

**Universidad de Buenos Aires
Facultad de Ciencias Económicas,
Cs. Exactas y Naturales e Ingeniería**

Carrera de Especialización en Seguridad Informática

Trabajo Final de Especialización

Bitcoin: Funcionamiento y Análisis de Seguridad

Autor: David Lajeunesse

Tutor: Dr. Hugo Scolnik

2020

Cohorte 2019

Declaración jurada de origen de los contenidos

“Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual”.

David Lajeunesse

Pasaporte: GL086416

Resumen

El presente trabajo de investigación recopila información referente al sistema de criptomoneda Bitcoin. Se inicia con una descripción general de Bitcoin, que está basado en la tecnología de cadena de bloques. Sigue una presentación de las primitivas criptográficas que soportan el sistema: funciones hash, árboles de Merkle y curvas elípticas. Luego se analiza el funcionamiento de Bitcoin según el marco propuesto por la Universidad de Cambridge en el documento *Distributed Ledger Technology Systems: A conceptual framework* [1] para los sistemas basados en la tecnología de registro distribuido (DLT), el cual divide el sistema en tres capas: protocolo, red y datos. Posteriormente, se presentan las principales vulnerabilidades de Bitcoin y los ataques probables al sistema, tanto como las contramedidas posibles o ya implementadas según el caso. Finalmente, se exponen los elementos importantes que influirán en el éxito futuro de Bitcoin y se hace una valoración sobre su futuro.

Palabras claves: Bitcoin, criptomoneda, cadena de bloques, criptografía, seguridad, vulnerabilidades, ataques, contramedidas.

Tabla de contenido

INTRODUCCIÓN	11
1 DESCRIPCIÓN GENERAL DE BITCOIN.....	13
2 HERRAMIENTAS CRIPTOGRÁFICAS.....	17
2.1 Funciones Hash	17
2.2 Árboles de Merkle	21
2.3 Curvas Elípticas	22
3 CAPA DE PROTOCOLO	27
3.1 Componente de Génesis	27
3.1.1 Dependencias entre sistemas	27
3.1.2 Creación del Código Base.....	27
3.1.3 Iniciación de Reglas	28
3.2 Componente de Alteración.....	28
3.2.1 Gobernanza del Protocolo.....	28
3.2.2 Cambios en el Protocolo	29
4 CAPA DE RED	33
4.1 Componente de Comunicación	33
4.1.1 Acceso a la Red	33
4.1.2 Difusión de Datos.....	36
4.1.3 Iniciación de Transacción.....	36
4.2 Componente de Procesamiento de Transacciones	38
4.2.1 Propuesta de Registro.....	38
4.2.2 Reglas de Resolución de Conflictos.....	42
4.2.3 Incentivos para el Procesamiento de Transacciones	43
4.3 Componente de Validación	45
4.3.1 Validación de Transacción	45
4.3.2 Validación de Registro	50
4.3.3 Finalización de Transacción.....	52
5 CAPA DE DATOS	55
5.1 Componente de Operaciones	55

5.1.1	Entradas.....	55
5.1.2	Transacción Ejecutable Mediante Programación	55
5.1.3	Lotes de Ejecución	56
5.2	Componente de Libro Mayor.....	57
5.2.1	Referencia.....	57
6	SEGURIDAD	59
6.1	Ataque del 51%.....	59
6.2	Ataque Sybil	60
6.3	Ataque Eclipse	60
6.4	Ataque de Doble Gasto.....	61
6.5	Denegación de Entrega de Bloques y Transacciones	62
6.6	Minería Egoísta.....	64
6.7	Combinaciones de Ataques	65
6.7.1	Doble gasto n-confirmaciones mediante un ataque eclipse	65
6.7.2	Ataque del 51% mediante un ataque eclipse	67
6.7.3	Doble gasto 0-confirmación mediante la denegación de entrega....	68
6.7.4	Doble gasto 1-confirmación mediante la denegación de entrega....	68
6.7.5	Minería egoísta y doble gasto	69
6.7.6	Minería egoísta y denegación de entrega de bloque	69
6.8	Preocupaciones Adicionales	70
6.8.1	Nodos Livianos.....	70
6.8.2	Privacidad	71
6.8.3	Maleabilidad de transacciones	71
6.8.4	Ataques contra la ECC.....	72
7	EL FUTURO DE BITCOIN	73
7.1	Marco Legal de las Criptomonedas.....	73
7.2	Criptomonedas Alternativas	74
7.3	Mejora de las debilidades de Bitcoin	75
	CONCLUSIÓN	77
	REFERENCIAS ESPECIFICAS	81
	REFERENCIAS GENERALES	89

Tabla de Figuras

Figura 1: Procesamiento de una transacción en Bitcoin.....	15
Figura 2: Visión global de la cadena de bloques Bitcoin y de sus componentes.....	16
Figura 3: Esquema iterativo de la función SHA-256 basado en la construcción de Merkle–Damgård.....	19
Figura 4: Una ronda de la función de compresión de SHA-256.....	19
Figura 5: Representación de un árbol de Merkle.....	21
Figura 6: Camino de Merkle (en azul) de la transacción K.....	22
Figura 7: Representaciones gráficas de curvas elípticas y del proceso de duplicación de puntos.....	23
Tabla 1: Parámetros de la curva elíptica secp256k1.....	23
Figura 8: Representación gráfica de la firma ECDSA y su verificación.....	25
Tabla 2: Nivel de seguridad de las curvas elípticas en comparación a RSA/DSA....	25
Figura 9: Soft fork.....	30
Figura 10: Hard fork.....	31
Figura 11: Representación de la red extendida de Bitcoin.....	35
Tabla 3: Formato estándar de una transacción.....	37
Tabla 4: Estructura de una entrada en una transacción.....	37
Tabla 5: Estructura de una salida en una transacción.....	37
Tabla 6: Estructura de un bloque en Bitcoin.....	39
Tabla 7: Componentes de la cabecera de un bloque.....	40
Figura 12: Verificación del script de desbloqueo en una transacción P2PKH.....	48
Figura 13: Uso del camino de Merkle para comprobar una transacción en la SPV..	49
Figura 14: Anuncio de bloques y transacciones en Bitcoin.....	63
Figura 15: Esquema de un doble gasto n-confirmaciones mediante un ataque eclipse.....	67
Figura 16: Esquema de un ataque del 51% mediante un ataque eclipse.....	67
Figura 17: Esquema de un doble gasto 0-confirmación mediante la denegación de entrega de transacción.....	68
Figura 18: Esquema de doble gasto 1-confirmación mediante una denegación de entrega de transacción.....	69
Figura 19: Ventaja de un minero egoísta denegando la entrega de bloques	70
Tabla 8: Propiedades de Bitcoin según un marco para sistemas DLT	78

Agradecimientos

A mi familia y amigos por su apoyo incondicional durante la especialización.

A mis padres Louise y Réjean, mi tía Lucie, mi tío Yves y mi gran amigo Louis-Philippe por visitarme desde Canadá.

A mis compañeros de la especialización, especialmente Pablo y Alex, por su eterna paciencia y por compartir sus conocimientos conmigo.

Al cuerpo docente de la especialización por su dedicación.

A Pedro Hecht por su valioso apoyo académico.

A mi tutor Hugo Scolnik por su gran disponibilidad, su aporte a la corrección y mejora del documento, y por compartir sus conocimientos conmigo.

A todos aquellos que he olvidado y que merecen ser agradecidos

Introducción

Con la publicación del *whitepaper Bitcoin: A Peer-to-Peer Electronic Cash System* en 2008 por Satoshi Nakamoto y luego su implementación exitosa, por primera vez un diseño de criptomoneda logró reunir de manera prometedora el concepto de descentralización con los de seguridad y privacidad. Hasta el día de hoy, ninguna otra criptomoneda ha tenido tanto éxito. Si bien otros esquemas han trazado el camino, como e-cash¹, hashcash², b-money y BitGold³, ninguno ha logrado lo que Bitcoin logró, es decir, establecer un sistema independiente de toda autoridad central confiable [2]. Las fundaciones criptográficas en las que se basa Bitcoin dieron nacimiento al concepto de cadena de bloques, el cual encuentra sus raíces en la tecnología de registro distribuido (DLT⁴) [2]. El concepto de DLT fue introducido por primera vez en 1982 por *Lamport et al.* [3], pero despertó poco interés hasta la eclosión de Bitcoin, desde cuando se hacen inversiones significativas en el desarrollo de cadena de bloques y sistemas DLT [1]. La evolución rápida en el entorno de los sistemas DLT llevó al establecimiento de una terminología y de un marco específico para estos sistemas y definidos en el documento *Distributed Ledger Technology Systems: A conceptual framework* [1] de la Universidad de Cambridge.

Este trabajo describe el funcionamiento de Bitcoin según el marco propuesto por la Universidad de Cambridge, y también analiza la seguridad de su sistema, aspecto esencial para su éxito. El objetivo es facilitar la comprensión de Bitcoin como una cadena de bloques y un sistema DLT, así como establecer una referencia para la replicación del análisis a otros sistemas similares. También, los resultados del análisis se pondrán en perspectiva para hacer una valoración sobre el futuro de Bitcoin.

En cuanto a la estructura del documento, en las dos primeras secciones se presenta una descripción general de Bitcoin y una explicación de sus herramientas criptográficas que son los conceptos básicos necesarios para entender el resto de

¹ <https://en.wikipedia.org/wiki/Ecash>

² <https://en.wikipedia.org/wiki/Hashcash>

³ <https://www.investopedia.com/terms/b/bit-gold.asp>

⁴ *Distributed Ledger Technology* (https://en.wikipedia.org/wiki/Distributed_ledger)

este trabajo. En las secciones siguientes, se describe el funcionamiento del sistema Bitcoin utilizando el marco de DLT mencionado anteriormente, que disecciona el sistema en tres capas: la capa de protocolo (sección 3), la capa de red (sección 4) y la capa de datos (sección 5). La sección 6 está dedicada a la seguridad de Bitcoin: se describen los principales ataques teóricos al sistema, así como sus contramedidas, se presentan ejemplos de ataques combinados, y se mencionan brevemente otros problemas de seguridad acerca de los nodos livianos, la privacidad, la maleabilidad de las transacciones y los ataques contra la criptografía de curvas elípticas (ECC). Finalmente, la sección 7 concluye con una apreciación del futuro de Bitcoin.

Están fuera del alcance de este trabajo los problemas de seguridad asociados con las billeteras y sus implementaciones: aunque actualmente constituyen un riesgo importante en el ecosistema de Bitcoin, son sistemas externos que están integrados en la red Bitcoin y no influyen en las operaciones definidas por el protocolo. Tampoco se analiza la seguridad de otros sistemas externos para ejecutar transacciones, como el *Lightning Network*⁵.

⁵ <https://lightning.network/lightning-network-paper.pdf>

1 Descripción General de Bitcoin

Bitcoin es un sistema de moneda digital pública que se ejecuta encima de Internet a través de una red *peer-to-peer*⁶ (P2P) donde todos los pares (es decir, nodos) están conectados entre sí a través de varias conexiones entrantes y salientes. Los usuarios pueden realizar pagos creando transacciones que transfieren bitcoins (BTC) a otros usuarios. Para enviar y recibir transacciones, los usuarios confían en pares de claves privadas/públicas criptográficas generadas a partir de curvas elípticas. A las claves públicas se aplica una función hash⁷ para formar direcciones anónimas a las cuales enviar fondos, mientras las claves privadas sirven para la autenticación de los usuarios al canjear BTC.

Una vez emitidas, las transacciones son transmitidas a otros pares conectados a la red y validadas por ellos mediante un conjunto de reglas de verificación. Las transacciones válidas se agregan a una lista de espera, denominada *memory pool* (*mempool*), para que se incluyan en un bloque. Algunos nodos, llamados mineros, seleccionan transacciones de su *mempool* y los agrupan en bloques que pueden contener varios cientos de transacciones. Para construir un bloque válido, los mineros tienen que resolver un rompecabezas criptográfico, llamado prueba de trabajo (PoW), para demostrar que han gastado suficientes recursos computacionales en el proceso: esto es un fundamento de la seguridad de Bitcoin, ya que asegura la descentralización del sistema. Los bloques que cumplen la PoW son transmitidos a la red y verificados por otros nodos de la misma manera que las transacciones. Cada nodo agrega los bloques válidos recién recibidos encima de una cadena de bloques que almacenan localmente y que contiene todos los bloques y transacciones válidos anteriores que recibieron. La cadena está formada por los bloques y las transacciones que están vinculados criptográficamente entre sí, y la integridad de los datos de la cadena se protege mediante funciones hash y firmas digitales⁸.

⁶ <https://en.wikipedia.org/wiki/Peer-to-peer>

⁷ https://es.wikipedia.org/wiki/Funci%C3%B3n_hash

⁸ https://es.wikipedia.org/wiki/Firma_digital

Por consenso, la cadena más larga de la red se considera la cadena de bloques pública principal. Debido a que cada bloque de la cadena es vinculado con el anterior, también la cadena de bloques principal es la que lleva la mayor prueba de trabajo, es decir la que gastó la mayor potencia computacional. Una vez que se agrega un bloque a la cadena principal, se considera que las transacciones que lo componen están confirmadas, y cada bloque agregado posteriormente suma una nueva confirmación para las mismas transacciones. Bitcoin considera que una transacción es permanente confirmada después de recibir seis confirmaciones, momento en el que la probabilidad de alteración no es significativa.

Todo el proceso descrito anteriormente, desde la creación de las transacciones hasta la confirmación de los bloques, forma el mecanismo de consenso del sistema, que permite a Bitcoin operar de forma segura sin necesidad de una autoridad central.

En cuanto a los mineros, son nodos responsables de que los datos se incluyan en la cadena de bloques y, por su rol clave en el sistema, su comportamiento honesto es alentado por incentivos económicos. Los mineros pueden ser nodos solitarios minando (es decir, construyendo bloques) por sí solos, o pueden formar *pools* de minería, en los que unen su poder computacional con otros mineros para resolver juntos la PoW y compartir las recompensas, lo que aporta más estabilidad a sus ingresos.

A continuación, la Figura 1 ilustra el procesamiento de una transacción desde su creación hasta su inclusión en la cadena de bloques, mientras la Figura 2 presenta una visión global de la cadena y de los elementos que la componen. Esos elementos se explicarán a lo largo del trabajo, por lo que se alienta al lector a volver a consultar la figura para facilitar su comprensión.

