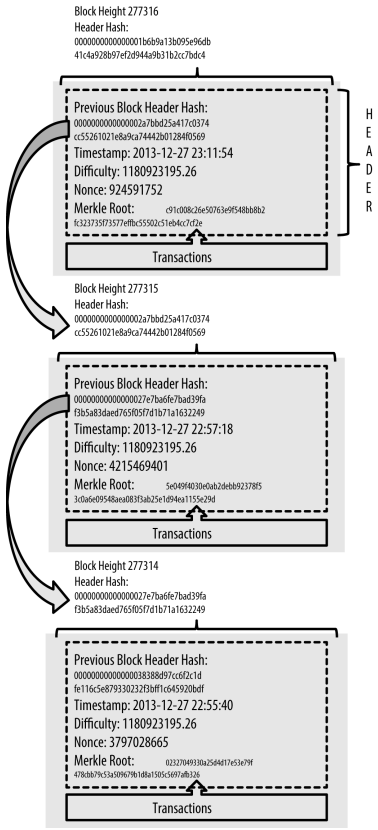
- Otros scripts permiten programar las transacciones para ejecutarse en el futuro en cierto bloque, poner condicionantes, etc.
- Se puede incluir un pequeño comentario.

Formación de bloques:

- Las nuevas transacciones flotan en la red y son recogidas por los mineros, para formar un nuevo bloque y añadirlo a la blockchain.
- El minero recopila las transacciones siguientes:
  - Una transacción coinbase, juntando la recompensa por bloque al minero (hoy día 6.25 Btc) con las fees del bloque (no se refiere a ninguna UTXO)
  - Las transacciones con alta prioridad (antigüedad*cuantía/Kbytes> 4)
  - Las transacciones con fees más altas
  - Otras transacciones hasta tamaño máximo (1MB).
- Con las transacciones se hace un árbol de Merkle para obtener un hash usando una estructura binaria.
- Puede añadir un comentario. Genesis block, 3 enero 2009: "Chancellor on the brink of second bailout for banks"

Prueba de trabajo:

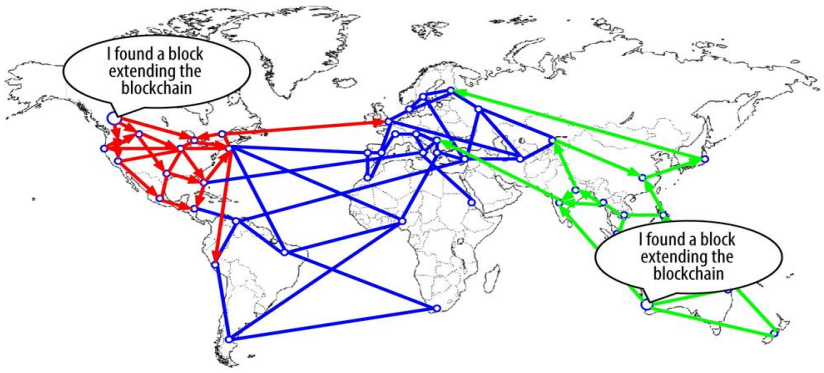
- La cabecera del bloque contiene el hash de la cabecera del bloque anterior, el hash del árbol de Merkle de las transacciones, la fecha, y el nuance.
- La dificultad es un número, y el nuance se ajusta para que el hash de la cabecera sea menor que ese número (obliga a comenzar por cierto número de ceros).
- La validación del bloque (minado) requiere resolver el siguiente problema de hash: se toma el bloque B, se une el nuance T y se calcula H=HASH(B,T). Sólo puede añadir el bloque a la blockchain si H$<$ dificultad (al día de hoy, comienza con 30 ceros).
- Para encontrarlo hay que realizar hashes repetidamente probando con diferentes T (proof of work). Una computadora tardaría años en encontrar T. Un Hash se hace rápido, pero 2^{30} lleva bastante trabajo. Entre todas las computadoras del mundo, alguna lo hace en 10 minutos.
- El minero (afortunado) envía el bloque a la red y lo añade a su blockchain. Se queda con la recompensa, pero sólo lo puede usar cuando haya tenido 100 confirmaciones.
- La dificultad (work-force) se ajusta cada dos semanas para que se genere un bloque cada 10 minutos.



