UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CRIPTOGRAFÍA

TECNOLOGÍA DE LA BLOCKCHAIN

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática

Curso 2021-22

Autores:

- Lishuang Sun (María)
- Fabián Scherle Carboneres

ÍNDICE

1. **Introducción** ---------- 3
2. **Desarrollo** ---------- 3
   - 2.1. Fundamentos ---------- 3
     - 2.1.1. Descripción básica ---------- 3
     - 2.1.2. Blockchains públicas y privadas ---------- 4
     - 2.1.3. Hash y funciones hash ---------- 5
     - 2.1.4. Tipos de consenso ---------- 5
   - 2.2. Cripto-monedas ---------- 6
   - 2.3. Problema de las 2 cadenas ---------- 6
   - 2.4. Generación de bloques en la blockchain ---------- 6
   - 2.5. Bitcoin ---------- 7
     - 2.5.1. Descripción ---------- 7
     - 2.5.2. Estructura de los bloques ---------- 8
     - 2.5.3. Función Hashcash ---------- 9
     - 2.5.4. Dirección Bitcoin ---------- 10
   - 2.6. Ethereum ---------- 10
     - 2.6.1. Descripción ---------- 10
     - 2.6.2. Contratos inteligentes ---------- 11
     - 2.6.3. Máquina virtual Ethereum ---------- 11
     - 2.6.4. Función Ethash ---------- 11
   - 2.7. Monero ---------- 12
     - 2.7.1. Descripción ---------- 12
     - 2.7.2. Algoritmo CryptoNight ---------- 12
     - 2.7.3. Mecanismos de privacidad ---------- 13
   - 2.8. Aplicaciones ---------- 13
     - 2.8.1. Sistema de distribución de firmware ---------- 13
     - 2.8.2. Sistema de seguimiento de transportes ---------- 14
3. **Conclusión** ---------- 14
4. **Bibliografía** ---------- 15

1- Introducción.

La blockchain es una tecnología que surge como un nuevo concepto en el mundo de las redes. Permite compartir cierta información, como registros, certificaciones o archivos, de forma digital y descentralizada, sin necesidad de una entidad de confianza que imponga su criterio a los participantes. Esa capacidad es lo que convierte a la tecnología blockchain en algo tan apasionante y que cambia muchos paradigmas sobre sistemas "clásicos" de redes e informática.

Comenzando por su definición, base de datos compartida por una gran cantidad de usuarios, distribuida en una red P2P, protegida criptográficamente y organizada en bloques relacionados entre sí, lo que permite almacenar información de forma inmutable y ordenada. Su principal característica por ende es su descentralización y su clave es el consenso.

La blockchain puede proporcionar robustez, seguridad, transparencia y escalabilidad a grandes sistemas de datos, lo que permite hacer frente a un amplio abanico de amenazas. Esto incluiría desde fugas de información a manipulación maliciosa del contenido.

Mediante la blockchain, estas amenazas pueden combatirse trazando individualmente todas las acciones realizadas sobre los datos, resultando en una auditoría constante.

2- Desarrollo.

2.1. Fundamentos

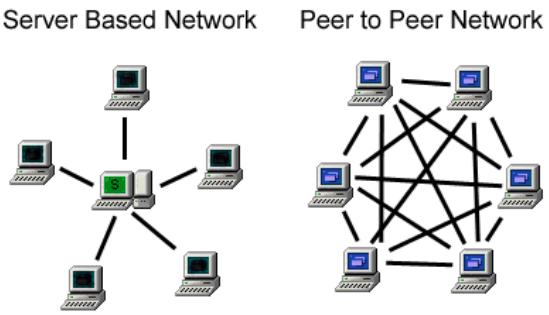
2.1.1. Descripción básica

Al igual que como su propio nombre lo indica, consiste en una cadena de bloques, donde los bloques contienen un cierto tipo de información que variará dependiendo de la implementación, como ejemplos reales algunas contienen transacciones de criptomonedas, contratos inteligentes, imágenes incluso sets de datos para entrenar modelos de inteligencia artificial.

Dichos bloques se conectan con el uso de *hashes*, generados por funciones matemáticas, las cuales serán explicadas más adelante.