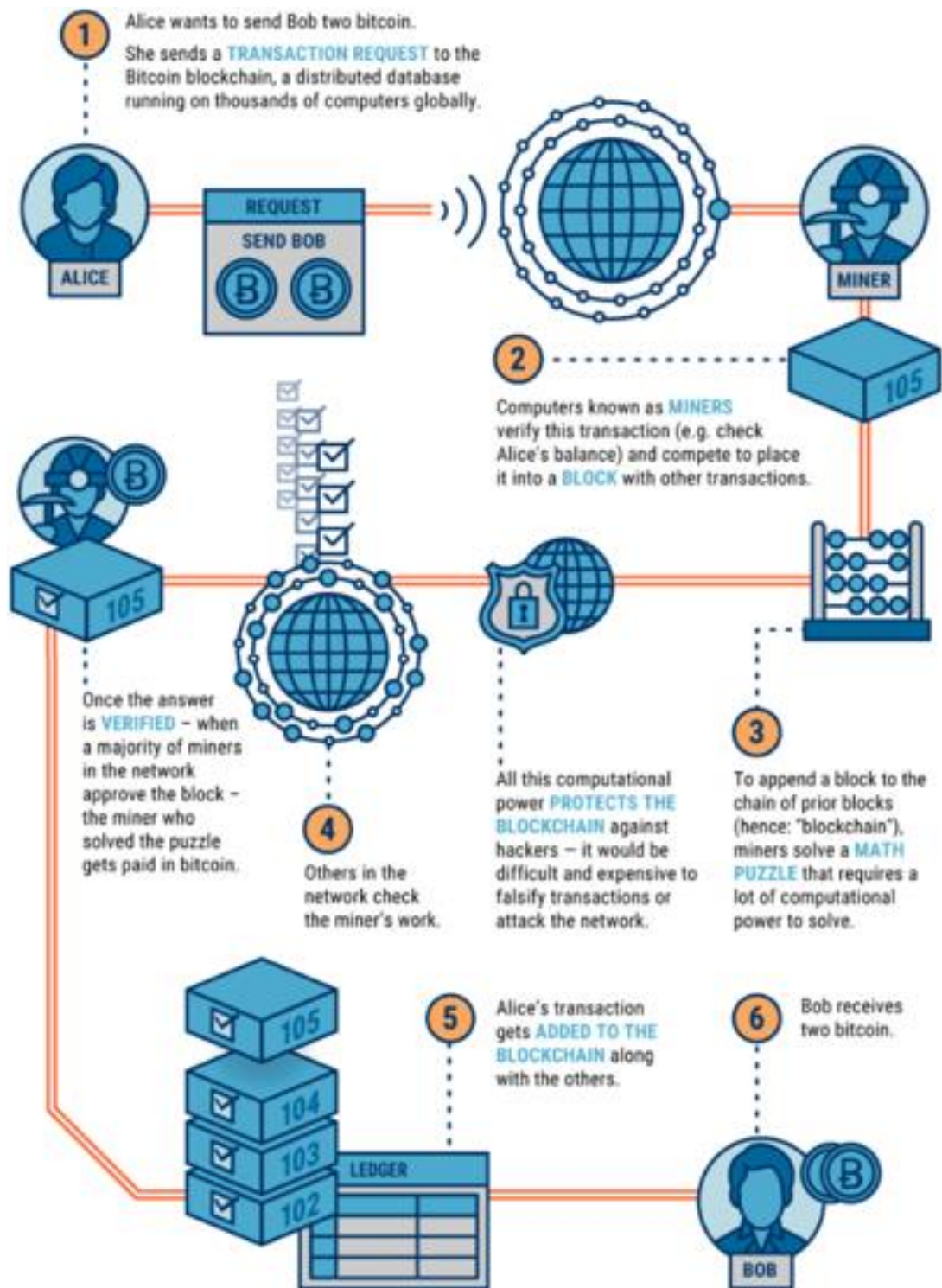


Figura 1: Procesamiento de una transacción en Bitcoin. Fuente: [4]

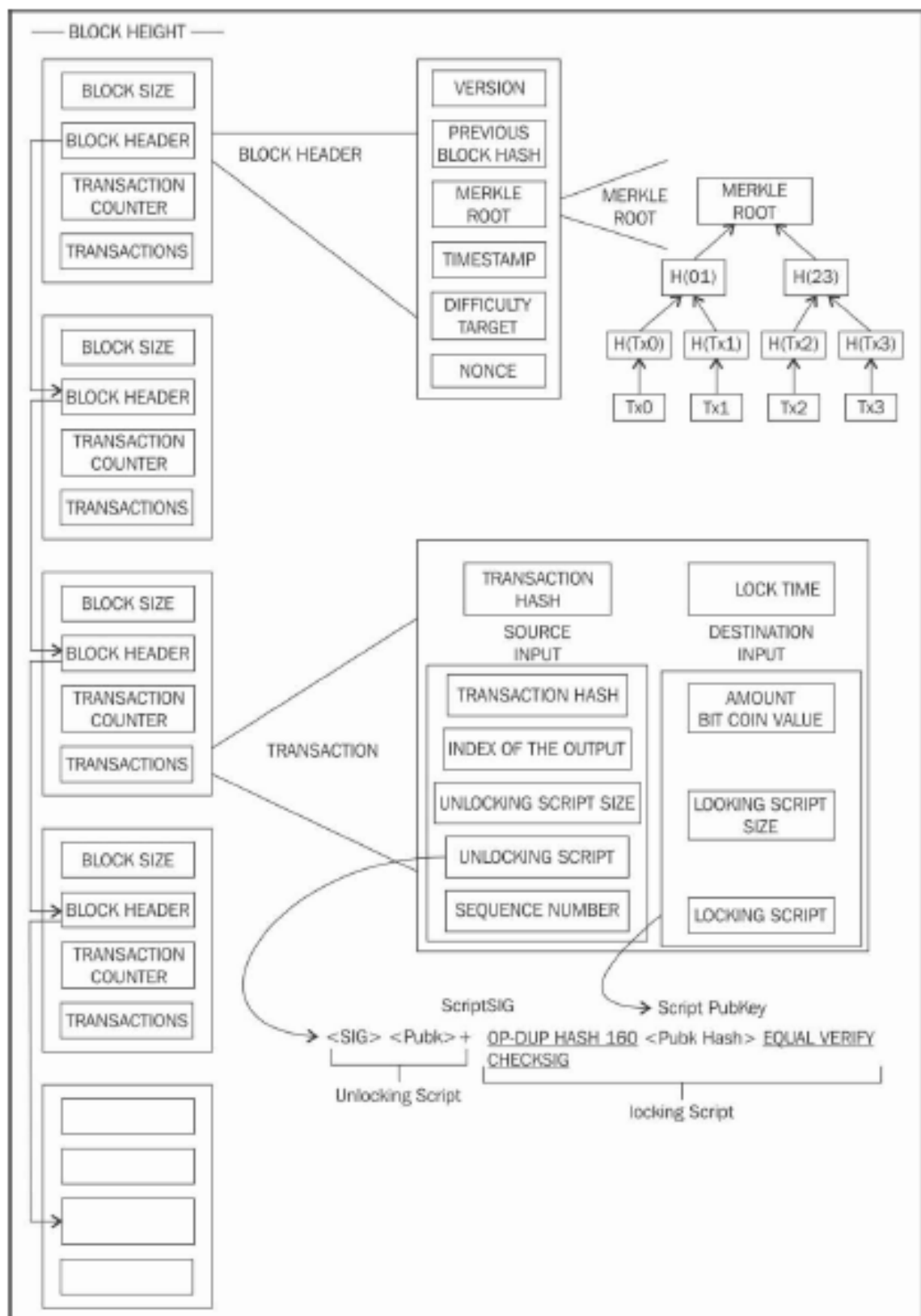


Figura 2: Visión global de la cadena de bloques Bitcoin y de sus componentes. Fuente: [2]

2 Herramientas Criptográficas

Esta sección presenta las primitivas criptográficas subyacentes al sistema Bitcoin y que constituyen la base de su seguridad. La sección 2.1 describe el funcionamiento de la función hash SHA-256 y cómo se utiliza principalmente en Bitcoin para garantizar la integridad de los datos y construir el rompecabezas de la prueba de trabajo. La sección 2.2 muestra cómo los árboles de Merkle sintetizan y realizan un seguimiento eficiente de todas las transacciones incluidas en los bloques, y la sección 2.3 explica la criptografía de curva elípticas (ECC) que se utiliza para generar pares de claves para las direcciones Bitcoin públicas y las firmas digitales.

2.1 Funciones Hash

El rol de una función hash es garantizar la integridad de los datos a los que se aplica el hash, previniendo así su alteración. Bitcoin casi sólo hace uso de la función hash SHA-256⁹, la cual es un estándar del Instituto Nacional de Estándares y Tecnología (NIST)¹⁰ desde 2002 y cumple con los tres requisitos necesarios para que una función hash sea resistente a los ataques criptoanalíticos [5]:

- ♦ *Resistencia a la pre-imagen*

Es computacionalmente inviable encontrar un mensaje m que produce un valor hash h conocido tal que $hash(m) = h$. Esto significa que es una función hash unidireccional.

- ♦ *Resistencia a la segunda pre-imagen*

Dado un mensaje m_1 y su valor hash correspondiente h_1 , de tal manera que $hash(m_1) = h_1$, es computacionalmente inviable encontrar otro mensaje m_2 que produce un hash de mismo valor que m_1 , tal que $hash(m_1) = hash(m_2) = h_1$.

⁹ Excepto para las direcciones Bitcoin, donde se utiliza la función RIPEMD-160 junto con SHA-256.

¹⁰ <https://www.nist.gov>

♦ *Resistencia a la colisión*

Es computacionalmente inviable encontrar dos mensajes diferentes m_1 y m_2 que produzcan un hash de mismo valor, de tal manera que $hash(m_1) = hash(m_2)$. Esto significa que es una función hash unidireccional fuerte.

Si bien la resistencia a la colisión no constituye una protección relevante en el contexto de Bitcoin, la resistencia a la pre-imagen asegura el proceso de extensión de la cadena de bloques (en el que se debe encontrar un hash por debajo de un cierto valor), y la resistencia a la segunda pre-imagen protege la integridad de los datos de la cadena, impidiendo la alteración de un bloque sin modificar el valor de su hash. [6]

El algoritmo SHA-256 sigue un esquema iterativo que corresponde a la construcción Merkle–Damgård¹¹ como se muestra en la Figura 3. Brevemente, toma como entrada un mensaje de longitud máxima de 2^{64} bits y lo divide en bloques de 512 bits¹². El primer bloque se combina con un vector de inicialización¹³ (IV) y se procesa con una función de compresión para generar un resultado intermedio de 256 bits. Este resultado intermedio se combina a su turno con el siguiente bloque de 512 bits y se procesa con la función de compresión para producir otro resultado intermedio de 256 bits, y así sucesivamente hasta que el resultado final de 256 bits de la función hash resulte del procesamiento del último bloque, que contiene el relleno (padding) y la longitud del mensaje. [7]

¹¹ https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction

¹² El mensaje inicial se rellena para alcanzar una longitud que es un múltiplo de 512 bits. La estructura de relleno incorpora la longitud del mensaje original en binario: esto se llama el *fortalecimiento Merkle-Damgård* y garantiza la seguridad del esquema. Para más detalles sobre el esquema de relleno, ver https://es.wikipedia.org/wiki/Esquema_de_relleno.

¹³ El IV de SHA-256 consiste en ocho palabras de 32 bits correspondientes a los primeros 32 bits de la parte decimal de la raíz cuadrada de los primeros ocho números primos.

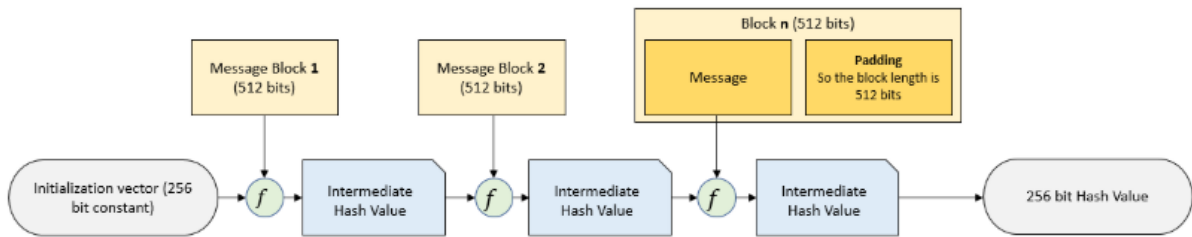


Figura 3: Esquema iterativo de la función SHA-256 basado en la construcción de Merkle–Damgård. Fuente: [8]

La función de compresión SHA-256 consta de 64 rondas. Divide cada bloque de 512 bits en 16 palabras de 32 bits y realiza una gran cantidad de rotaciones y mezclas bit a bit [8], como se muestra en la Figura 4.

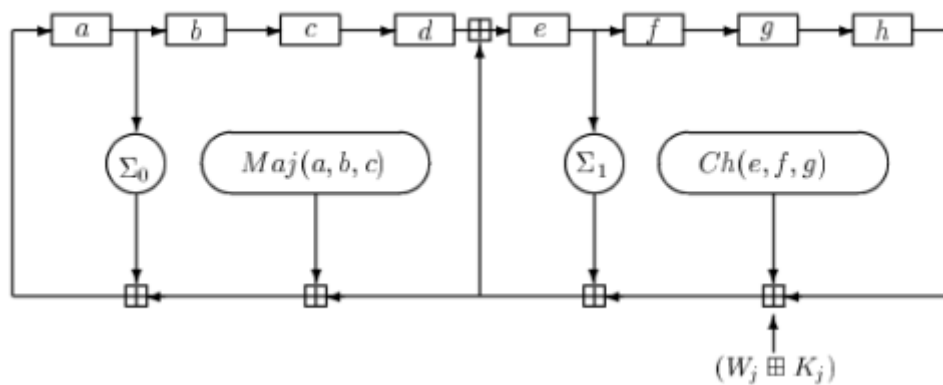


Figura 4: Una ronda de la función de compresión de SHA-256. Fuente: [2]

A continuación, se enumeran las numerosas y diversificadas aplicaciones de SHA-256 en Bitcoin:

- ♦ *Direcciones*

Las direcciones Bitcoin se derivan de un doble hash de una carga útil (por ejemplo, clave pública ECDSA, script), primero con SHA-256 y luego con RIPEMD-160¹⁴ [9] .