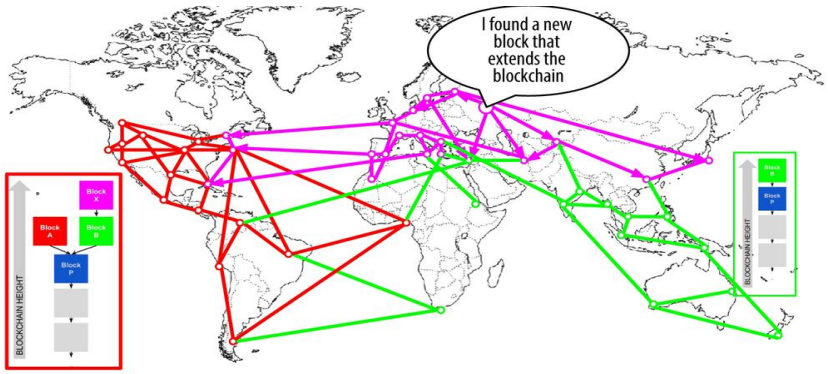
Blockchain:

- El minero exitoso transmite el bloque a la red, y los demás lo copian a su blockchain.
- 10 minutos es el tiempo medio de distribución de una información enviada a la red, y que se distribuye nodo a nodo por internet.
- Es improbable que varios mineros encuentren un bloque a la vez o casi a la vez, si varios grupos de mineros se encuentran trabajando con cadenas distintas, produce un fork.

Resolviendo forks:



- Eventualmente una será mayor. La cadena corta será ignorada por los nodos. Y las recompensas de los mineros que consiguieron esos bloques se desvanecen =(
- La regla es elegir la cadena que acumula mayor cantidad de Prueba de Trabajo.
- Para seguridad de que una transacción se necesitan varias confirmaciones (usualmente 3 confirmaciones, para seguridad 6 confirmaciones)



Hard fork:

- Un fork de un bloque ocurre una vez a la semana (cada 1000 transacciones).
- Un fork de dos bloques ocurre muy rara vez.
- Se pueden producir forks por disensiones entre grupos de mineros, por ejemplo en relación al código (si un grupo pide un cambio en el código y no se llega a un consenso). En este caso ocurre un *hard fork*, y la cadena se bifurca en dos, cada una gestionada por un grupo de mineros.
- En realidad se producen dos monedas distintas, simplemente comparten la parte inicial de la blockchain (ejemplo: Bitcoin Btc - Bitcoin Cash BCH).

Ataques maliciosos:

- Un cambio en un bloque intermedio cambia el Hash de ese bloque y de todos los posteriores. Habría que rehacer la Proof-of-Work de todos los bloques, lo que es casi imposible.
- Además debe hacerse de forma que se consiga la cadena más larga (para ello se necesita al menos el 51% del poder de computo de la red).
- Un ataque malicioso puede ejecutarse con éxito si es con pocos bloques. En 2013, Satoshi dice fue robado 1000 Btc con un ataque de doble gasto. El error fue que el exchange solicitaba 0 confirmaciones.
- Carreras de blockchain (double spend races):
  - Si un grupo de mineros con un hashrate de $p < 50\%$ quiere colocar una cadena maliciosa, la probabilidad de que haga n bloques antes que los mineros honestos (con $q = (1 - p)$ de

hashrate), será:

$$\frac{p^n}{(p^n + q^n)}$$

- Para $p = 10\%, n = 3$, tenemos una probabilidad de éxito del ataque de 0.13%.
- Para $n = 6$, la probabilidad es de 2.10^{-6}.

REFERENCES

[Al] MARK ALIZART, *Criptocomunismo* Ed. La Cebra (2020).

[Am] SAIFEDEAN AMMOUS, *The Bitcoin Standard: The Decentralized Alternative to Central Banking* Wiley (2018).

[An] ANDREAS M. ANTONOPOULUS, *Mastering Bitcoin: Programming the Open Blockchain.* O'Reilly Media, 2nd Edition (2017).

[Nak] SATOSHI NAKAMOTO, *Bitcoin: A Peer to Peer Electronic Cash System.* bitcoin.org (2009).

[PM] RICARDO PEREZ-MARCO, *Bitcoin and Decentralized Trust Protocols.* Newsletter of the European Math. Soc., 100 p.32, 2016.

[ST] JOSEPH H. SILVERMAN & JOHN T. TATE, *Rational Points on Elliptic Curves.* Springer, Undergraduate Texts in Mathematics, 2015.

[So] JIMMY SONG, *Programming Bitcoin: learn how to program Bitcoin from scratch.* O'Reilly Media, 2019.