Principalmente hay que tener en cuenta una serie de elementos claves (1) para la comprensión de la cadena de bloques:

- **Nodo:** también denominado minero o equipo de minería, podría ser desde un ordenador personal o incluso una Raspberry Pi hasta una mega computadora o ASIC. Su función es participar en la red, en el proceso de escritura de datos en la cadena de bloques a cambio de una recompensa, la escritura de una transacción a su cuenta. Dicha información sólo podrá ser añadida si existe un acuerdo entre la mayoría de los nodos.
- **Protocolo:** Establece un lenguaje de comunicación entre los nodos.
- **Red P2P (Peer-to-Peer):** se basa en una red en la que la comunicación se realiza entre 2 nodos, sin intervención de un intermediario (servidor central).



Un ejemplo bastante conocido es BitTorrent (3), el cuál funciona muy resumidamente de la siguiente forma: cada vez que un nodo o usuario quiere descargar un archivo, no lo descarga directamente sino que hace uso de un fichero torrent que es una especie de mapa cuyo objetivo es establecer conexión con otros clientes que tengan el archivo que se desea descargar. De esta manera, se descarga por fragmentos provenientes de diferentes nodos o usuarios, lo que significa que el archivo no está alojado en un lugar concreto, por lo que no se corre peligro de que sea eliminado, pero sí que se necesita que haya otros nodos con él para su disponibilidad.

- **Descentralización:** el sistema deja de ser controlado por una entidad y pasa a ser controlado por los nodos que participan en él. La idea principal es que los nodos sean iguales entre sí sin necesidad de establecer niveles jerárquicos, aunque existen implementaciones privadas que tienen jerarquía.
- **Consenso:** protocolo que se establece para que los nodos participantes verifiquen y confirmen una escritura a la cadena.
- **Privacidad en la red:** es la capacidad de mantener ocultos una serie de datos, archivos e información. Evitando que posibles intrusos puedan acceder a dichos datos.
- **Anonimato en la red:** a diferencia de la privacidad, no busca ocultar todos los datos, pero sí aquellos relacionados con la identidad.
- **Pseudo-anonimato en la red:** en este caso no se oculta ningún dato, sin embargo aquellos relacionados con la identidad se modifican, haciendo uso de seudónimos (nombre falso) como identificación, evitando tener que revelar la verdadera identidad.

2.1.2. Blockchains públicas y privadas

Existen unas series de características y aspectos que ha hecho dividir al sistema en dos grupos (1).

Las implementaciones públicas se destacan de forma general por:

- Cualquier persona sin ser usuario puede consultar las transacciones realizadas.
- Cualquier persona puede convertirse en usuario y participar.
- No existe un usuario que tenga más poder que otro en la red y todos los nodos son iguales entre sí.
- Los propietarios del contenido insertado en el bloque, transacciones por ejemplo, no son identificables a simple vista, pero sus direcciones en algunos casos sí pueden ser rastreables.

Por otra parte en caso de las versiones privadas:

- Solo personas o entidades invitadas a participar son capaces de convertirse en usuarios del sistema. Pueden incluirse distintos niveles de acceso a los usuarios.

- El número de nodos de los que se componga la blockchain privada puede estar limitado al número de participantes o nodos.
- Pueden establecer el nivel de anonimato que se quiera para proteger el contenido de los bloques.
- No todos o ninguno de los datos inscritos tienen difusión pública.

2.1.3. Hash y funciones hash

Un hash no es más que una secuencia de números o caracteres que se obtienen aplicando una función matemática dado unos datos, dicha función se denominará función hash y siempre que se aplique la misma función al mismo dato, obtendremos el mismo hash y cualquier modificación del contenido, cambiará por completo el hash resultante.

Muchas de estas funciones hash cumplen la definición de función unidireccional, las cuales consisten en que si se conoce el método el algoritmo de la función y un hash calculado por dicha función es extremadamente costoso computacionalmente hablando el poder calcular el dato que originó ese hash. Adicionalmente cumplen las siguientes características:

- Eficiencia de cálculo: permite generar rápidamente el hash y a bajo coste.
- Resistencia a preimagen: es muy difícil obtener un mensaje de entrada que produzca un hash dado.
- Resistencia a segunda preimagen y a la colisión: es muy difícil encontrar dos mensajes distintos de tal forma que ambos den como resultado el mismo hash.

2.1.4. Tipos de consenso

Existe un dilema clásico en seguridad para sistemas descentralizados relacionado con la sincronización para evitar fallos y ataques, denominado problema de los “generales bizantinos” (1). Este problema se basa en un grupo de generales que están dispersos, se comunican mediante mensajes entre sí y deben de llegar a un acuerdo para atacar una ciudad o retirarse, para tener éxito deben atacar al mismo tiempo o retirarse.