- ♦ *Hash de transacción*

Cada entrada de una transacción tiene un campo con un doble hash SHA-256 (llamado SHA-256d) de la transacción que generó el UTXO canjeado [10], lo que

¹⁴ <https://es.wikipedia.org/wiki/RIPEMD-160>

garantiza que este último fue incluido en la cadena de bloques y que el UTXO puede ser gastado.

- ♦ *Árbol de Merkle*

Todos los hashes en los árboles de Merkle (sección 2.2) en Bitcoin son hashes SHA-256d. Cada cabecera de bloque contiene el hash raíz de Merkle, el cual es el hash en la cima del árbol de Merkle y que actúa como una huella digital de todas las transacciones incluidas en el bloque.

- ♦ *Hash del bloque anterior*

Cada cabecera de bloque contiene un hash SHA-256d de la cabecera de su bloque padre (es decir, el bloque anterior en la cadena), vinculando así los bloques entre sí y creando la cadena de bloques.

- ♦ *Prueba-de-Trabajo*

Para crear un bloque válido, los mineros deben calcular un hash SHA-256d de la cabecera del bloque que tenga un valor menor que un cierto nivel de dificultad.

Aunque Nakamoto no explicó el motivo detrás de su decisión, el uso de SHA-256d fue propuesto por Ferguson y Schneier en su libro *Practical Cryptography* para prevenir ciertos tipos de ataques criptográficos contra las construcciones de Merkle-Damgård, llamados ataques *length extension*¹⁵ (citado en [11], [12], [13]). Sin embargo, Craig Wright¹⁶ [13] no está de acuerdo con esta idea por lo que ya existen en Bitcoin otros mecanismos más eficientes para prevenir este tipo de ataques. Más bien, afirma que el uso de SHA-256d permite que el sistema separe las funciones de validación de transacciones y de prueba de trabajo entre múltiples entidades, además de ofrecer la posibilidad de regular el contenido de las transacciones mediante el filtrado de los hashes.

¹⁵ https://en.wikipedia.org/wiki/Length_extension_attack

¹⁶ https://en.wikipedia.org/wiki/Craig_Steven_Wright

2.2 Árboles de Merkle

Los árboles de Merkle son estructuras de datos jerárquicas de hashes y se utilizan en Bitcoin para comprobar la integridad de las transacciones incluidas en un bloque. La base del árbol se compone de las hojas del árbol, las cuales son hashes SHA-256d de todas las transacciones del bloque. Las hojas se concatenan en pares y después se hace un doble hash de cada par para formar la segunda línea del árbol¹⁷. El mismo proceso se repite hasta la cima del árbol, que consiste en un solo hash, llamado la raíz de Merkle, derivado de los hashes de todas las transacciones del bloque, como se puede ver en la Figura 5.

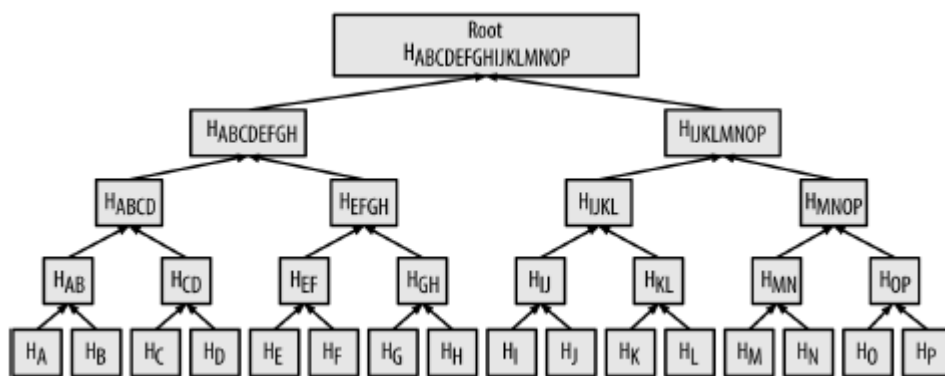


Figura 5: Representación de un árbol de Merkle. Fuente: [9]

La raíz de Merkle es parte de la cabecera de bloque y actúa como una huella digital en el proceso de verificación realizado por los nodos completos¹⁸ de todas las transacciones incluidas en un bloque [9]. Los nodos que no almacenan la cadena de bloques completa (es decir, los nodos livianos¹⁹) también pueden comprobar si una transacción está incluida en un bloque pidiendo a un nodo completo el camino de Merkle de la transacción. El camino de Merkle es el conjunto de los hashes que fueron concatenados con el hash de la transacción y sus padres en el árbol de Merkle, desde la hoja hasta la raíz de Merkle. Este método es eficaz y requiere hasta mil veces menos datos que la cantidad de datos contenidos en un bloque (1 MB) [9].

¹⁷ Un bloque Bitcoin puede contener varios cientos de transacciones. En caso de que haya un número impar de transacciones, la última se duplica y se concatena con sí misma para formar lo que se llama un árbol equilibrado [9].

¹⁸ https://en.bitcoin.it/wiki/Full_node o ver sección 4.1.1 para más información sobre los nodos completos.

¹⁹ Ver sección 4.1.1 para más información sobre los nodos livianos.

Por ejemplo, en la figura siguiente, los hashes en azul forman el camino de Merkle de la transacción en verde.

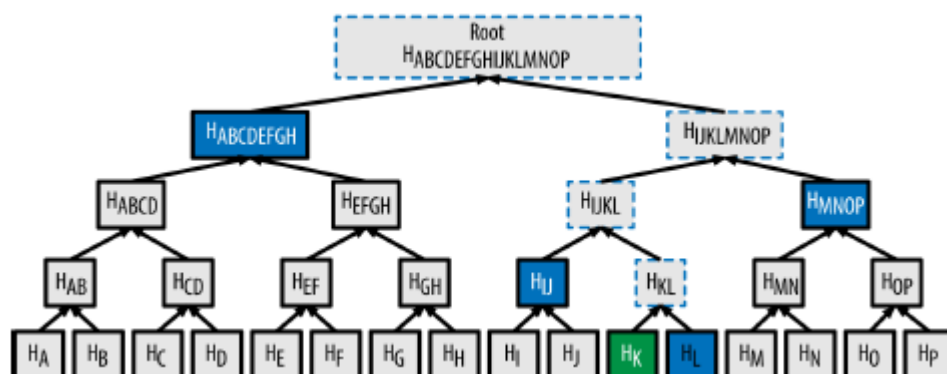


Figura 6: Camino de Merkle (en azul) de la transacción K. Fuente: [9]

2.3 Curvas Elípticas

La criptografía de curvas elípticas (ECC) utiliza curvas de la forma $y^2 = x^3 + ax + b \mod p$, donde p es un número primo. Los puntos que satisfacen la ecuación forman un grupo cíclico y obedecen a una operación de grupo llamada duplicación de puntos, que consiste en multiplicar un entero por un punto de la curva, resultando en otro punto en la curva. Esta propiedad se encuentra en la base del Problema del Logaritmo Discreto de Curva Elíptica (ECDLP), el cual es subyacente a las curvas elípticas sobre campos finitos y proporciona seguridad a la ECC. Específicamente, el ECDLP afirma que dada una curva elíptica sobre un campo finito, un generador G de un subgrupo cíclico y un punto Q de ese grupo cíclico, encontrar un entero d tal que $Q = d \cdot G$ es computacionalmente inviable [14]. Por lo tanto, el entero aleatorio d que multiplica un punto de la curva se puede utilizar como clave privada, y la clave pública se puede definir como el punto Q de la curva resultante de la multiplicación, junto con los parámetros de la curva. Para ayudar a visualizar la ECC, la Figura 7 presenta tres gráficos: el de la izquierda muestra una curva elíptica (secp256k1) sobre los números reales ($y^2 = x^3 + ax + b$), el del centro ilustra cómo el proceso de duplicación de puntos resulta en otro punto en la curva, y el de la derecha muestra el mismo proceso de duplicación, pero aplicado a una curva sobre un campo finito ($y^2 = x^3 + ax + b \mod p$). [2], [15]

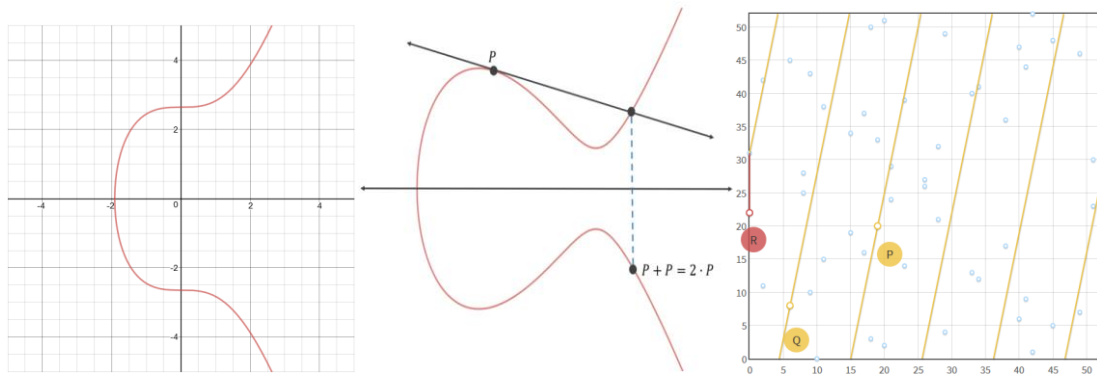


Figura 7: Representaciones gráficas de curvas elípticas y del proceso de duplicación de puntos.
Fuente: [15]

Las curvas elípticas se utilizan en Bitcoin para generar pares de claves:

- ♦ Los hashes de las claves públicas se utilizan como direcciones Bitcoin para el envío de fondos;
- ♦ Las claves privadas se utilizan en el Algoritmo de Firma Digital de Curva Elíptica (ECDSA), el cual sirve para firmar transacciones y garantizar que sólo el legítimo propietario puede gastar los fondos [15].

Para generar pares de claves, Bitcoin utiliza la curva secp256k1 recomendada por los *Standards for Efficient Cryptography* (SEC) [16] y cuyos parámetros son descritos en la Tabla 1.

Parameter	Value	Description
p	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFC2F = 2256 -232 -29 -28 -27 -26 -24 -1	Prime number that specifies the size of the finite field
a	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	Coefficient of the curve equation
b	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007	Coefficient of the curve equation
G	02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 (compressed) 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8 (uncompressed)	Basepoint that generates the subgroup (i.e. generator)
n	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141	Order of the subgroup
h	01	Cofactor of the subgroup

Tabla 1: Parámetros de la curva elíptica secp256k1. [2], [16]

Cuando se crea una transacción Bitcoin, el remitente identifica al nuevo propietario de los fondos por su dirección, la cual es un hash de su clave pública, y sólo él puede gastar los fondos demostrando que posee la clave privada asociada. Para esto, cuando el nuevo propietario crea otra transacción para gastar estos fondos, demuestra su legitimidad firmando la transacción y proporcionando su clave pública. A continuación, la transacción se transmite a la red y es verificada por los nodos receptores: se aplica un hash a la clave pública del remitente para asegurarse de que corresponde a la dirección a la que el propietario anterior envió los fondos y se comprueba que la firma se generó con la clave privada necesaria. La firma y su verificación se realizan de la siguiente manera y son representadas gráficamente en la Figura 8 [2], [15], [17]:

Firma

- 1- Se elige un entero aleatorio r , donde $0 < r < n$
- 2- Se calcula $R = r \cdot G$, donde $R = (x_R, y_R)$ es un punto en la curva
- 3- Se calcula $s = r^{-1}(h(m) + dx_R) \bmod n$
- 4- La firma es (x_R, s)

Verificación de la firma

- 1- Se calculan $u_1 = s^{-1}h(m) \bmod n$ y $u_2 = s^{-1}x_R \bmod n$
- 2- Se calcula $V = u_1 \cdot G + u_2 \cdot Q$, donde $V = (x_V, y_V)$ es un punto en la curva
- 3- La firma es válida si $x_V = x_R$

Donde:

- n y G son los parámetros de la curva descritos en la Tabla 1
- m es la transacción a firmar
- $h(m)$ es el hash de la transacción
- d es la clave privada (es decir, un entero aleatorio)
- Q es la clave pública (es decir, $Q = d \cdot G$)

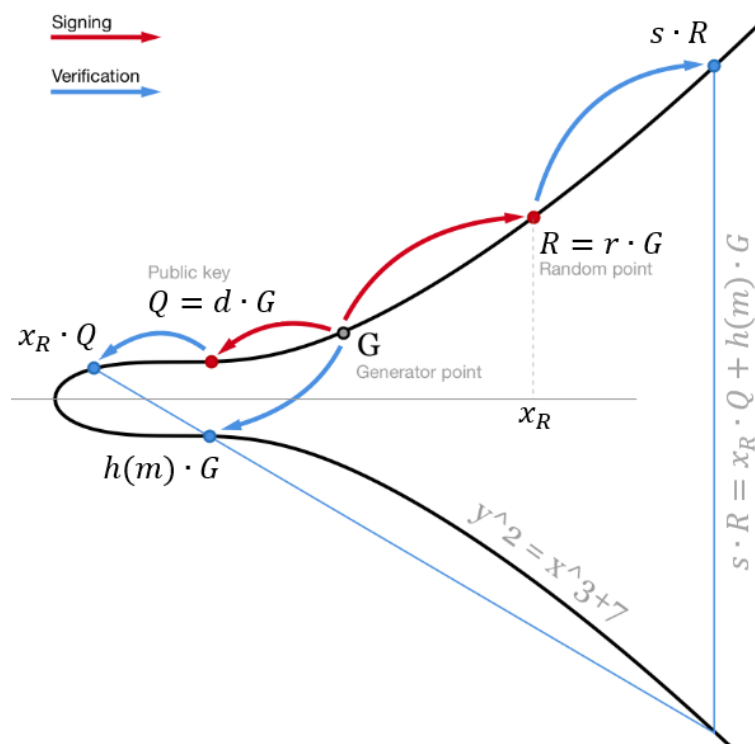


Figura 8: Representación gráfica de la firma ECDSA y su verificación. [18]

Una gran ventaja de ECDSA sobre otros algoritmos como DSA (Algoritmo de Firma Digital) ²⁰ y RSA ²¹ es que ofrece una mejor relación entre el tamaño de la clave y el nivel de seguridad, como lo demuestra la Tabla 2.

