Algoritmo de consenso, criptografía y seguridad

Blockchain y computación Cuántica

tech

CONTENIDO

Objetivos

- Conocer los principales algoritmos de consenso y sus características.
- Aprender las características de la solución criptográfica que utiliza *blockchain*.
- Entender los distintos aspectos de la seguridad en *blockchain*.

Algoritmos de consenso

La tecnología *blockchain* requiere de un mecanismo que permita seleccionar el nodo o nodos que generan o validan los bloques que se incluyen en la cadena.

Existe una gran variedad de algoritmos de consenso[1,2], casi tantos como soluciones *blockchain*.

La elección del algoritmo de consenso a utilizar en un *blockchain* no es una decisión sencilla, ya que hay que tener en cuenta las características, la finalidad y la proyección que se espera del *blockchain* para poder elegir o idear el algoritmo de consenso que mejor rendimiento pueda ofrecer a la solución *blockchain*.

Todos los algoritmos de consenso cuentan con ventajas e inconvenientes. En el siguiente listado se analizan los algoritmos más representativos:

Proof of work (PoW)

La prueba de trabajo es el primero de los algoritmos de consenso utilizados en *blockchain* y fue ideado para seleccionar el bloque que se añade a la cadena de bloques en *bitcoin ethereum* también usa este algoritmo de consenso, aunque está inmerso en un proceso de cambio de algoritmo de consenso a *proof of stake* que se completará en 2021.

El algoritmo plantea un reto que ha de ser resuelto por los participantes, el primer participante en resolver el reto se proclama ganador y recibe las recompensas.

- **Las principales ventajas de PoW son:**
  - Realizar un ataque a este sistema requiere de una gran inversión y plantea una gran complejidad, por lo que se considera un sistema seguro cuando la red de mineros es suficientemente grande.
  - No existen limitaciones para participar, todos los nodos mineros participan en la resolución del reto.
- **Los inconvenientes de PoW son:**
  - El excesivo gasto de energía que requiere para generar un bloque.
  - Se necesita de una gran capacidad de cómputo para poder resolver el reto planteado en este algoritmo de consenso.
  - El número de transacciones procesadas por minuto es limitado. En *bitcoin* los bloques se generan con una media de 10 minutos.
- **Soluciones *blockchain* que utilizan PoW:**
  - *Bitcoin*
  - *Ethereum* (en proceso de cambio a PoS).

Proof of stake (PoS)

El algoritmo de prueba de participación especifica la necesidad de disponer de una determinada criptomoneda para poder formar parte del conjunto de nodos que participan en la creación de un bloque. De este conjunto se seleccionan aleatoriamente un número determinado de nodos que son los encargados de realizar las validaciones del contenido del nuevo bloque.

En muchas implementaciones de este algoritmo, los nodos que disponen de mayor cantidad de criptomonedas tiene más probabilidades de ser elegidos para participar en la generación de un nuevo bloque.

Para participar en proceso de generar nuevos bloques, se necesita realizar el depósito de una cantidad de criptomoneda. Esta cantidad se devuelve si la validación es correcta y puede no devolverse si el minero no realiza correctamente la validación del contenido del bloque.

La implementación de este algoritmo de consenso tiene múltiples variantes, pero el concepto es similar al de participar en una rifa, donde los mineros depositan una cantidad de criptomoneda para adquirir las papeletas y luego por medio de un algoritmo aleatorio se determina los nodos seleccionados para crear el nuevo bloque. Este algoritmo contempla la recompensa con criptomonedas a los nodos que validan el contenido de un bloque.

- **Las principales ventajas de PoS son:**
  - El ahorro de energía en comparación con PoW.
  - No se requiere inversión en máquinas con un gran poder computacional como ocurre en PoW.
- **Los principales inconvenientes de PoS son:**
  - No se considera un algoritmo completamente descentralizado, ya que se elige un conjunto de nodos limitado para verificar el contenido del nuevo bloque.
  - Si se establece una cantidad de criptomoneda muy alta puede hacer que ciertos nodos no tengan acceso a participar en el minado, limitando el acceso a las recompensas únicamente a los nodos que cuentan con una mayor cantidad de criptomonedas.
  - Los nodos que poseen una mayor cantidad de criptomonedas tienen mayor probabilidad de minar bloques. Esta regla incita a los nodos a no gastar el beneficio conseguido, generando un escenario donde los que más tienen son los que más ganan.