Con tal de solventar el dilema destacan tres diferentes tipos de consensos (4):

- **Proof-of-Work (PoW):** consiste en encontrar una solución a un problema matemático de dificultad avanzada o dinámica. De esta manera el sistema se protege contra nodos que traten de burlar a la red ya que estos tendrían que resolver una cantidad de problemas tan grande que la posibilidad de que lo hagan se vuelve insignificante. En este tipo de consenso existe una vulnerabilidad que es el ataque del 51% que podría pasar si un nodo controla el 51% del poder computacional de la red, teniendo el control de la misma para realizar todo tipo de trampas como bloquear escrituras o en caso de transacciones poder gastar el mismo dinero dos veces.
- **Proof-of-Stake (PoS):** los participantes solo pueden contribuir a la computación en proporción a sus recursos. Esto quiere decir que la creación de bloques es llevada a cabo por los nodos que poseen la cripto-moneda subyacente (stakeholders), lo que por sí solo consigue que no haya interés en corromper el correcto funcionamiento del sistema. Adicionalmente, se asigna una probabilidad de crear bloques proporcional a la cantidad de moneda que posee cada nodo. Es por ello

que no solo es más seguro según la teoría de juegos, sino que también es más sustentable.
- **Proof-of-Importance (PoI):** es un mecanismo de consenso que asigna un puntaje a los nodos participantes basados en su participación en el sistema, en vez de asignarlo sólo en sus recursos (cantidad de monedas). Siendo un incentivo muy útil.

2.2. Cripto-monedas

Una blockchain puede servir como centro de intercambios de confianza entre múltiples entidades sin que unas confíen en la otras, ni tan siquiera en un intermediario.

Las criptomonedas basadas en blockchains eliminan también la necesidad de una autoridad central. El criterio de emisión de nuevas unidades monetarias se encuentra prefijado. En el caso del Bitcoin, por ejemplo, se emite nueva moneda cada vez que se mina un bloque (cada 10 minutos aproximadamente) y se pone en posesión del nodo que lo ha minado.

Existen decenas de criptomonedas. Todas ellas comparten su utilidad como sistema de pago. Algunas utilizan una blockchain propia y otras funcionan encima de la blockchain de bitcoin. Su funcionamiento es bastante heterogéneo y todas ellas pretenden aportar alguna mejora respecto a bitcoin (2).

2.3. Problema de las 2 cadenas

Podría llegar a ocurrir que dos bloques se generen en el mismo instante (con muy pocos segundos de diferencia entre ellos). Ambos bloques pueden ser válidos y cumplir con las condiciones exigidas, pero sólo uno de los dos debe formar parte de la cadena de bloques principal, puesto que solo se admite una cadena en el sistema.

Para solucionarlo, los nodos de la red generarán nuevos bloques sobre aquel que han recibido en primer lugar a través de la red. Desde ese entonces los nodos acuerdan que la cadena más larga es la que formará parte de la cadena de bloques principal. Esto quiere decir que el otro bloque se descarta (incluyendo los bloques generados después de este) y pasa a ser inválido, y todas las transacciones incluídas en el mismo pasan a la cola de transacciones pendientes para ser añadidas a un nuevo bloque.

El significado de la cadena de mayor longitud, no es la cadena con más bloques desde el bloque génesis (primer bloque), sino que se interpreta como la cadena que combina los bloques con mayor dificultad acumulada.

2.4. Generación de bloques en la blockchain

En primer lugar, una persona debe convertirse en usuario dentro del sistema para poder escuchar y emitir nueva información.

En segundo lugar, si el usuario desea convertirse en minero y crear nuevos bloques debe competir contra el resto de mineros en la red para resolver el rompecabezas criptográficos y así ser el que escriba el nuevo bloque en la blockchain.

La nueva información (transacciones por ejemplo) emitida es validada por los nodos más cercanos al emisor, descartando todas aquellas que sean inválidas y propagando al resto de nodos las válidas, es decir, aquellas que cumplen con las especificaciones de la red. Posteriormente, se procede a añadirlas a la cadena de bloques.

Este proceso de confirmación de datos se lleva a cabo mediante el protocolo de consenso.

Finalmente, los nodos comprueban que, en el nuevo bloque creado/minado, toda la información es válida y que el bloque está directamente vinculado con su predecesor, es decir, que contiene el hash del bloque anterior en su cabecera. En caso afirmativo el bloque es añadido a la blockchain incrementando así la cadena.