Parameters	Strenght	Size	RSA/DSA
secp192k1	96	192	1536
secp192r1	96	192	1536
secp224k1	112	224	2048
secp224r1	112	224	2048
secp256k1	128	256	3072
secp256r1	128	256	3072
secp384r1	192	384	7680
secp521r1	256	521	15360

Tabla 2: Nivel de seguridad de las curvas elípticas en comparación a RSA/DSA. [16]

²⁰ https://en.wikipedia.org/wiki/Digital_Signature_Algorithm

²¹ [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))

Por ejemplo, para un nivel de seguridad de 128 bits, lo que significa que se necesita un promedio de 2^{128} operaciones para atacarlo por fuerza bruta [19], secp256k1 necesita una clave de 256 bits, mientras que RSA y DSA necesitan una clave de 3072 bits [2]. Esta diferencia en el tamaño de las claves permite a ECDSA ahorrar tiempo de cálculo y reducir los requisitos de memoria, dos beneficios importantes para un sistema como Bitcoin [19].

Además, con el fin de prevenir los ataques contra el ECDLP (por ejemplo, *baby-step giant-step* ²², rho de Pollard ²³), es aconsejable elegir el tamaño (en bits) del primo p , y por lo tanto el tamaño de las claves, el doble de grande que el nivel de seguridad deseado (por ejemplo, para un nivel de seguridad de 128 bits, p debería ser de 256 bits) [20].

²² https://en.wikipedia.org/wiki/Baby-step_giant-step

²³ https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm

3 Capa de Protocolo

La capa de protocolo es la base del sistema Bitcoin en la que todos los participantes se ponen de acuerdo. Define, administra y actualiza el conjunto de reglas que rigen el sistema y codifica el diseño de su arquitectura. El protocolo contiene dos componentes: el de génesis y el de alteración. [1]

3.1 Componente de Génesis

El componente de génesis define el sistema en el momento del lanzamiento de la red: su arquitectura, el código base y las reglas de compromiso dentro del sistema, incluyendo el primer registro ("génesis"). [1]

3.1.1 Dependencias entre sistemas

Bitcoin es un sistema autosuficiente que puede funcionar completamente por sí solo sin la necesidad de intervención de ningún sistema externo. Los nodos que componen la red P2P generan, procesan, distribuyen y almacenan los datos del sistema mediante el conjunto de reglas definidas por el protocolo Bitcoin. Dicho esto, Bitcoin permite la interoperabilidad con varios sistemas externos (por ejemplo, las billeteras) que facilitan su uso por parte de los usuarios.

3.1.2 Creación del Código Base

A diferencia de muchas otras criptomonedas que se basan en un derivado del protocolo Bitcoin, este último fue construido desde cero, inspirado por otros esquemas como Hashcash de Adam Back [21] y b-money [22].

3.1.3 Iniciación de Reglas

El conjunto de reglas en el que operará la red Bitcoin es dictado por el cliente Bitcoin Core, también conocido como el cliente de referencia o el "cliente Satoshi"²⁴, y las reglas especificadas en sus versiones tienden a ser adoptadas por las demás implementaciones del cliente. Sin embargo, no hay ninguna especificación de protocolo formal: el protocolo Bitcoin es de código abierto y cualquier persona tiene derecho de proponer y aplicar cambios en el código base [1]. Bitcoin Core se considera la implementación de referencia de Bitcoin, desde la cual todas las demás implementaciones y personalizaciones se derivan [23].

3.2 Componente de Alteración

Este componente establece cómo evoluciona el protocolo con el tiempo. Define el proceso de toma de decisiones sobre el protocolo (gobernanza) y como esas decisiones se traducen en nuevas implementaciones del protocolo (cambios en el protocolo). [1]

3.2.1 Gobernanza del Protocolo

En sus fundamentos, la gobernanza de Bitcoin pretende ser anárquica, lo que significa que las propuestas de cambios en el protocolo se proporcionan y se aprueban de forma cooperativa y voluntaria, debido a la ausencia de una autoridad central [1]. Sin embargo, en realidad, aunque cualquiera puede proponer cambios a través del proceso de Propuesta de Mejora de Bitcoin (BIP), el proyecto Bitcoin Core está controlado por un pequeño grupo de desarrolladores llamados "mantenedores" que tienen autoridad para decidir sobre los cambios en el protocolo. Es decir, los mantenedores pueden decidir unilateralmente si una modificación propuesta en una BIP será adoptada o no en futuros despliegues [24], [25], [26]. Por eso, la gobernanza de Bitcoin podría describirse mejor como jerárquica, debido a que los cambios en el protocolo se basan en el consentimiento de los líderes.

²⁴ Debido a que fue desarrollado y lanzado por primera vez por Satoshi Nakamoto el 9 de enero de 2009.

Los acontecimientos pasados han demostrado este modelo de gobernanza jerárquico. En la versión de cliente Bitcoin 0.8.2, los mantenedores decidieron unilateralmente reducir de 0.0005 BTC a 0.0001 BTC el cargo predeterminado para las transacciones de baja prioridad. Tal control sobre el proceso de modificación del protocolo da claramente a los mantenedores el poder de afectar a toda la economía de Bitcoin [24]. Otro buen ejemplo que muestra la falta de democracia en la gobernanza de Bitcoin se refiere a los eventos que rodean el debate sobre el aumento del límite de tamaño de bloque de 1 MB. Este debate condujo a la salida de varios de los desarrolladores principales originales de Bitcoin que estaban a favor de aumentar el límite²⁵. Algunos alegan que la decisión de no aumentar el tamaño del bloque en el cliente de referencia fue motivada por el deseo de limitar el crecimiento del sistema en beneficio de los *pools* de minería chinos que controlan la mayoría del poder computacional, a expensas de la comunidad Bitcoin. [26]

Esta falta de democracia y la centralización del poder de decisión van en contra de los fundamentos de Bitcoin, que pretende ser un sistema totalmente descentralizado. Esto suscita preocupaciones por la legitimidad del proceso de decisión y por lo tanto sobre la seguridad del sistema, ya que tiende a devolver Bitcoin al esquema de un tercero de confianza.

3.2.2 Cambios en el Protocolo

El código fuente de Bitcoin es abierto, lo que permite a los desarrolladores mantener y apoyar el crecimiento del sistema [24] proponiendo mejoras en el protocolo a través de un proceso estándar de Propuesta de Mejora de Bitcoin (BIP), que puede requerir un consenso de la comunidad y tiene que pasar por varios estados tales como redactado, verificado, aceptado, rechazado o reemplazado [27]. En última instancia, los mantenedores de Bitcoin Core tienen autoridad para decidir sobre los cambios que se aplicarán [1]. La mayoría de los desarrolladores de Bitcoin Core son voluntarios y contribuyen sin que se les pague, mientras que los desarrolladores más experimentados pueden ser patrocinados por una persona o

²⁵ Lo que llevó al lanzamiento de Bitcoin XT, un cliente Bitcoin donde los mineros pueden votar por cambiar el límite de tamaño de bloque.

una empresa para trabajar en el proyecto [28]: por lo tanto, el componente de financiación de Bitcoin se describe como altruista. [1]

El código abierto también permite a cualquiera modificar el código fuente y ejecutar su propia versión del protocolo en la red, creando así una bifurcación en la cadena de bloques. Esto puede resultar en una moneda alternativa si suficientes implementaciones de clientes soportan el mismo conjunto de reglas (por ejemplo, Bitcoin XT [29], Bitcoin Cash [30]) o incluso llevar a una nueva aplicación de cadena de bloques (por ejemplo, Namecoin²⁶). [1]

Cuando se aplican cambios en el cliente de referencia y que los usuarios ejecutan distintas versiones de Bitcoin Core al mismo tiempo en la red, resultan bifurcaciones en la cadena que pueden ser de dos tipos, dependiendo de la naturaleza de los cambios: *soft fork* o *hard fork*.

Soft fork

Un *soft fork* es un cambio a una implementación que es compatible con las versiones anteriores²⁷: los nodos no actualizados pueden seguir realizando transacciones con los nodos actualizados, pero los bloques que violan las nuevas reglas quedan obsoletos por los nodos actualizados. Si una mayoría del poder computacional de la red sigue las nuevas reglas, no habrá divergencia permanente en la cadena porque los nodos no actualizados aceptarán la cadena de los nodos actualizados como cadena principal. [31], [32]

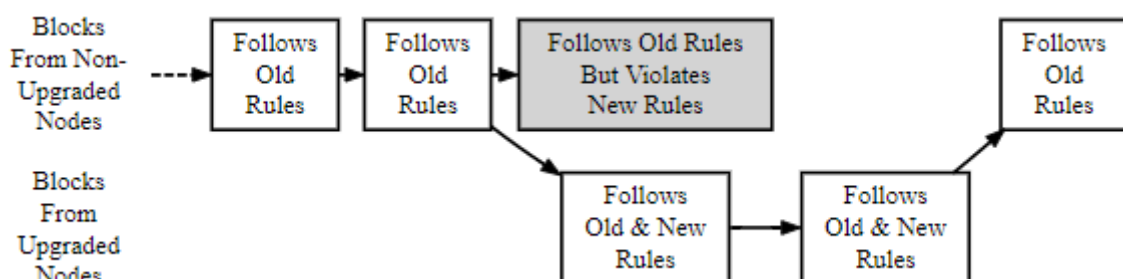


Figura 9: Soft fork. Fuente: [31]

²⁶ <https://www.namecoin.org/>

²⁷ Por ejemplo, una reducción del límite de tamaño de bloque.

Hard fork

Un *hard fork* es un cambio a una implementación que no es compatible con las versiones anteriores²⁸: los nodos deben actualizarse al nuevo protocolo para que no rechacen los bloques formateados con las nuevas reglas. Los nodos no actualizados siguen publicando bloques utilizando el formato anterior y solo aceptan bloques con este formato. Esto resulta en una divergencia permanente de la cadena de bloques donde existen dos versiones simultáneamente, una para los nodos actualizados y otra para los nodos no actualizados, como se ve en la Figura 10. [31], [32]

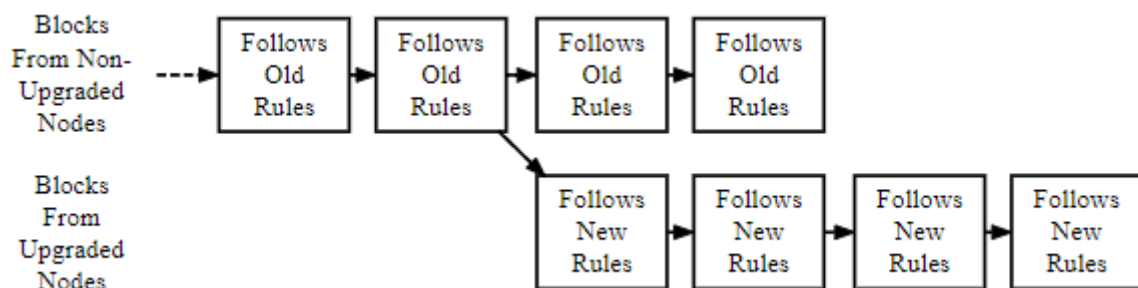


Figura 10: Hard fork. Fuente: [31]

²⁸ Por ejemplo, un aumento del límite de tamaño de bloque o la adición de una nueva característica de transacción.

4 Capa de Red

La capa de red es la implementación práctica de las reglas del protocolo. Establece como se hace la comunicación entre los participantes de la red, como se comparten y procesan los datos, y cómo se validan para ser parte del libro mayor. Contiene tres componentes principales: comunicación, procesamiento de transacciones y validación. [1]

4.1 Componente de Comunicación

Especifica quien puede participar y acceder a la red (abierta o cerrada), cómo se comparten los datos entre los participantes (público o privado) y quien tiene la autorización para iniciar transacciones (sin restricciones o restringidas). Se divide en las subsecciones acceso a la red, difusión de datos e iniciación de transacción. [1]

4.1.1 Acceso a la Red

La red de Bitcoin es una red *peer-to-peer* (P2P) descentralizada y sin restricciones que se ejecuta encima de Internet, en la que cualquier nodo puede conectarse y desconectarse cuando lo desee. La red está formada por todos los nodos que están interconectados a través de conexiones TCP estándares en el puerto 8333. De forma predeterminada, cada nodo puede establecer hasta 8 conexiones salientes y 117 conexiones entrantes [33]. Existen cuatro funciones que pueden realizar los nodos:

- ♦ Proporcionar funciones de **enrutamiento** para descubrir y mantener conexiones a otros nodos;
- ♦ Mantener una copia actualizada de la **cadena de bloques completa**;
- ♦ Realizar transacciones en la red Bitcoin (función de **billetera**);
- ♦ Construir nuevos bloques resolviendo el algoritmo de prueba de trabajo (función de **minería**).

Las funciones que pueden realizar los nodos varían en función de sus tipos. Hay principalmente dos tipos de nodos en la red principal de Bitcoin: los nodos completos y los nodos livianos.

Nodos Completos

Pueden realizar las cuatro funciones y, además, validar y pasar transacciones y bloques [34].

Nodos Livianos

También llamados nodos de Verificación de Pago Simplificada (SPV). No almacenan una copia de la cadena de bloques completa y dependen de los nodos completos para recibir la información necesaria para validar solo las transacciones y los bloques que son relevantes para ellos²⁹, y no los reenvían.

La red principal de Bitcoin consiste en nodos que ejecutan diferentes versiones del cliente Bitcoin Core u otras implementaciones del protocolo Bitcoin P2P, como BitcoinJ³⁰ y Libbitcoin³¹. La red extendida de Bitcoin también incluye nodos que ejecutan protocolos especializados que están conectados a la red principal mediante servidores de *pool* o puertas de enlace de protocolo. Estos nodos son principalmente nodos de *pool* de minería y billeteras de nodos livianos. La Figura 11 ilustra la red extendida de Bitcoin con los distintos tipos de nodos, funciones, puertas de enlace y protocolos que contiene. [9]

²⁹ Los nodos livianos reciben un subconjunto de transacciones, filtradas usando filtros de Bloom. Los filtros de Bloom se explican en la sección 4.3.1.