- **Soluciones *blockchain* que utilizan PoS:**
  - Stratis[3] (no establece un mínimo para participar en el proceso de validación) (es capaz de procesar entre 33 y 67 transacciones por segundo).
  - *DASH* (requiere un mínimo de 1000 DASH para participar en el minado).

Proof of Importancy (PoI)

Es similar a proof of stake ya que se establece la necesidad de realizar un depósito de la criptomoneda para poder participar en el proceso de crear nuevos bloques. La diferencia radica en el mecanismo utilizado para seleccionar el listado de nodos validadores, ya que no se realiza de forma aleatoria. A cada nodo se le calcula un nivel de importancia con base en las transacciones procesadas, su participación en la red y otros factores, que determinan la selección de los nodos que pueden crear nuevos bloques.

- **Las principales ventajas de PoI son:**
  - No requiere de inversión en aumentar la capacidad computacional.
  - Se premia a los nodos que más participación tienen en la red *blockchain*.
- **Los principales inconvenientes de PoI son:**
  - No se considera un algoritmo completamente descentralizado.
  - Requiere mantener una actividad constante en la red para mantener las posibilidades de minado.
- **Soluciones *blockchain* que utilizan PoI:**
  - *NEM* (es capaz de procesar hasta 4000 transacciones por segundo).

Practical Byzantine Fault Tolerance (pBFT)

Es una variante del algoritmo a prueba de fallos bizantino (BFT), en la que se asume la presencia de nodos maliciosos.

El proceso comienza estableciendo una numeración a los nodos, de manera que se conoce el número de nodos que forman la red. Este algoritmo establece un criterio de aceptación que debe cumplir la transacción para su consolidación. El criterio suele ser superar los 2/3 de confirmaciones iguales para consolidar a transacción.

El sistema en *exonum*[4] funciona como una máquina de 4 estados, donde se genera un mensaje con transacciones que pasa por los siguientes estados:

- *Propose*
- *Prevote*
- *Precommit*
- *Block Commit*

El funcionamiento es el siguiente; se selecciona un nodo que ejerce como líder y el resto son súbditos. El líder envía un mensaje (*propose*) a los súbditos para que se procese un conjunto de transacciones. Los súbditos comprueban el mensaje (*prevote*) y si todo es correcto procesan la transacción (*precommit*) y responden al emisor de la transacción. La transacción se considera consolidada (*block commit*) si el cliente recibe al menos (2/3)+1 resultado iguales.

Este algoritmo establece que los nodos parten del mismo escenario para completar la ejecución de la transacción, o lo que es lo mismo, el saldo inicial de la cuenta origen es el mismo en todos los nodos, para que todos puedan producir la misma respuesta.

El algoritmo tiene implementado las acciones a tomar en el caso que el cliente no reciba las respuestas necesarias.

El líder cambia cada cierto tiempo siguiendo el sistema de *round-robin*.

- **Las principales ventajas de pBFT son:**
  - Ahorro en energía comparándolo con PoW.
  - Las transacciones no requieren confirmación para su consolidación.
- **Los principales inconvenientes de pBFT son:**
  - Requiere de una gran comunicación entre los nodos para verificar que la información recibida es correcta.
- **Soluciones blockchain que utilizan pBFT:**
  - *Exonum*

Delegated Byzantine Fault Tolerance (dBFT)

Este algoritmo[5] establece un sistema de votación para la elección de representantes que son los que participan en la generación de bloques siguiendo el algoritmo de prueba de fallos bizantino (BFT), donde se establece el criterio de confirmaciones a 2/3 del total de nodos para dar por consolidado un bloque.

El proceso comienza con la votación de los representantes. En esta votación participan los poseedores de *tokens* que votan a los representantes que van a ejercer como validadores. Del conjunto de validadores seleccionados, se elige un orador que es el encargado de proponer un bloque con un conjunto de transacciones. El resto de validadores han de comprobar la validez del mensaje y ejecutar las transacciones comunicando el resultado.

Si un bloque supera los 2/3 de respuestas idénticas el bloque se añade a la cadena y se elige otro orador.

Los nodos que quieren ser representantes han de cumplir ciertas condiciones. En algunos casos, los nodos tienen que identificarse para poder participar.

- **Las principales ventajas de dBFT son:**
  - Ahorro en energía en comparación con PoW.
  - Se pueden llegar a procesar grandes volúmenes de transacciones.