Este proceso se repite generando una nueva ronda de minado con la nueva información emitida que aún no haya sido agregada en ningún bloque anterior de la blockchain. Si el bloque es inválido, es descartado, y el resto de nodos siguen el proceso de minado hasta encontrar un bloque válido.

2.5. Bitcoin

2.5.1. Descripción

La primera de todas las blockchains fue la blockchain pública de Bitcoin, presentada en 2008 por el usuario Satoshi Nakamoto. Este sistema, aunque es abierto, es también semi-anónimo, debido a que los usuarios se identifican con claves públicas y no con sus identidades reales y la información de sus bloques viene constituida por transacciones de criptomonedas.

La creación de nuevos bloques es realizada por nodos denominados mineros con el consenso de prueba de trabajo o Proof-of-work (PoW).

Su proceso de autenticación de transacciones se basa en criptografía asimétrica (clave pública). Donde cada cuenta de usuario de Bitcoin posee dos llaves relacionadas matemáticamente: una pública (identificador del usuario en la red, conocida por todos) y una privada (secreta, conocida por el usuario). La llave privada se usa para firmar las transacciones emitidas por el usuario; este especifica las cantidades de moneda a transferir y las llaves públicas de destino. La red y el resto de usuarios, usando la llave pública del emisor, pueden obtener una prueba matemática de que la transacción fue efectivamente firmada por ese usuario y por nadie más, puesto que nadie más tiene su llave privada.

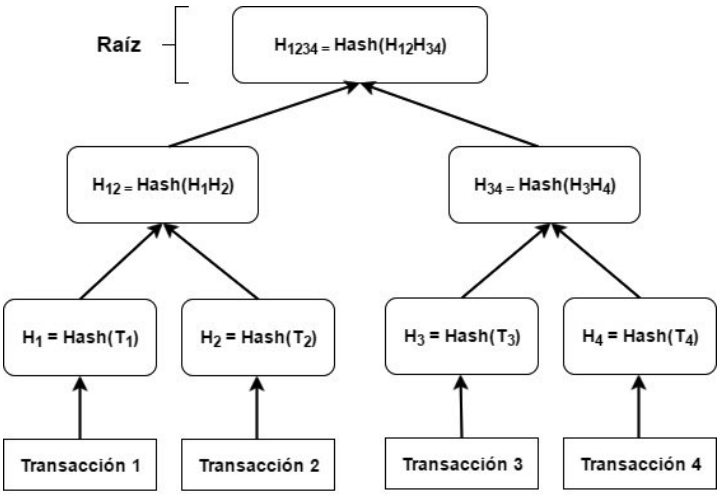
En concreto, para que un bloque sea aceptado, el minero tiene que ser el primero en completar un rompecabezas matemático, mencionado en el apartado 2.1.4, el cual presenta una dificultad que se reajusta cada 2016 bloques (que equivalen a catorce días),

con tal de que la creación de nuevos bloques tenga una frecuencia aproximada de un bloque cada diez minutos. La fórmula para dicho ajuste es la siguiente: dificultad_nueva = dificultad_previa * 2 semanas / (tiempo en minar los últimos 2016 bloques).

2.5.2. Estructura de los bloques

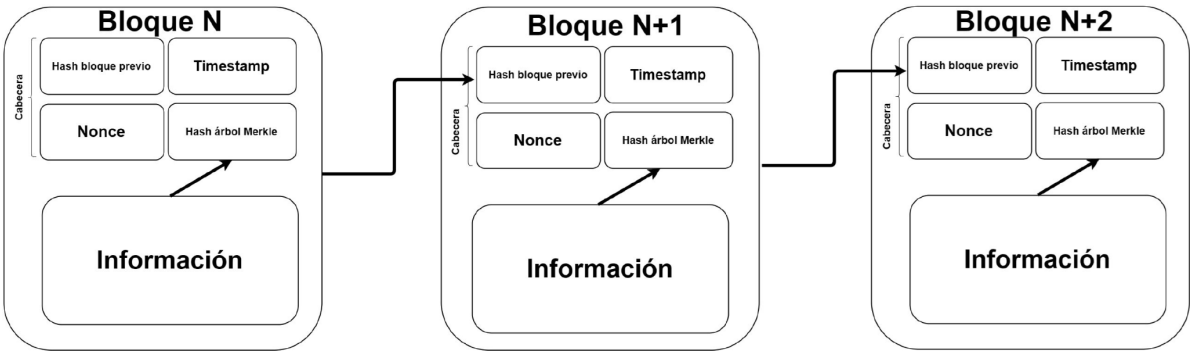
La blockchain almacena una gran cantidad de datos y además su tamaño es creciente con el tiempo ya que en la misma solo se añade información. Es por ello que en la blockchain de bitcoin, se propone utilizar un árbol hash de Merkle, que permite realizar consultas sin tener que descargar toda la información almacenada.