³⁰ <https://bitcoinj.github.io/>

³¹ <https://github.com/libbitcoin/libbitcoin-system>

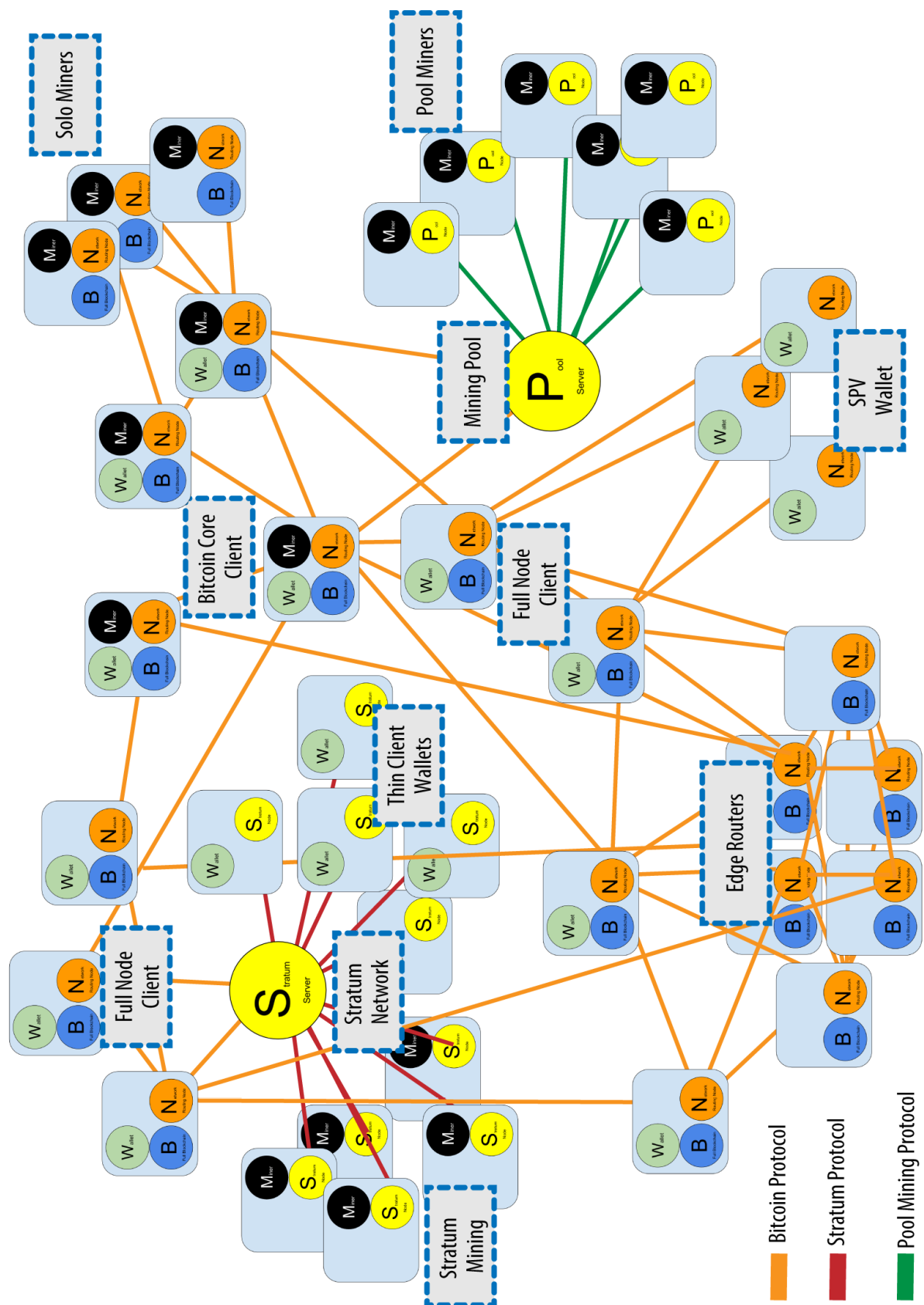


Figura 11: Representación de la red extendida de Bitcoin. Fuente: [9]

4.1.2 Difusión de Datos

Todos los datos se difunden universalmente a la red Bitcoin, lo que significa que se transmiten y se pasan a todos los nodos. Los datos se propagan a través de la red en un formato estandarizado en forma de transacciones y bloques [1], los cuales se componen de transacciones. Una vez que una transacción se ha creado y firmado con la clave privada del remitente, se transmite a al menos un nodo en la red para asegurarse de que se propaga. Después de validar la transacción, cada nodo receptor la envía a tres o cuatro nodos más a los que esté conectado, y al final todos los nodos conectados a la red reciben la transacción en unos segundos. De lo contrario, si la transacción no es válida, los nodos receptores la rechazan y detienen su propagación, evitando así ataques al sistema, como spam y denegación de servicio (DoS). En cuanto a la validación de transacciones, se explica más detalladamente en la sección 4.3.1. [9]

Una vez que una transacción válida llega a un nodo minero, este último puede incluirla en un bloque, que se transmitirá a la red de la misma manera que una transacción después de haber completado la prueba de trabajo.

Dado que los datos transmitidos a la red no contienen ninguna información sensible, no se cifran y se pueden transmitir públicamente a través de cualquier red siempre y cuando llegue a un nodo Bitcoin que los propagarán. [9]

4.1.3 Iniciación de Transacción

La generación de transacciones no está restringida, lo que significa que cualquier persona pueda crear una transacción [1] utilizando un software de billetera o alguna otra interfaz. Para ser válida y propagada, se debe respetar la estructura estándar de una transacción, presentada en la Tabla 3. [9]

Size	Field	Description
4 bytes	Version	Specifies which rules this transaction follows
1–9 bytes (VarInt)	Input Counter	How many inputs are included
Variable	Inputs	One or more transaction inputs
1–9 bytes (VarInt)	Output Counter	How many outputs are included
Variable	Outputs	One or more transaction outputs
4 bytes	Locktime	A Unix timestamp or block number

Tabla 3: Formato estándar de una transacción. Fuente: [9]

Una transacción debe contener al menos una entrada y una salida. El campo *Version* especifica el conjunto de reglas que se deben seguir para validar la transacción, mientras el campo *Locktime*, cuando se usa, especifica el número del bloque o el momento en el cual la transacción se convierte en válida y se puede agregar a la cadena de bloques. [24]

La Tabla 4 y la Tabla 5 detallan respectivamente la estructura de una entrada y de una salida en una transacción.

Size	Field	Description
32 bytes	Transaction Hash	Pointer to the transaction containing the UTXO to be spent
4 bytes	Output Index	The index number of the UTXO to be spent; first one is 0
1-9 bytes (VarInt)	Unlocking-Script Size	Unlocking-Script length in bytes, to follow
Variable	Unlocking-Script	A script that fulfills the conditions of the UTXO locking script.
4 bytes	Sequence Number	Currently disabled Tx-replacement feature, set to 0xFFFFFFFF

Tabla 4: Estructura de una entrada en una transacción. Fuente: [9]

Size	Field	Description
8 bytes	Amount	Bitcoin value in satoshis (10^{-8} bitcoin)
1-9 bytes (VarInt)	Locking-Script Size	Locking-Script length in bytes, to follow
Variable	Locking-Script	A script defining the conditions needed to spend the output

Tabla 5: Estructura de una salida en una transacción. Fuente: [9]

Una entrada apunta a una transacción anterior mediante el hash de la transacción (creando así una cadena de transacciones ordenadas) y gasta sus Fondos No Gastados (UTXO) especificadas en el campo *Output Index*. Una salida y una entrada de transacción contienen respectivamente un script de bloqueo y un script de desbloqueo que sirven para validar que el remitente puede legítimamente gastar los UTXO a las que se hacen referencia. El script de bloqueo indica las condiciones que el script de desbloqueo tiene que cumplir para gastar la salida. Esto es parte del proceso de validación de las transacciones, que se explica en detalles en la sección 4.3.1.

De forma predeterminada, todas las entradas y salidas de una transacción se firman con la clave privada del remitente, después de lo cual, la transacción se transmite a la red. Si la transacción es válida, los nodos que la recibieron la propagarán más, y eventualmente se añadirá a la cadena de bloques a través de una propuesta de registro (es decir, propuesta de bloque).

4.2 Componente de Procesamiento de Transacciones

Este componente establece cómo se procesan las transacciones después de su creación hasta llegar al libro mayor: la sección 4.2.1 describe cómo las transacciones se agregan a bloques con el fin de extender la cadena, la sección 4.2.2 explica las reglas de resolución de conflictos en caso de que dos bloques se anuncien simultáneamente, y la sección 4.2.3 trata de los incentivos que alientan los nodos a proponer bloques. [1]

4.2.1 Propuesta de Registro

La propuesta de registro en Bitcoin no requiere permiso, lo que significa que cualquier nodo de la red tiene derecho a proponer un nuevo bloque para extender la cadena [1]. El acto de crear y proponer nuevos bloques se llama minar y se puede hacer de dos maneras: minar en solitario³² o minar en un *poo*³³. Los nodos que llevan a cabo esta tarea se denominan mineros y pueden ser nodos completos o

³² Un nodo construye y propone bloques por sí solo y recibe todas las recompensas.

³³ Los nodos unen su poder computacional para resolver la PoW y comparten las recompensas.

incluso nodos livianos que participan en un *pool* de minería y que cuentan con un servidor de *pool* para mantener un nodo completo. En cualquier caso, los requisitos para proponer un bloque válido son los mismos, sin importar que sea un nodo completo, un nodo liviano, un minero en solitario o un *pool* de minería. Las diferencias residen más bien en la eficiencia para producir bloques y los mecanismos de recompensa, aspectos explicados en la sección 4.2.3.

Cuando un nodo recibe una nueva transacción de un par en la red, comprueba su validez realizando una serie de verificaciones (explicadas en la sección 4.3.1), y si es válida, el nodo agrega la transacción a su *mempool*, que es una lista temporal de transacciones no confirmadas³⁴ que se encuentra en la memoria local del nodo. Luego, la transacción es incluida en un bloque en función de su prioridad³⁵ y del cargo³⁶ de la transacción. Para ser válido, un bloque debe respetar una estructura estándar, que se explica a continuación.

Estructura de Bloque

La Tabla 6 define la estructura de un bloque en Bitcoin:

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1-9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

Tabla 6: Estructura de un bloque en Bitcoin. Fuente: [9]

Un bloque se compone de una cabecera (*Block Header*), que contiene metadatos, y de una lista de transacciones que se incluirán en la cadena de bloques.

³⁴ Una transacción no confirmada es una transacción válida que aún no se ha incluido en un bloque. Una transacción se convierte en confirmada cuando se incluye en un bloque que ha completado el PoW y que se ha agregado a la cadena de bloques.

³⁵ La prioridad de una transacción se determina como la suma del valor y de la edad de sus entradas dividida por el tamaño total de la transacción: $\text{Prioridad} = \text{Suma}(\text{Valor input} * \text{Edad input}) / \text{Tamaño transacción}$. [9]

³⁶ Las transacciones con cargos más altos son más propensas a ser incluidas en un bloque por los mineros.

El tamaño de la cabecera de bloque es de 80 bytes, y el propio bloque puede contener más de 500 transacciones (la transacción media es de al menos 250 bytes) [9] y alcanzar un tamaño total de 1 MB.

Cabecera de Bloque

La Tabla 7 muestra los componentes de la cabecera de un bloque:

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

Tabla 7: Componentes de la cabecera de un bloque. Fuente: [9]

La versión de bloque especifica el conjunto de reglas que se deben seguir para validar el bloque [2]. Los campos *Previous Block Hash* (hash del bloque anterior) y *Merkle Root* (raíz de Merkle) son hashes SHA-256d [2] que sirven para asegurar la integridad de los datos y para validar transacciones incluidas en el bloque (ver sección 4.3.1). Los campos *Timestamp*, *Difficulty Target* y *Nonce* son metadatos relativos a la prueba de trabajo (PoW), el mecanismo utilizado para llegar a un consenso entre los participantes de la red Bitcoin sobre los bloques para agregar a la cadena. Para crear un bloque válido, cada nodo debe completar la PoW.

Prueba de Trabajo

El mecanismo de prueba de trabajo se utiliza en la red Bitcoin con el fin de lograr un consenso entre los pares sobre los bloques válidos para extender la cadena. La PoW sólo se puede completar mediante la búsqueda por fuerza bruta y tiene como objetivo elegir aleatoriamente un nodo para cada nuevo bloque

construido, seleccionando el primero que logra la PoW. Este método demuestra que un minero ha gastado suficientes recursos computacionales para construir un bloque válido. Para resolver la PoW, los mineros tienen que encontrar un nonce (número aleatorio en la cabecera del bloque) que produzca un hash³⁷ abajo de un cierto nivel de dificultad³⁸. Si se agota la cantidad de nonces posibles de 32 bits, el campo *Extranonce* de la transacción *coinbase*³⁹ también se puede utilizar para cambiar el valor de la raíz de Merkle y, en consecuencia, generar toda una nueva gama de posibilidades de hashes de la cabecera del bloque.

La idea de este modelo es fijar un nivel de dificultad que resultará en un tiempo medio de generación de bloques de diez minutos. Este intervalo representa un compromiso entre tiempos de confirmación de transacción rápidos y la probabilidad de generar bifurcaciones (las bifurcaciones se explican en la sección 3.2.2). Cada 2016 bloques, lo que corresponde a un tiempo teórico de dos semanas, el nivel de dificultad se reajusta⁴⁰ en función del tiempo de generación de los últimos 2016 bloques generados⁴¹ en la red. Si el tiempo de generación de bloques fue más rápido que diez minutos en promedio, el nivel de dificultad aumentará, y si fue más lento, la dificultad disminuirá. Esta es una característica central de Bitcoin, ya que permite al sistema controlar la frecuencia de emisión de monedas y la velocidad de procesamiento de las transacciones independientemente de la potencia computacional de la red [9].

El hash del bloque anterior en la cabecera de bloque vincula el bloque actual al bloque anterior (es decir, su bloque padre), por lo tanto, haciendo imposible la alteración de cualquier bloque sin modificar todos los bloques posteriores en la cadena de bloques. Junto con la PoW, esta característica contribuye a proteger la red contra la alteración de la cadena, ya que se requeriría suficiente potencia computacional para poder reconstruir la cadena, desde el bloque modificado hasta el último, más rápido que la tasa de crecimiento actual de la cadena. En otras palabras, la capacidad del atacante para producir hashes tendría que superar la del resto de la

³⁷ El hash de la PoW es un SHA-256d.

³⁸ El nivel de dificultad es un número de 256 bits en un formato compacto [71].

³⁹ La primera transacción de un bloque, que recompensa al minero exitoso del bloque anterior.

⁴⁰ Nueva Dificultad = Dificultad anterior * (Tiempo de los últimos 2016 bloques / 20160 minutos) [9].