- **Los principales inconvenientes de dBFT son:**
  - La red no es descentralizada, se limita a cierto número de nodos validadores.
  - En algunos casos, se conoce la identidad de los nodos validadores.
- **Soluciones *blockchain* que utilizan dBFT:**
  - *NEO* (los representantes han de tener una cantidad de criptomonedas e identificarse entre otros requisitos) (Puede llegar a procesar entre 1000 y 10 000 transacciones por segundo).

Criptografía

La criptografía ha sido y es un elemento clave en la vida de las personas. Durante la Segunda Guerra Mundial se utilizó la criptografía para la codificación y descodificación de la información. El ejército alemán utilizaba una máquina llamada 'enigma' para encriptar sus mensajes, permitiendo al ejército alemán establecer comunicaciones de manera secreta. El método de encriptación fue descubierto por un matemático británico llamado Alan Turing quien es conocido como uno de los precursores de la ciencia de la computación.

La idea de la criptografía de clave pública fue expuesta por Whitfield Diffie y Martin Hellman en 1976 como parte de la presentación del protocolo Diffie-Hellman que permite la distribución de claves utilizando un medio de comunicación inseguro.

La criptografía asimétrica o de clave pública cuenta con dos soluciones que se pueden utilizar para generar el conjunto de claves denominadas pública y privada:

RSA

Fue inventado por Ron Rivest, Adi Shamir y Leonard Adleman en 1977. Se basa en la factorización de dos números primos y en la dificultad para revertir la operación.

Este sistema de encriptado ha sido la base de la encriptación durante décadas. En este tiempo el poder computacional de los ordenadores ha aumentado y eso ha obligado a aumentar el tamaño las claves, haciendo que sean procesos demasiado pesados para ciertos dispositivos.

Curva elíptica

El algoritmo de firma de curva elíptica (ECDSA)[6, 7] o de criptografía de curva elíptica (ECC) nace con la propuesta de los matemáticos Neal Koblitz[8] y Victor S. Miller.

El algoritmo se basa en la formula ($y^2 = x^3 + ax + b$) que describe los puntos de una curva elíptica. El algoritmo hace que sea sencillo multiplicar dos puntos de la curva, pero que sea muy difícil invertir el proceso.

Comparación entre RSA y ECC

Los dos algoritmos (RSA y ECC) cumplen los mismos propósitos, pero lo hacen de manera diferente. Una de las razones que influyen a la hora de utilizar uno u otro es el tamaño de claves que se manejan en cada uno de los algoritmos. En la *(tabla 1)* se muestra el tamaño de claves que ha de tener cada algoritmo para lograr un determinado nivel de seguridad.

La principal ventaja de la criptografía de clave elíptica (ECC) es que requiere un tamaño mucho menor que el de RSA para conseguir el mismo nivel de seguridad.

blockchain utiliza la criptografía de curva elíptica para la firma de transacciones. Al contar con claves de tamaño reducido, se pueden utilizar en un gran número de dispositivos.

Seguridad

Aunque una de las características de blockchain es la seguridad de la información que se guarda en la cadena, esta puede verse comprometida en algunas circunstancias muy definidas que se plantean a continuación:

Doble gasto

Un doble gasto consiste en el uso de una cantidad de criptomoneda dos veces. Esto se consigue alterando la cadena de bloques de dos maneras:

- Enviando transacciones que utiliza el mismo conjunto de criptomonedas y que, por tanto, son excluyentes. Solo puede ocurrir una de las transacciones (*race attack*).
- Enviando una transacción que va a ser rechazada por falta de saldo (*finney attack*).

Security (in bits)	*RSA key length required (in bits)*	*ECC key length required (in bits)*
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511

Tabla 1. Tabla complejidad RSA y ECC[7].

Race attack[9,10]

Consiste en enviar una transacción *A* para la compra de un producto o servicio a un establecimiento que no requiere esperar a la confirmación de la transacción para completar y enviar el pedido. Tras enviar la transacción *A* se genera una transacción *B* donde se transfieren los fondos a otra dirección de la misma persona. Esta transacción *B* con más prioridad se ejecutará antes y la transacción *A* quedará descartada por falta de fondos.

Finney attack[9,10,11]

Este ataque requiere de la complicidad de un nodo. En este caso el nodo ha de generar un bloque con una transacción *A* donde se transfiere todo el dinero a otra cuenta del atacante. Cuando se consigue minar el boque, este no se publica y, en ese momento, el atacante genera una transacción *B* a un establecimiento que no necesita que las transacciones se minen para proveer el servicio. Cuando el establecimiento confirma el envío, el atacante publica el bloque minado con su transacción, haciendo que la transacción *B* sea descartada por falta de fondos.