El árbol de hash de Merkle permite almacenar diversas piezas de información independientes en las hojas de una estructura en árbol.



Se puede consultar de forma autenticada cualquier contenido del árbol con una cantidad de valores hash proporcional al logaritmo del número de nodos del árbol. Esto es porque para validar un contenido únicamente hay que proporcionar los nodos adyacentes en cada nivel y el nodo raíz autenticado. Entonces, para validar un contenido se calcula el valor raíz a partir de los nodos adyacentes proporcionados y se comprueba que coincide con el valor raíz autenticado (2).

A continuación se muestran los campos que contiene cada bloque de bitcoin, es una especie de guía para las implementaciones de las blockchains, ya que hay muchas que han optado por seguir esta estructura.



- **Número mágico (4 bytes):** siempre toma el valor 0xD9B4BEF9.

- **Tamaño del bloque (4 bytes):** número máximo de bytes contenidos en el bloque sin contar el número mágico ni el tamaño del bloque.
- Cabecera del bloque (80 bytes): consta de 6 elementos:
  1. **Versión (4 bytes)**: versión del bloque.
  2. **Hash del bloque previo (32 bytes):** Vincula secuencialmente los bloques formando una cadena.
  3. **Raíz del árbol Merkle (32 bytes):** hash asociado al nodo raíz del árbol Merkle.
  4. **Marca de tiempo o timestamp (4 bytes)**: identifica el instante en segundos en el que fue creado el bloque.
  5. **nBits (4 bytes):** valor de dificultad requerida para generar el bloque.
  6. **Nonce (4 bytes):** es un contador encontrado por fuerza bruta en el proceso de minado que permite obtener los hashes de los bloques.
- **Cantidad de transacciones (1-9 bytes):** número de transacciones contenidas en el bloque, incluyendo la coinbase, la cual se encarga de transmitir las nuevas monedas al minero que encontró el hash. Esto significa que el valor total de cada transacción coinbase está formado únicamente por nuevas monedas que nunca han circulado por la blockchain.
- **Transacciones/Información (limitado por el tamaño del bloque):** transacciones incluidas en el bloque en el mismo orden utilizado para generar la raíz del árbol Merkle. Debe contener como mínimo la transacción coinbase. En el caso de Bitcoin para calcular el id de la transacción (TXID) se utiliza una doble función hash con el algoritmo SHA-256 (se aplica 2 veces SHA-256(SHA-256(x))) como medida de seguridad ante el criptoanálisis.

2.5.3. Función Hashcash

Es un sistema de proof-of-work que fue inicialmente diseñado en 1997, con el objetivo de limitar el spam en el correo electrónico o los ataques de denegación de servicio (7). Este sistema se convirtió en el algoritmo a utilizar en Bitcoin y para cada hash que se genera prevé una media de 10 minutos.

Hashcash exige que el emisor debe calcular un hash que contenga x bits de dificultad a la izquierda con valor de cero. Para ello requiere de tres parámetros para funcionar y adaptarse a Bitcoin:

- **Cabecera del bloque:** los 6 campos del apartado anterior.
- **Nonce:** número de 32 bits que se modifica secuencialmente para obtener un hash distinto. Una vez agotado es necesario modificar algún campo de la cabecera o el extra nonce.
- **ExtraNonce:** forma parte de la transacción coinbase. Requiere recalcular todos los nodos de la rama izquierda del árbol Merkle debido a que el coinbase se sitúa en el nodo más a la izquierda.

Un ejemplo de implementación:

https://gist.github.com/i3visio/388ef5154052ed8173df4b7b9eda541b

2.5.4. Dirección Bitcoin

Como otra característica de Bitcoin destaca el uso de ECDSA como algoritmo de clave pública para firmar y verificar transacciones (1). Este algoritmo no utiliza números primos como RSA, sino coordenadas en una curva elíptica, la cual se expresa con la ecuación $y^2 = x^3 + ax + b$. Dependiendo del valor de a y b las curvas elípticas tomarán diferentes formas en el plano, en el caso de Bitcoin se utiliza un valor de a=0 y b=7. Cabe destacar que se optó por este algoritmo porque requiere números menores en relación a RSA.