⁴¹ El tiempo de generación de los últimos 2016 bloques se calcula a partir del campo *Timestamp* de la cabecera del bloque.

red que trabaja en extender la cadena. En la práctica, la probabilidad que tal escenario suceda en una red descentralizada es casi inexistente.

Sin embargo, el defecto de la PoW reside justamente en el hecho de que el control de la cadena está en manos de los que poseen mayor capacidad computacional y es independiente del contenido de las transacciones. En [35], se destaca la vulnerabilidad de Bitcoin a ataques del 51 por ciento⁴², debido a su estado actual basado en la PoW y a la centralización del poder computacional en China (los *pools* de minería chinos controlan 66% [36] de la potencia de la red)⁴³. En la sección 6.1 sobre la seguridad, los ataques del 51 por ciento se analizan más a fondo.

4.2.2 Reglas de Resolución de Conflictos

En la red Bitcoin, los mineros compiten entre sí para crear bloques válidos que se añadirán a la cadena. El algoritmo de la PoW que tienen que resolver se ajusta para generar un nuevo bloque cada diez minutos en promedio. Como hemos visto, este intervalo reduce la probabilidad de una bifurcación en la cadena, que ocurre cuando dos bloques válidos que hacen referencia al mismo padre se anuncian simultáneamente en la red. Pero aun así, las bifurcaciones son inherentes a Bitcoin debido a su naturaleza distribuida. Sin embargo, mientras las bifurcaciones de un solo bloque pueden ocurrir más o menos cada semana, es muy poco probable que ocurran bifurcaciones de dos bloques o más [9].

Si dos bloques válidos que extienden el mismo padre se anuncian en la red dentro de un corto período, cada nodo completo aceptará y extenderá su cadena con el primer bloque que recibe y conectará el segundo en una cadena secundaria. Esto dará lugar a dos versiones de cadena diferentes en la red, dependiendo del bloque que recibió primero cada nodo. Para abordar esta división temporal y hacer

⁴² Este tipo de ataques consiste en controlar la mayoría de la tasa de hash de la red con el fin de alterar el historial de transacciones e impedir que las nuevas transacciones sean confirmadas [70]. Por ejemplo, modificar un bloque en el historial de la cadena con el fin de realizar un ataque de doble gasto.

⁴³ A ese efecto, el sistema alternativo de criptomoneda Dash (<https://www.dash.org/>) pretende solucionar el problema y prevenir los ataques del 51% usando un protocolo, llamado ChainLocks, que se basa en un híbrido PoW/PoS (*Proof of Stake* - <https://www.investopedia.com/terms/p/proof-stake-pos.asp>) [35].

que la red converja de nuevo a una sola versión de la cadena, Bitcoin se basa en la regla de la cadena de bloques más larga, que lleva a los nodos a aceptar como cadena principal la que lleva la mayor cantidad de PoW acumuladas. Al producir una bifurcación, el siguiente bloque válido propagado en la red solo extenderá la versión de la cadena que contiene su bloque padre: esto hará que esta versión sea la más larga y, por lo tanto, la red la considerará como la cadena principal. Consecuentemente, las transacciones incluidas en los bloques de la cadena descartada y que no forman parte de la cadena principal se agregarán nuevamente al *mempool* de los nodos para que se incluyan en un nuevo bloque.

Es muy improbable que una bifurcación causada por anuncios de bloques simultáneos dure más de un bloque, pero en tal caso, la misma regla se aplica hasta que una de las versiones de la cadena finalmente suma un bloque válido antes que la otra, haciéndola más larga y confirmándola como la versión universalmente reconocida por la red.

4.2.3 Incentivos para el Procesamiento de Transacciones

Se alienta a los mineros a producir bloques mediante dos formas de incentivos económicos: los cargos de transacción y las nuevas monedas acuñadas. Ambos tipos de recompensas utilizan los activos nativos (bitcoins) como incentivo para incluir transacciones en bloques y realizar la prueba de trabajo.

Cargos de Transacción

Los cargos de una transacción se establecen como la diferencia entre el valor de todas las entradas y el de todas las salidas. El minero cuyo bloque resuelve el algoritmo de la PoW y se añade a la cadena principal recibe el total de los cargos de transacción incluidos en el bloque. Como mencionamos anteriormente, las transacciones son incluidas en bloques por los mineros en función de sus prioridades (dependiendo de la edad de la entrada, su valor y el tamaño de la transacción) y de los cargos de transacción. Los primeros 50 KB del espacio para las

transacciones en un bloque están reservados para transacciones de alta prioridad⁴⁴, independientemente de sus cargos, y el resto del bloque suele llenarse con las transacciones del *mempool* que conllevan los mayores cargos [9]. Los cargos no son obligatorios ni fijados por el protocolo Bitcoin [2]. Una transacción sin cargo siempre será procesada, pero es más probable que se agregue como una transacción de alta prioridad debido a su edad, después de mucho tiempo de esperar en el *mempool* [9].

Nuevas Monedas Acuñadas

Cuando un minero crea un bloque válido que se añade a la cadena principal, también es recompensado con nuevas monedas creadas. Este es el modelo que diseñó Nakamoto para emitir gradualmente más monedas en el sistema. La recompensa por bloque se calcula a partir de su altura (es decir, su número de bloque en la cadena), empezando con 50 bitcoins por el primer bloque de la cadena (bloque génesis), creado el 3 de enero de 2009 [37], y reducido a la mitad cada 210.000 bloques [9]. Debido a que el tiempo de generación de bloques se controla a través del nivel de dificultad de la PoW, resulta en recompensas de bloque reducidas a la mitad aproximadamente cada 4 años. Desde el 8 de septiembre de 2019, es decir el bloque #593.886, la recompensa por bloque es de 12,5 bitcoins y disminuirá a 6,25 bitcoins al bloque #630.000, que se espera que sea alrededor del 24 de mayo de 2020 [38]. Con un límite establecido a 64 reducciones por la mitad, después de qué no se acuñarán más nuevas monedas [9], se espera que el suministro fijo de Bitcoin de casi 21 millones de bitcoins⁴⁵ se alcance en 2140 [2].

Transacción Coinbase

Los mineros reclaman sus recompensas iniciando una transacción llamada *coinbase* que se incluye en el siguiente bloque. Una transacción *coinbase* es la primera transacción de un bloque, que paga nuevas monedas acuñadas y los cargos

⁴⁴ Alta Prioridad > 100,000,000 satoshis * 144 bloques / 250 bytes = 57,600,000, lo que corresponde a un bitcoin (100M satoshis) de un día de edad (144 bloques) en una transacción de tamaño total de 250 bytes [9].

⁴⁵ Precisamente 20.99999998 millones de bitcoins [9].

de transacción del bloque padre a su minero⁴⁶ [9]. Para evitar que un bloque confirmado se declare más tarde obsoleto y se excluya de la cadena principal, las salidas de las transacciones *coinbase* tienen que esperar 100 confirmaciones antes de ser gastadas [2].

El mecanismo de dar incentivos económicos en forma de activos nativos para el procesamiento de transacciones y la minería de bloques es un elemento clave para reforzar la seguridad de Bitcoin. Como la recompensa anima a los mineros a crear bloques y como tienen que respetar reglas de validación estrictas al hacerlo, previene la aceptación de transacciones fraudulentas por la red, siempre y cuando la mayoría de los nodos sea honesta. Un nodo deshonesto que intenta crear y propagar un bloque fraudulento no solo verá a los nodos honestos rechazarlo, sino que también incurrirá en el costo de la electricidad asociada con la PoW además de perder la recompensa. Junto con el algoritmo de la PoW, que requiere mucho poder computacional para modificar el historial de la cadena, los incentivos económicos evitan ataques de doble gasto: puede ser más rentable para un atacante usar su potencia computacional al servicio de la minería honesta a cambio de la posibilidad de ser recompensado con bitcoins. [2], [9]

4.3 Componente de Validación

Esta sección describe cómo la validación de los datos que componen la cadena de bloques se hace tanto a nivel de transacción (sección 4.3.1) que a nivel de bloque (sección 4.3.2). Además, la sección 4.3.3 establece cuándo las transacciones pueden considerarse confirmadas (es decir, permanentes). [1]

4.3.1 Validación de Transacción

Una vez que se ha creado una transacción, se envía a los nodos vecinos en la red, que verifican de forma independiente su exactitud y la propagan a otros nodos si es válida, o de lo contrario, detienen la propagación y la rechazan de su

⁴⁶ Excepto por el bloque génesis (número de bloque 0), que es el primer bloque de la cadena de bloques Bitcoin y que no paga ningún cargo por no tener un bloque padre.

mempool. Para validar las transacciones, los nodos realizan principalmente las siguientes verificaciones⁴⁷ [9], [39]:

- ♦ Asegurarse de que la sintaxis y la estructura de la transacción son correctas (formato y exactitud de sus campos);
- ♦ Verificar que el script de desbloqueo cumpla las condiciones establecidas por el script de bloqueo;
- ♦ Comprobar que el script de bloqueo coincida con los tipos estándares de script de transacción admitidos⁴⁸;
- ♦ Verificar que la suma de las monedas a las que hacen referencia las entradas coincida con la de las salidas (y los cargos);
- ♦ Verificar que las monedas de entrada no se hayan gastado previamente (comprobando el historial de todas las transacciones ejecutadas en la red);
- ♦ Verificar que las monedas de entrada hagan referencia a transacciones correctas que ya han sido confirmadas en la red (es decir, incluidas en un bloque).

Un nodo minero que no ejecuta el cliente de referencia y que no sigue sus especificaciones todavía puede aceptar como válida una transacción que contiene un script no estándar e incluirla en un bloque [9], pero el bloque no será aceptado ni transmitido por los nodos que ejecutan el cliente de referencia y, por lo tanto, no se incluirá en la cadena [40].

Verificación de Script

Para entender el mecanismo de verificación del script de desbloqueo, tomemos como ejemplo el tipo de script de transacción más común en Bitcoin, *Pay To Public Key Hash* (P2PKH), que se utiliza para enviar una salida a una o varias direcciones Bitcoin [41]. En P2PKH, el script de bloqueo (es decir, el script *pubkey*) contiene [42]:

⁴⁷ Una lista completa de las verificaciones realizadas se puede encontrar en [39].

⁴⁸ El cliente de referencia de Bitcoin solo admite cinco tipos de script de transacción, dependiendo de los requisitos de la transacción [2]. A partir de Bitcoin Core 0.9, los tipos estándares de script *pubkey* son: *Pay To Public Key Hash* (P2PKH), *Pay To Script Hash* (P2SH), *MultiSig*, *Pubkey* and *Null data* [68]. Consultar [41] para obtener más detalles sobre los tipos de script estándar.

- ♦ Un hash de la clave pública del beneficiario (*pubkeyhash*);
- ♦ Algunos operadores a ejecutar para verificar el script de desbloqueo.

En cuanto al script de desbloqueo (es decir, el script *signature* o *sig*), contiene [42]:

- ♦ La clave pública completa del remitente sin hash (*pubkey*);
- ♦ La firma del remitente (*sig*), que es un cifrado con la clave privada del remitente de la mayoría de los datos de la transacción⁴⁹, excepto el propio script *signature*.

En forma de código, los scripts de bloqueo y desbloqueo se expresan de la siguiente manera [2], donde los operadores están en mayúsculas y los parámetros mencionados anteriormente están entre corchetes:

```
ScriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY
OP_CHECKSIG
ScriptSig: <sig> <pubKey>
```

Cuando se llega a la verificación del script de desbloqueo, el script *signature* y el script *pubkey* se concatenan (el script *signature* primero) y las instrucciones se ejecutan siguiendo una lógica basada en pilas (*stack-based* en inglés), como se muestra en la Figura 12 [2]. La ejecución del script comprueba la legitimidad del remitente de la siguiente manera:

- ♦ Verifica que la clave pública del remitente incluida en el script de desbloqueo produce un hash de mismo valor que el hash de la clave pública incluido en el script de bloqueo;
- ♦ Verifica que el remitente es el propietario de la clave privada asociada a la clave pública especificada en el script de bloqueo.

⁴⁹ Los datos firmados incluyen el hash de la transacción anterior (*txid*), el índice de la salida de la transacción anterior (*output index*), el script *pubkey* de la salida anterior, el script *pubkey* creado en la transacción actual y la cantidad de bitcoins para transferir al siguiente destinatario [42].

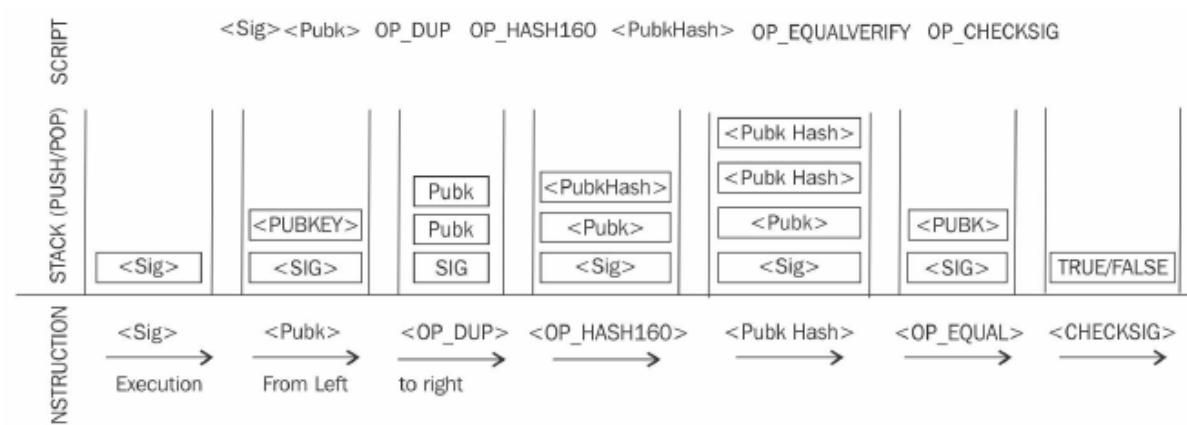


Figura 12: Verificación del script de desbloqueo en una transacción P2PKH. Fuente: [2]

Nodos Livianos

Los nodos livianos necesitan validar ciertas transacciones que son relevantes para ellos. Para cualquier nodo, el proceso de validación de una transacción incluye la comprobación de que ninguna de las entradas se ha gastado anteriormente. Esto requiere comprobar el historial de todas las transacciones ejecutadas en la red. Si bien no es un problema para los nodos completos, que mantienen una copia local completa de la cadena de bloques, los nodos livianos necesitan acceder a la cadena de bloques completa para llevar a cabo esa comprobación. Para ello, Bitcoin utiliza la Verificación de Pago Simplificada (SPV) que fue propuesto por Nakamoto en su *whitepaper* [43].