Ataque del 51 %[9,10]

Consiste en controlar más del 50 % del poder computacional de la red para que se valide el contenido de un bloque donde se ha realizado un doble gasto.

Una de las ventajas de *blockchain* es que todas las operaciones son visibles, esto hace que las empresas que trabajan con el *blockchain* como casas de cambio y la comunidad que soporta la red realicen revisiones periódicas sobre las operaciones validadas, con la finalidad de detectar anomalías en la cadena de bloques.

Aunque un ataque del 51 % consiguiera validar una operación de doble gasto, existen muchos observadores que detectarían el doble gasto, esto permitiría tomar acciones para revertir la operación del doble gasto o para minimizar su impacto.

Ethereum classic[12, 13] fue víctima de varios ataques del 51 % en 2020 debido en gran medida a una considerable pérdida de nodos en la red, propiciada por la baja cotización de la criptomoneda frente al dólar y por el coste energético de la minería.

Anteriormente en 2019[14] también en *ethereum classic* se detectó un *pool* de minería con una gran capacidad de cómputo que, en algunos casos, llegó a superar el 50 % del cómputo total de la red. Un *pool* de minería es un sistema donde varios ordenadores trabajan en paralelo para resolver el reto de la prueba de trabajo.

A consecuencia de estos ataques muchas empresas dedicadas al cambio de divisas congelaron sus operaciones con esta divisa o establecieron tiempos muy altos para aceptar las operaciones.

Smart contracts

Los *blockchain* que soportan *smart contracts* pueden verse afectados por un *smart contract* mal programado, de manera que, si alguien encuentra un fallo en el código de un *smart contract,* puede explotar este fallo para modificar la información asociada al *smart contract* o en el peor de los casos traspasar criptomonedas gestionadas por el *smart contract* a una cuenta o hacerlas desaparecer.

Cualquiera de estos casos el problema no es la tecnología *blockchain*, sino el código que se ha instalado en *blockchain* y que contenía algún tipo de vulnerabilidad. Cuando ocurre este tipo de ataques, se tiende a pensar que *blockchain* no funciona, pero el caso es que blockchain ha ejecutado un código de un *smart contract* con fallos, la responsabilidad de este problema recae en los desarrolladores y auditores que participaron en la creación del *smart contract*.

Muchos de los contratos inteligentes se hacen públicos antes de su despliegue para que la comunidad pueda evaluar su comportamiento y detectar posibles vulnerabilidades.

Gestión de claves

Un aspecto de la seguridad importante en *blockchain* es la gestión del par de claves, en concreto la clave privada ha de guardarse en un lugar seguro donde nadie tenga acceso a ella. Si alguien consigue obtener la clave privada de otra persona, puede hacer transacciones y, por tanto, transferir el saldo a otra dirección.

Hay que tener en cuenta que si se pierde la clave privada o si olvidas el *password* que permite desbloquear la clave privada, no hay manera de poder recuperar la clave y, por tanto, el control de la dirección *blockchain* asociada. Esta circunstancia ha pasado en numerosas ocasiones y existen direcciones con saldo, de personas que han perdido las claves y que no pueden operar con esta dirección.

Es aconsejable mantener las claves fuera del entorno de internet para evitar ser víctima de un hacker. Una opción es utilizar dispositivos como pendrives específicos para almacenar claves. Algunos de estos dispositivos requieren la huella dactilar o un código para poder acceder a la clave guardada. Estos dispositivos son menos accesibles para los hackers al no estar conectados a internet.

Resumen

En este tema se han explicado tres elementos clave de la tecnología blockchain, los algoritmos de consenso, la criptografía y la seguridad, estos permiten modelar una tecnología que necesita de estos elementos para poder generar un entorno de intercambio de información accesible y distribuido.

- **Algoritmos de consenso:**

  La tecnología *blockchain* requiere de un mecanismo que permita seleccionar el minero o mineros que generan o validan los bloques que se incluyen en la cadena. Estas y otras funciones son realizadas por el algoritmo de consenso.

  Todos los algoritmos de consenso tienen sus ventajas y sus inconvenientes, por lo que no existe una solución única para *blockchain*. Cada implementación hace uso del algoritmo de consenso que mejor servicio puede proporcionar a cada uno de los *blockchain*.