Adicionalmente, se debe tener en cuenta en este protocolo que la clave privada siempre es secreta, la clave pública es secreta hasta que se hace una transacción y la dirección (hash de la clave pública) siempre es pública.

Tal y como se puede ver en la figura Bitcoin define el uso de dos tipos de algoritmos de obtención hash: SHA-256 y RIPEMD-160.

2.6. Ethereum

2.6.1. Descripción

Probablemente, la más prometedora entre las alternativas es Ethereum, que es la segunda criptomoneda con mayor capitalización. Ethereum cuenta con una blockchain propia distinta a bitcoin y por el momento también basada en PoW (aunque tiene planeada una migración a PoS). Más allá de su funcionamiento como cripto-moneda, la aportación principal de Ethereum son los contratos inteligentes (Smart Contracts).

2.6.2. Contratos inteligentes

Los contratos inteligentes son scripts (pequeños códigos) auto-ejecutables bajo determinadas condiciones que residen en la blockchain y que permiten automatizar gran cantidad de procesos comerciales de una forma segura y transparente para todos los participantes.

Dichas acciones a ejecutar han sido revisadas y aceptadas por las distintas partes que han firmado dicho contrato. De esta forma, el contrato inteligente hace cumplir sus condiciones programadas presentando una respuesta de acuerdo a sus cláusulas de forma completamente autónoma (9).

2.6.3. Máquina virtual Ethereum

Ethereum Virtual Machine (EVM), es un software con el objetivo de servir como capa de abstracción en la ejecución del código que se almacena en la blockchain. Busca evitar que una DApp (aplicación descentralizada) maliciosa o programador de contrato inteligente amenace la seguridad de los nodos de la red y la propia red.

La EVM realiza una abstracción completa del sistema gestionando el acceso a los recursos de los ordenadores y limitando sus acciones en un entorno controlado o máquina virtual. Además, permite la operación de contratos inteligentes y las DApps gracias al uso del lenguaje de programación Solidity.

2.6.4. Función Ethash

Ethash usa la función hash Keccak, también conocida como SHA-3 (9). Su funcionamiento comienza por la creación de una semilla calculada a través de las cabeceras de bloque. Dicha semilla se utiliza para calcular y crear una caché pseudoaleatoria con un tamaño de 16 MB. La cual se utiliza para crear un conjunto de datos con un tamaño de más de 4 GB. Estos datos conforman una estructura que usa un DAG (Grafo Acíclico Dirigido).

Dicho DAG se actualiza cada 30 mil bloques y su creación da lugar al proceso de minería. Durante este proceso se toman valores aleatorios del DAG y estos son combinados con datos suministrados por la red y el resto de transacciones para ser verificadas. Para la fase final se realiza la verificación a través de un proceso que genera partes determinadas del conjunto de datos anteriores mediante la memoria caché para acelerar dicho proceso.

Su principal característica era su resistencia a la minería usando dispositivos ASIC, debido a que sus creadores buscaban mantener la minería por GPU y evitar la centralización de la misma.

2.7. Monero

2.7.1. Descripción

Este sistema surgió con el enfoque de incrementar el nivel de anonimato. Inicialmente se basó en el protocolo CryptoNote que ya utilizaba el sistema Bytecoin y que es lo que le proporciona el nivel de anonimato que Bitcoin carece (10).

Sus principales características son:

- Basado en el protocolo CryptoNote.
- Uso del algoritmo PoW de CryptoNight.
- Programada para generar 1 bloque cada 2 minutos.
- Utiliza Ring Signatures o firmas en anillo, tecnología criptográfica que le da privacidad e impide que el receptor de una transacción pueda rastrear la dirección del remitente.
- La conexión de los nodos se hace mediante la red anónima I2P, dificultando que puedan conocerse datos sobre transacciones.

Destacar que se puede hacer transferencias de Monero a bitcoins usando servicios como el de xmr.to y mantener la privacidad.

2.7.2. Algoritmo CryptoNight

El algoritmo de minería CryptoNight (11) es un algoritmo de minería concebido para ser singularmente eficaz en CPU y resistente a ASIC, con el propósito de permitir una mayor descentralización de la minería de las criptomonedas que lo apliquen, y así puedan ofrecer avanzadas opciones de privacidad y anonimato.