La SPV solo requiere que los nodos livianos mantengan una copia de las cabeceras de bloque que componen la cadena más larga, lo que se puede obtener consultando nodos de red⁵⁰. Luego, para comprobar un pago, un nodo ligero solicita a un nodo completo el camino de Merkle que vincula la transacción a su bloque en la cadena. De tal manera, se puede probar que la transacción está incluida en un bloque y que el bloque se ha confirmado con la adición de nuevos bloques a la cadena. [43]

⁵⁰ La descarga inicial de las cabeceras de bloque se puede hacer durante el proceso de sincronización inicial.

La Figura 13 ilustra cómo, en la SPV, el camino de Merkle vincula una transacción (Tx3) con un bloque de la cadena, a partir del hash de la raíz de Merkle incluido en la cabecera del bloque.

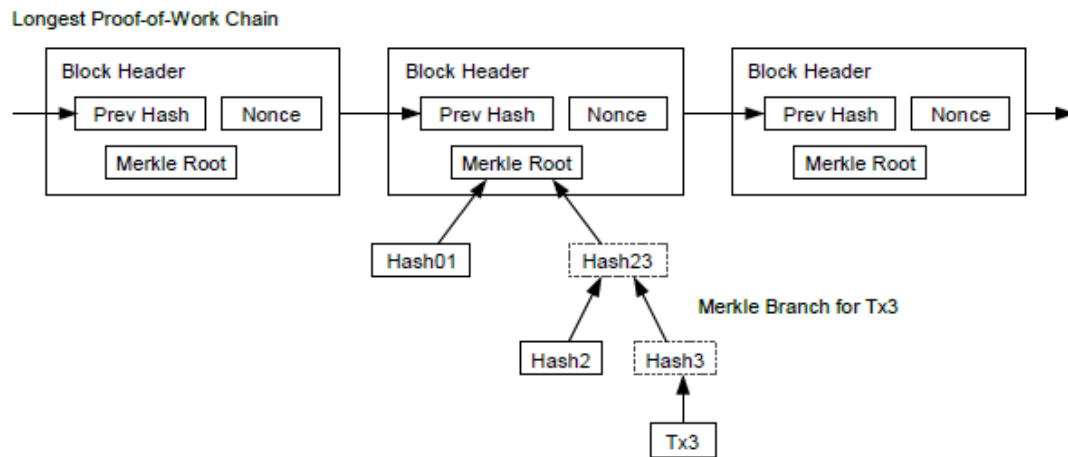


Figura 13: Uso del camino de Merkle para comprobar una transacción en la SPV. Fuente: [43]

Filtros de Bloom

Para solicitar las transacciones que son relevantes para ellos, los nodos livianos utilizan filtros de Bloom⁵¹. Estos filtros son estructuras de datos probabilísticas que permiten verificar si un elemento es miembro de un conjunto. Un nodo liviano construye un filtro de Bloom y lo envía a un nodo completo con el que está conectado para que filtre las transacciones de la red y le mande solo aquellas que coinciden con el filtro. En estos filtros, los falsos positivos son posibles, pero los falsos negativos no, e incluso se puede establecer el nivel deseado de falsos positivos. En Bitcoin, esta característica se usa para prevenir fugas de privacidad: además de pedir las transacciones que corresponden a las direcciones bajo su propiedad, un nodo liviano puede pedir transacciones implicando otras direcciones (a expensas del consumo de ancho de banda) y luego descartarlas. Al hacerlo, se evita que el nodo completo tenga una visión completa de las direcciones asociadas con el nodo liviano. [24], [44]

⁵¹ https://es.wikipedia.org/wiki/Filtro_de_Bloom.

Sin embargo, aunque los filtros de Bloom tienen como objetivo preservar la privacidad, todavía existen problemas a este efecto relacionados con su uso, los cuales se explican en detalles en [24] e incluyen las contramedidas propuestas por Gervais et al. [45].

4.3.2 Validación de Registro

Una vez que un minero crea un bloque, lo envía a sus nodos vecinos para que puedan validarlo de forma independiente y, en tal caso, propagarlo aún más en la red. De lo contrario, si el bloque no supera el proceso de verificación, los nodos lo rechazan y detienen su propagación. Para validar un bloque, los nodos realizan principalmente las siguientes verificaciones⁵² [9], [39]:

- ♦ La estructura de datos del bloque es sintácticamente válida;
- ♦ El hash de la cabecera de bloque es menor que el nivel de dificultad;
- ♦ El tamaño del bloque está dentro de los límites aceptables;
- ♦ La primera transacción (y sólo la primera) es una transacción *coinbase*;
- ♦ Todas las transacciones dentro del bloque son válidas según la lista de verificación de transacciones.

Como se mencionó anteriormente, el proceso de verificación independiente realizado por cada nodo garantiza la seguridad del sistema siempre y cuando haya una mayoría de nodos honestos en la red. La amplia gama de verificaciones indicadas en las reglas de validación detectará y rechazará cualquier intento de agregar una transacción o un bloque fraudulento a la cadena de bloques. Por ejemplo, cualquier intento de un minero para generar una transacción *coinbase* que le recompensa más que el valor de los cargos de transacción no cumplirá con la verificación de bloque necesaria para su agregación a la cadena de bloques principal y, por lo tanto, el bloque será rechazado por los nodos [9].

Una vez validados, los bloques se manejan de tres maneras diferentes dependiendo de la situación. Cada nodo mantiene una cadena de bloques principal,

⁵² Una lista completa de las verificaciones realizadas se puede encontrar en [39].

una cadena de bloques secundaria y un *pool* de bloques huérfanos, que sirven para almacenar los bloques validados. [39]

Cadena de Bloques Principal

En cualquier momento, la cadena de bloques principal es la que lleva la mayor prueba de trabajo. Los bloques válidos cuyo bloque padre es el último de la cadena principal se conectan a esta cadena. [9]

Cadena de Bloques Secundaria

Cuando dos bloques válidos, que tienen el último bloque de la cadena principal como padre, se propagan en la red aproximadamente al mismo tiempo, el primer bloque recibido por los nodos se conectará a su cadena principal y el otro se conectará a una cadena secundaria, que es una rama formada por fuera de la cadena principal⁵³. Cada nodo guarda a dicha cadena en caso de que un bloque válido futuro haga referencia a un padre de esta cadena y la extienda. Debido a que la cadena de bloques principal es por definición la que lleva la mayor prueba de trabajo, una cadena secundaria puede eventualmente extenderse hasta superar la principal y así convertirse en la cadena principal, convirtiendo consecuentemente la cadena principal antigua en una cadena secundaria. [9]

Pool de Bloques Huérfanos

Puede suceder que un nodo reciba un bloque huérfano, que es un bloque válido cuyo padre no forma parte de la cadena principal ni de ninguna cadena secundaria. Esta situación suele producirse cuando dos bloques válidos, donde uno tiene el otro como padre, se generan en un corto período de tiempo y algunos nodos de la red los reciben y validan en el orden incorrecto. En este caso, el bloque válido sin padre se coloca en lo que se llama un *pool* de bloques huérfanos, a la espera de la llegada de su padre, después de qué se pueda vincularlo y ambos pueden conectarse a una cadena (principal o secundaria). [9]

⁵³ Los bloques de la misma altura en distintas cadenas se denominan hermanos.

4.3.3 Finalización de Transacción

Bitcoin es un sistema basado en la PoW, lo que significa que la generación de bloques depende de la potencia computacional gastada por los productores de bloques (es decir, los mineros). Cada nuevo bloque se agrega al fin de la cadena y está vinculado al bloque anterior (es decir, su padre). Tal esquema hace imposible modificar un bloque en la cadena sin modificar todos los bloques posteriores y dedicar los recursos computacionales necesarios para ello, lo que ofrece protección contra ataques de doble gasto. Sin embargo, en teoría, cada bloque y transacción en la cadena de bloques es modificable: el aseguramiento de un usuario para que considere que un bloque no puede ser alterado aumenta a medida que la probabilidad de reconstruir la cadena disminuye cuanto más un bloque se adentra en ella.

Nakamoto demuestra en su *whitepaper* [43] que la probabilidad de construir una cadena alternativa y llegar a la misma altura que la cadena principal disminuye exponencialmente a medida que aumenta el número de bloques que tiene que reconstruir. La probabilidad exacta también depende de la potencia computacional dedicada a la cadena alternativa frente a la dedicada a la cadena principal. En general, Bitcoin recomienda a los usuarios esperar que un bloque reciba al menos seis confirmaciones antes de estar lo suficientemente seguro de que no será modificado. Esto significa esperar hasta que se agreguen cinco bloques por encima del bloque en la cadena⁵⁴. A este punto, el bloque se considera permanentemente confirmado. Sin embargo, seis confirmaciones es sólo un número recomendado que prácticamente elimina la posibilidad de doble gasto.

Dado que el tiempo de generación de bloques se establece a diez minutos en promedio, significa que un usuario tiene que esperar aproximadamente una hora para obtener las seis confirmaciones recomendadas. Como puede no ser un problema para compras importantes (por ejemplo, un coche), donde las transacciones usualmente no tienen que ser confirmadas rápidamente, esto trae una preocupación para las transacciones pequeñas (por ejemplo, un café), donde los

⁵⁴ Se dice que el bloque tiene una profundidad de 5 bloques.

clientes probablemente no están dispuestos a esperar tanto tiempo. A ese efecto, para transacciones de valor pequeño, uno podría considerar una transacción como definitiva después de una sola confirmación, ya que el riesgo de doble gasto sigue siendo bastante bajo. Sin embargo, al hacerlo, la seguridad del sistema se ve afectada y, por lo tanto, tal solución no es posible para mejorar la escalabilidad segura de Bitcoin, la cual es esencial para su adopción social masiva. El problema sigue abierto y se realizan actualmente una gran cantidad de investigaciones para reducir el tiempo de confirmación de las transacciones y mejorar la escalabilidad del sistema.

5 Capa de Datos

La capa de datos se refiere a la información procesada y almacenada en la cadena de bloques, y constituye la finalidad de todas las operaciones del sistema. Estos datos presentan las características buscadas por el sistema: persistencia, transparencia e inmutabilidad. La capa de datos se divide en el componente de operaciones y el componente de libro mayor. [1]

5.1 Componente de Operaciones

Este componente define qué datos se usan como entradas en las operaciones, establece qué tipos de operaciones se pueden ejecutar sobre los datos, y también donde se ejecutan las operaciones. [1]

5.1.1 Entradas

El uso principal de Bitcoin es gestionar las transferencias de activos nativos (es decir, bitcoins) entre los usuarios. Por lo tanto, las transacciones toman principalmente Fondos No Gastados (UTXO) o scripts como entradas, que son datos que se originan en fuentes internas al sistema. Sin embargo, para ciertas transacciones, las entradas también pueden referirse a datos externos arbitrarios para fines de marcas de tiempo resistentes a manipulaciones (*tamper-resistant timestamping* en inglés) [1]. Un buen ejemplo de esto es el primer bloque jamás creado de la cadena de bloques (es decir, el bloque 0 o bloque génesis), que toma como entrada el mensaje " *The Times 03/Jan/2009 Chancellor on brink of second bailout for banks* ", que proviene del título de un artículo del London Times fechado el 3 de enero de 2009 [46].

5.1.2 Transacción Ejecutable Mediante Programación

Una transacción ejecutable mediante programación (*programmatically-executable transaction* en inglés), también denominada contrato inteligente (*smart contract* en inglés), es una transacción automatizada que se ejecuta cuando se

cumplen las condiciones codificadas en un script [1]. En Bitcoin, buenos ejemplos de eso son las transacciones multi-firmas (es decir, *MultiSig*)⁵⁵ y los Contratos Bloqueados con Hash de Tiempo (HTLC)⁵⁶. El uso de scripts para bloquear las salidas permite definir una cantidad infinita de condiciones para que se gasten, sin embargo, el cliente de referencia de Bitcoin solo admite cinco tipos de script de transacción⁵⁷, dependiendo de los requisitos de la transacción [2].

El lenguaje de script Bitcoin se llama *Script* y es un lenguaje de programación basado en pilas (*stack-based* en inglés) similar a *Forth*⁵⁸. Contiene muchos operadores, pero no es un lenguaje Turing completo⁵⁹, por lo que los scripts tienen una complejidad muy limitada y un tiempo de ejecución predecible. La simplicidad y el propósito específico de los scripts califican el sistema Bitcoin como sin estado, pues sólo permite la ejecución de cálculos limitados [1]. Esta es una decisión de desarrollo deliberada y representa una característica de seguridad para el sistema, al impedir la inserción de bucles infinitos o "bombas lógicas"⁶⁰ en los scripts que podrían conducir a ataques de denegación de servicio (DoS). Por lo tanto, como todos los nodos completos de la red validan cada transacción, un lenguaje limitado protege el proceso de validación de las transacciones de ser vulnerable, ya que por lo contrario se podría insertar código malicioso en el script. [9]

5.1.3 Lotes de Ejecución

El sistema Bitcoin está diseñado para ejecutar todas las operaciones internamente (es decir, el lote de ejecución es *on-chain*). Las transacciones se crean, transmiten, validan, e incluyen en bloques que se conectan a la cadena de bloques dentro del sistema. [1]

⁵⁵ Transacciones que requieren más de una firma (clave) para ser autorizadas [74].

⁵⁶ Transacciones que requieren que un beneficiario reconozca recibir el pago antes de una fecha límite, de lo contrario el pago se devuelve al pagador [72].

⁵⁷ A partir de Bitcoin Core 0.9, los tipos de script *pubkey* estándares son: *Pay To Public Key Hash* (P2PKH), *Pay To Script Hash* (P2SH), *MultiSig*, *Pubkey* y *Null data* [68]. Consultar [41] para obtener más detalles sobre los tipos de script estándares.

⁵⁸ <https://es.wikipedia.org/wiki/Forth>

⁵⁹ Una máquina Turing completa puede simular cualquier máquina y resolver cualquier problema de computación [75].