  Los algoritmos más importantes son:

  - **PoW:** prueba de trabajo
  - **PoS:** prueba de participación
  - **PoI:** prueba de importancia
  - **PBFT:** tolerancia práctica de errores bizantinos
  - **dBFT:** tolerancia de errores bizantinos delegada

- **Criptografía:**

  La criptografía es otro de los elementos principales de *blockchain*, que utiliza algoritmos asimétricos de curva elíptica para generar las claves.

  Este tipo de algoritmos utilizan un tamaño de claves más reducido que el algoritmo de RSA para lograr el mismo nivel de seguridad, lo que permite que puedan utilizarse en un gran número de dispositivos.

- **Seguridad:**

  Otro de los principales valores de *blockchain* es la seguridad. Desde el comienzo de *bitcoin*, *blockchain* ha demostrado ser una tecnología robusta y segura. La gran mayoría de los problemas de seguridad que se han vinculado con la tecnología *blockchain* ocurren fuera del entorno de la tecnología blockchain, pero no por ello se han de descuidar.

  La tecnología *blockchain* registra todas las operaciones consolidadas y esto es vital para poder monitorizar el correcto funcionamiento de la red. Si se detecta algún fallo o incorrección, la comunidad tiene el poder para revisarlo y corregirlo. Este comportamiento no se produce en otros ámbitos de los negocios, por lo que *blockchain* ofrece mayores garantías que los sistemas tradicionales.

  La gran mayoría de los ataques descritos se resuelven implementando una buena política de confirmación de las operaciones con criptomonedas. Estas políticas dependen de los comercios o plataformas que aceptan operaciones de pago con criptomonedas.

Bibliografía

1. H. Anwar, *Consensus algorithms: the root of blockchain technology.* 2018 [En línea]. Disponible en: https://101Blockchains.com/consensus-algorithms-Blockchain/
2. D. Hernandez Lopez, *PBFT-based consensus algorithms for blockchain: a case study.* 2020 [En línea]. Disponible en: https://upcommons.upc.edu/bitstream/handle/2117/328029/PBFT-BASED%20CONSENSUS%20ALGORITHMS%20FOR%20BLOCKCHAIN%20A%20CASE%20STUDY.pdf
3. Stratis FAQ, *What is stratis?* [En línea]. Disponible en: https://stratisfaq.com/what-is-stratis
4. Exonum, *Exonum's custom blockchain consensus algorithm.* 2019 [En línea]. Disponible en: https://medium.com/meetbitfury/our-custom-Blockchain-consensus-algorithm-37db107d006f
5. C. Comben, *Explicación de la tolerancia a errores bizantina delegada (dBFT).* 2019 [En línea]. Disponible en: https://coinrivet.com/es/delegated-byzantine-fault-tolerance-dbft-explained/
6. Webedia Brand Services, *La criptografía de curvas elípticas: segura mientras la computación cuántica lo permita.* 2019 [En línea]. Disponible en: https://www.xataka.com/espacioutad/criptografia-curvas-elipticas-segura-computacion-cuantica-permita
7. A. Russell, *¿Qué es la criptografía de curva elíptica (ECC)?* 2019 [En línea]. Disponible en: https://www.ssl.com/es/faqs/%C2%BFQu%C3%A9-es-la-criptograf%C3%ADa-de-curva-el%C3%ADptica%3F/
8. Neal Koblitz. 2020 [En línea]. Disponible en: https://sites.math.washington.edu/~koblitz/
9. *Bitcoin: mitigating attacks.* 2020 [En línea]. Disponible en: https://www.tutorialspoint.com/blockchain/bitcoin_mitigating_attacks.htm
10. C. Evett, *Blockchain security.* 2016 [En línea]. Disponible en: https://www.simplexityanalysis.com/blog/2016/9/20/blockchain-security
11. M. Rosenfeld, *What is a Finney attack?* 2012 [En línea]. Disponible en: https://bitcoin.stackexchange.com/questions/4942/what-is-a-finney-attack
12. E. David, *Ethereum Classic sufre otro ataque del 51 %.* 2020 [En línea]. Disponible en: https://es.cointelegraph.com/news/ethereum-classic-suffers-another-51-attack
13. B. Pirus, *Dos ataques a la red de ETC dejan a la comunidad necesitando una solución, y que sea rápida.* 2020 [En línea]. Disponible en: https://es.cointelegraph.com/news/two-attacks-on-etc-network-leave-community-needing-a-solution-fast
14. CoinNess, *Exclusive: one ETC private pool claimed over 51 % hashrate.* 2019 [En línea]. Disponible en: https://www.coinness.com/news/198264