Se fundamenta en una serie de propiedades:

- Utilización del cifrado AES nativo. Las CPU con capacidad de aceleración por hardware para cálculos AES pueden verse favorecidas y tener un potencial de minería superior.
- Uso de funciones hash como Keccak y Blake-256.
- Presenta un conjunto de multiplicadores veloces de 64 bits, arquitecturas de CPU de 64 bits, son eficaces.
- Utilización intensiva de memorias caché de la CPU.

Para su funcionamiento, CryptoNight toma una entrada de datos y lo lleva a la función Keccak-1600 de 1600 bits de ancho. Luego, toma los primeros 31 bytes de este hash y los cifra en diez rondas utilizando AES-256. De esta forma todos los datos generados se envían al espacio de trabajo o bien scratchpad de CryptoNight. Al acabar con este proceso, empieza el ciclo de generación del hash.

Para generar el hash se toma todo el conjunto de datos creado por las funciones AES-256 y Keccak, y lo pasa por el resto de funciones hash. De esta forma se consigue un hash final con una extensión de 256 bits.

2.7.3. Mecanismos de privacidad

En Monero se utilizan 3 mecanismos enfocados en la privacidad durante una transacción. El primero es la forma en anillos, el cual es un protocolo que sirve para probar que un firmante pertenece a un grupo sin tener que identificarlo.

Para lograr el correcto funcionamiento de este grupo de firmas se deben cumplir tres condiciones (12):

- Ambigüedad con el firmante: un testigo sabe que un firmante pertenece a un determinado anillo, pero no debe conocer la identidad de dicho firmante.
- Enlazabilidad: En caso de que se use la misma firma privada para firmar dos mensajes diferentes, los mensajes estarán enlazados con el objetivo de impedir los ataques de doble gasto.
- Probabilidad de falsificar una firma es nula: Esto permite evitar el robo de fondos.

En el sistema de firmas en anillo se mezcla la dirección del emisor de la transacción con un grupo de otras direcciones, aquellas que hayan emitido una transacción con una cantidad similar de monedas.

Como segundo mecanismo existe otro sistema de privacidad, llamado Círculo de Transacciones Confidenciales o Ring Confidential Transaction, que tiene como objetivo ocultar el monto transferido. Lo hace transmitiendo solo una pieza de información en vez del monto de la transacción, dicha información es suficiente para verificar que la cantidad a transferir es válida.

Y por último se encuentra el mecanismo de Dirección Secreta o Stealth Address, el cuál tiene como objetivo esconder la identidad del receptor. Se basa en que el remitente de la transacción crea direcciones aleatorias únicas y gracias al uso de un direccionamiento oculto se puede saber solo para el remitente y destinatario dicha dirección. De esta forma el receptor puede gastar los fondos de una dirección secreta de único uso, con su clave secreta.

2.8. Aplicaciones

2.8.1. Sistema de distribución de firmware

Muchos dispositivos no están supervisados con los niveles usados en el mundo de la computación. Además, para los fabricantes resulta bastante costoso tener que enviar directamente a millones de dispositivos actualizaciones y realizarlo incluso después de años de haber abandonado la fabricación de tales dispositivos.

Al mismo tiempo, para el consumidor, existe cierta desconfianza en dispositivos que se comunican con su fabricante sin la supervisión del usuario final y sería mucho mejor un enfoque de seguridad más transparente.

Este escenario, bien podría solucionarse con una blockchain. En este caso, se aprovecharía tanto su característica de sistema distribuido y su transparencia, como su robustez y fiabilidad. Los dispositivos consultarán la blockchain para averiguar si su

firmware está actualizado. En caso que no lo esté, pedirán a otros nodos que les manden la nueva versión.

Una vez recibida, podrían usar el código de la blockchain para comprobar que el firmware no ha sido alterado en modo alguno, evitando así las intrusiones.

El fabricante solo tendría que mandar la actualización a unos pocos nodos y dejar que la actualización se propague.

2.8.2. Sistema de seguimiento de transportes

En un envío internacional de mercancías participan normalmente varias empresas ya que se utilizan varios medios de transporte. Todas ellas tienen sus bases de datos independientes donde actualizan el estado del envío en función de la información proporcionada por las otras o por sus agentes. La base de datos (la propia blockchain) sería compartida por todos los intermediarios y por el remitente y destinatario, reduciendo los costes.

Al llegar el contenedor en cuestión a un cierto puerto, se añadiría una actualización a la base de datos. Al estar todas las actualizaciones firmadas con las claves privadas del que entrega y el que recoge y con la clave del contenedor, esta actualización actuaría como prueba criptográfica de que el contenedor se encuentra ahora en posesión del administrador del puerto. Además, el sistema incluye marcas de tiempo (timestamps) para hacer el seguimiento.

La transparencia y fiabilidad del concepto ayudaría sin duda a la resolución de disputas entre los participantes.

Si a este enfoque innovador se le añade IoT se puede ganar aún más eficiencia. Los contenedores y los lugares de intercambio pueden tener dispositivos incorporados que automaticen totalmente el proceso, disminuyendo los costes aún más y reduciendo las probabilidades de error y de fraude.

3. Conclusión

Como podemos observar, la blockchain es un sistema robusto para guardar información distribuida sobre la red entera, difícil o imposible de alterar o hackear gracias a su descentralización, que además es supervisada por varios nodos en la red constantemente. Sus diferentes protocolos de consenso hacen que no sea interesante para el atacante burlar el sistema blockchain, ya que requiere de más recursos para atacar que de los que puede obtener por hacerlo.

Dada esta seguridad del sistema, hace posible su aplicación en campos como la distribución de firmware y seguimiento de transportes de mercancías.

4. Bibliografía:

(1) Preukschat, A. (23 mayo 2017). *BLOCKCHAIN: LA REVOLUCIÓN INDUSTRIAL DE INTERNET*. Gestión 2000. https://www.academia.edu/36701339/Blockchain_La_revoluci%C3%B3n_industrial_de_internet_Alexander_Preukschat

(2) DOLADER RETAMAL, C., BEL ROIG, J., & MUÑOZ TAPIA, J. L. (n.d.). *LA BLOCKCHAIN: FUNDAMENTOS, APLICACIONES Y RELACIÓN CON OTRAS TECNOLOGÍAS DISRUPTIVAS*. https://www.mincotur.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/405/DOLADER,%20BEL%20Y%20MU%C3%91OZ.pdf

(3) Fernández, Y. (2021, February 8). *BitTorrent: qué es y cómo funcionan los torrents*. Xataka. Retrieved December 8, 2021, from https://www.xataka.com/basics/bittorrent-que-como-funcionan-torrents

(4) Niebuhr, K. (2017, May 31). *Los diferentes tipos de consenso del Blockchain – Karlbooklover*. Karlbooklover. Retrieved December 8, 2021, from https://www.karlbooklover.com/consensos-del-blockchain/

(5) *Entendiendo los Protocolos de Consenso de Blockchain | by Federico Ast | Astec*. (2019, May 25). Medium. Retrieved December 8, 2021, from https://medium.com/astec/entendiendo-los-protocolos-de-consenso-de-blockchain-4858c71722d2

(6) Zamorano, V. (2020, August 17). *Más protocolos de consenso blockchain: PoI, PoA, PoC, PoB… – BLOCKCHAIN SERVICES*. BLOCKCHAIN SERVICES. Retrieved December 8, 2021, from https://www.blockchainservices.es/uncategorized/mas-protocolos-de-consenso-blockchain-poi-poa-poc-pob/

(7) *Crypt4you*. (2014, November 3). Crypt4you. Retrieved December 8, 2021, from http://www.criptored.upm.es/crypt4you/temas/sistemaspago/leccion3/leccion03.html

(8) Muñoz, M. (2021, June 13). *Todo sobre el algoritmo Ethash*. Bitnovo Blog. Retrieved December 8, 2021, from https://blog.bitnovo.com/todo-sobre-el-algoritmo-ethash/

(9) *Qué es Ethereum: lo que debes saber sobre esta criptomoneda*. (2021, July 20). Mafius. Retrieved December 8, 2021, from https://www.mafius.com/curiosidades/que-es-ethereum/

(10) *Qué es Monero (XMR)? Así funciona esta criptomoneda privada*. (n.d.). Bits of Proof. Retrieved December 8, 2021, from https://bitsofproof.com/es/monero

(11) *¿Qué Es El Algoritmo De Minería CryptoNight?* (n.d.). Entrecriptos. Retrieved December 8, 2021, from https://entrecriptos.com/que-es-el-algoritmo-de-mineria-cryptonight/

(12) *Todo sobre las firmas de Anillo*. (2021, April 23). Crypto Dummy. Retrieved December 8, 2021, from https://crypto4dummy.com/que-son-las-firmas-de-anillo/