⁶⁰ "Una bomba lógica es una parte de código, insertada en un sistema operativo o una aplicación de software, que implementa una función maliciosa después de un cierto tiempo, o cuando se cumplen condiciones específicas." [73]

El hecho de que cada nodo completo conozca y valide cada transacción en la red es la característica principal que permite a Bitcoin operar de forma segura las transferencias de valores sin la necesidad de un tercero de confianza. Pero, por otro lado, este modelo plantea problemas de escalabilidad en cuanto a su capacidad para abarcar un volumen creciente de transacciones. A ese efecto, se han propuesto varios modelos e implementaciones para ejecutar transacciones en sistemas externos⁶¹ (es decir, *off-chain*), con el fin de reducir el tráfico en la red, minimizar la carga de validación y mejorar el tiempo de procesamiento de las transacciones. Sin embargo, esos modelos presentan limitaciones que aún deben abordarse.

5.2 Componente de Libro Mayor

El componente de libro mayor define a qué hacen referencia los datos incluidos en los bloques de la cadena [1].

5.2.1 Referencia

La cadena de bloques Bitcoin se compone de bloques vinculados criptográficamente y compuestos de múltiples transacciones: las transacciones tienen como objetivo principal transferir los activos nativos (bitcoins) entre los usuarios y la cadena de bloques se esfuerza por realizar un seguimiento a través del tiempo de los flujos de bitcoins. Las variables contenidas en los bloques se refieren a datos generados dentro del sistema y no necesitan ninguna conexión a sistemas externos (es decir, los datos son endógenos) [1]. Estos datos determinan los efectos de las transacciones (gasto y creación de UTXO) o son datos necesarios para validar, pero no determinar, las transacciones y el estado de la cadena de bloques [47].

⁶¹ La idea es iniciar las transacciones en el sistema Bitcoin, pero ejecutar la lógica central de transacción (por ejemplo, los contratos inteligentes) en un sistema externo [1]. Un ejemplo de tal sistema es el *Lightning Network* [65].

6 Seguridad

La seguridad de Bitcoin es un elemento esencial para su éxito y su adopción por los usuarios. Este trabajo no pretende hacer un análisis exhaustivo del asunto, sino abordar los principales problemas que enfrenta Bitcoin. Las secciones 6.1 a 6.6 presentan los principales ataques teóricos al sistema juntos con sus contramedidas. Ciertas contramedidas ya fueron implementadas para prevenir o reducir la probabilidad de ataques, mientras otros tipos de ataques siguen siendo posibles en la práctica. En particular, los *pools* de minería son muy vulnerables si sus puertas de enlaces a la red Bitcoin pueden eclipsarse, y los ataques a los mineros pueden dañar a todo el ecosistema de Bitcoin [48]. La sección 6.7 combina varios ataques entre sí en diferentes escenarios realistas, y finalmente, la sección 6.8 hace brevemente referencia a otras preocupaciones que afectan la seguridad del sistema: los nodos livianos, la privacidad, la maleabilidad de las transacciones y los ataques contra la criptografía de curvas elípticas (ECC).

6.1 Ataque del 51%

Un ataque del 51% ocurre cuando el 51% o más de la potencia computacional total de la red está bajo el control de un minero o grupo de mineros coludidos. Controlar la mayoría del poder computacional permite al atacante mantener la cadena más larga, rechazar bloques de otros mineros, duplicar transacciones y recoger todas las recompensas. Tal situación puede ser devastadora para Bitcoin, haciendo que todo el sistema sea vulnerable y destruyendo los cimientos de su seguridad basada en una arquitectura descentralizada. [49], [50]

Contramedidas

En junio de 2014, un *pool* de minería llamado GHash.IO controló durante un día más del 50% de la potencia computacional de la red, poniendo así en riesgo todo el sistema. En reacción, la comunidad Bitcoin boicoteó al *pool*, que luego se comprometió voluntariamente a mantenerse por debajo del 40% del poder computacional de la red. En este sentido, el comportamiento y las expectativas de la

comunidad hacia los *pools* de minería pueden constituir una protección contra los ataques del 51%. [49], [51], [52]

6.2 Ataque Sybil

El ataque Sybil es un tipo de ataque en el que un nodo controla varios otros nodos en la red, con diferentes direcciones IP, con el objetivo de aumentar su influencia en la red y realizar acciones ilegales en el sistema. Por ejemplo, los ataques Sybil se pueden utilizar para lograr ataques eclipse (sección 6.3), para aumentar el poder computacional del atacante y realizar ataques del 51% o para obtener una ventaja en la propagación de bloques en la red y realizar ataques de doble gasto (sección 6.4) [53]. También puede aumentar las posibilidades del atacante de enviar mensajes *inv* primero en intentos de denegar la entrega de bloques (sección 6.5) [24].

Contramedidas

Bitcoin hace que estos ataques sean más difíciles al permitir solo una conexión saliente con las direcciones IP de cada subred /16⁶². En intentos de realizar un ataque eclipse, esto hace que sea más complejo y más caro para un atacante monopolizar todas las conexiones salientes de la víctima, ya que rara vez es posible pedir varias direcciones IP en varias subredes para el mismo servidor. [54], [55]

6.3 Ataque Eclipse

El ataque eclipse tiene como objetivo aislar (es decir, eclipsar) a la víctima del resto de la red sin su conocimiento con el fin de controlar la información disponible para ella y distorsionar su visión de la cadena de bloques. Se puede hacer forzando todas sus conexiones entre pares para estar con el atacante [48] o denegándole la entrega de transacciones y bloques [34]. Los mineros eclipsantes pueden hacer particiones del poder computacional en la red, lo que puede conducir a ataques de

⁶² Corresponde a una máscara de 255.255.0.0 (clase B) y un rango de 65534 direcciones IP [69].

doble gasto, ataques del 51% y ataques de minería egoísta mejorada (*improved selfish mining* en inglés) [48], [34]. En la sección 6.7, se explica cómo un ataque eclipse puede ser aprovechado para realizar estos ataques.

Contramedidas

Heilman et al. [48] han propuesto una serie de contramedidas relacionadas con la forma en la que las tablas *tried* y *new*⁶³ se rellenan y se explotan en el proceso de creación de conexiones entre pares. Algunas de ellas se han incluido en el cliente Bitcoin v0.10.1 y reducen significativamente la probabilidad de éxito de un ataque eclipse basado en monopolizar las conexiones entre pares de la víctima [56], sin embargo, los ataques de eclipse basados en la denegación de entrega de bloques siguen siendo posibles y difíciles de frustrar. [24], [34]

6.4 Ataque de Doble Gasto

En un ataque de doble gasto, el atacante gasta intencionalmente el mismo UTXO en dos transacciones diferentes con el fin de invalidar una de ellas. Una de las transacciones paga el UTXO a una dirección bajo la propiedad de la víctima, mientras que la otra paga el mismo UTXO, pero a una dirección bajo la propiedad del atacante. El primero pago tiene como objetivo convencer a la víctima para que libere un servicio o un producto al atacante, mientras el objetivo de la segunda transacción es ser incluida en la cadena de bloques principal a expensas de la primera. Por lo tanto, esta última será descartada y el atacante habrá adquirido un servicio o un producto sin pagar.

Los ataques de doble gasto se pueden realizar fácilmente en transacciones de cero confirmación⁶⁴ (0-confirmación) [24] y la sección 6.7 muestra cómo la denegación de entrega de transacciones se puede utilizar a este efecto. La misma sección también explica cómo un adversario puede aprovechar un ataque eclipse

⁶³ Para obtener más detalles sobre las tablas *tried* y *new*, ver [48, p. 4]

⁶⁴ Para transacciones de valores pequeños y para acelerar el tiempo de confirmación, Bitcoin anima a los proveedores a aceptar transacciones tan pronto como las reciban en la red, sin necesidad de que se agreguen a un bloque: estas transacciones se denominan transacciones de 0 confirmación.

para duplicar el gasto y cómo se puede utilizar el doble gasto para mejorar la minería egoísta.

Contramedidas

La mejor protección contra el doble gasto sigue siendo esperar que una transacción se confirme varias veces mediante cada bloque adicional que extiende la cadena. Cuantos más bloques se agreguen y más profunda sea la transacción en la cadena de bloques, menos probable será que una cadena secreta más larga la reemplace y duplique la transacción.

6.5 Denegación de Entrega de Bloques y Transacciones

Este ataque aprovecha una implementación⁶⁵ de Bitcoin para optimizar el ancho de banda de la red y mejorar la escalabilidad, evitando que un nodo reciba nuevas transacciones y bloques durante un cierto período de tiempo. Se puede realizar el ataque enviando a la víctima un mensaje *inv*⁶⁶ que anuncia una nueva transacción o un nuevo bloque y, posteriormente, evitando responder a los mensajes de solicitud de la víctima para esas transacciones o bloques (Figura 14). Los únicos requisitos son mantener una conexión directa con la víctima y ser el primer nodo en enviarle un mensaje *inv*⁶⁷: eso debido a que para minimizar el consumo de ancho de banda, un nodo solo puede solicitar un objeto determinado de un solo par a la vez, normalmente él que envió primero el mensaje *inv*. No responder a un mensaje de solicitud retrasa la entrega de una transacción durante 2 minutos y la entrega de un bloque durante 20 minutos⁶⁸, después de que el nodo realizará la solicitud al siguiente nodo en su lista. Por lo tanto, si un atacante abusa de la medida y envía n

⁶⁵ Sistema de gestión de solicitudes de objetos (*Object request management system*) [24, pp. 61-63].

⁶⁶ Un mensaje *inv* contiene solo el hash y el tipo del objeto (bloque o transacción) y es mucho más pequeño que el bloque o la transacción real. Esto minimiza el ancho de banda al garantizar que solo las transacciones y los bloques que no se han recibido previamente por los nodos sean transferidos [24].

⁶⁷ Un atacante puede aumentar significativamente sus posibilidades de ser el primero en anunciar un objeto al ahorrar tiempo omitiendo su verificación y transmitiéndolo tan pronto como le reciba [34].

⁶⁸ Ver [24, p. 63] para obtener detalles sobre como los tiempos de espera estáticos (*static time-outs*) se determinan.

mensajes *inv*⁶⁹, puede retrasar una transacción durante $2n$ minutos y un bloque durante $20n$ minutos. Esto resulta en un tipo de ataque eclipse donde el atacante aísla a la víctima y controla la información disponible para ella. El ataque se puede aprovechar para realizar dobles gastos, dividir a los mineros o mejorar la minería egoísta. [24], [34]

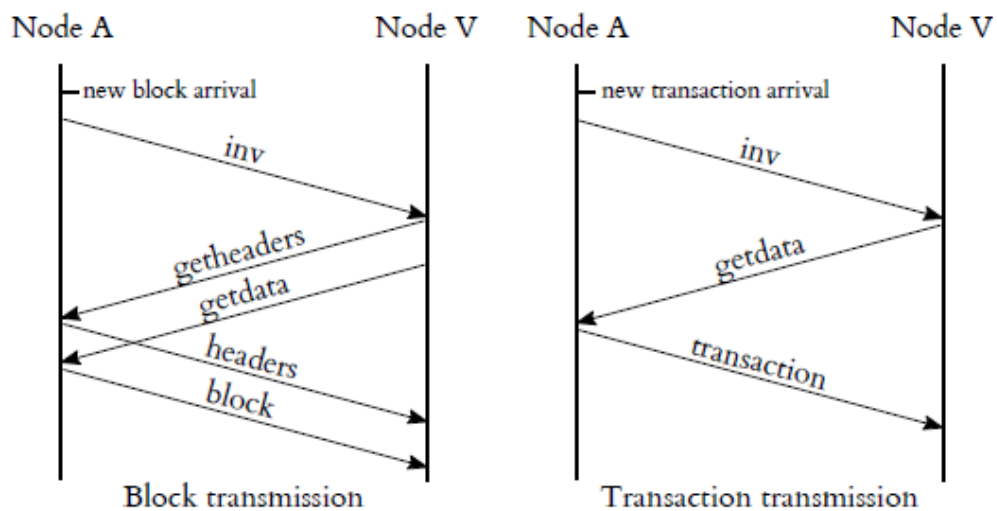


Figura 14: Anuncio de bloques y transacciones en Bitcoin. Fuente: [34]

Denegación de Servicio

Los hallazgos en [34] sugieren ataques de Denegación de Servicio (DoS) se pueden realizar al denegar la entrega de objetos y se estima que 45 nodos completos serían suficientes para denegar constantemente la entrega de bloques en toda la red Bitcoin.

Contrameditas

Poco se puede hacer actualmente para evitar este ataque, ya que un atacante siempre puede controlar varios nodos con distintas direcciones IP para conectarse con la víctima e inundarlo con mensajes *inv*, incluso si Bitcoin hace que sea más complicado al permitir solo una conexión saliente con las direcciones IP de cada

⁶⁹ Un atacante puede denegar continuamente la entrega de objetos enviando un número ilimitado de mensajes *inv*. Para lograr esto, debe llenar todos los enlaces de conexiones abiertos de la víctima con el fin de evitar que reciba otro mensaje *inv* para el mismo objeto de otro par, en cual caso la víctima solicitaría el objeto a este último después del tiempo de espera [34].

subred /16. Esto hace difícil para la víctima saber qué conexiones son confiables e impedir mensajes *inv* maliciosos basados en direcciones IP, y señala las limitaciones del sistema de gestión de solicitudes de objetos, que puede necesitar ser rediseñado. [24]

6.6 Minería Egoísta

Eyal y Sirer [52] demostraron por primera vez que un minero que posee más del 33% de la potencia computacional total de la red siempre puede ganar más que su justa parte de ingresos al retener sus bloques recién acuñados y al construir una cadena de bloques privada. Resulta en una bifurcación en la cadena en la que la cadena privada compite con la pública para ser la cadena de bloques más larga. Al anunciar estratégicamente sus bloques, un minero egoísta puede asegurarse de que los otros nodos abandonen la cadena pública y se reúnan a la más larga. Mientras su cadena privada sea más larga⁷⁰, el minero egoísta se beneficia del hecho de que los otros mineros están desperdiciando su poder computacional en la construcción de una cadena pública más corta. Esto permite al minero egoísta aumentar sus recompensas con respecto al poder computacional desplegado y, como resultado, los mineros racionales preferirán unirse a él. Se crea un efecto dominó en el que la adquisición de más potencia computacional facilita la minería egoísta, que a su vez atrae más potencia computacional. Si un minero egoísta finalmente logra mantener más del 50% del poder computacional de la red, puede realizar ataques de 51% [50] y toda la seguridad de Bitcoin se vería comprometida porque el sistema ya no sería descentralizado. [52]

Retención de Bloques en un Pool

Eyal [49] demostró que un minero egoísta puede aumentar su ventaja infiltrándose en otros *pools* de minería. El *pool* atacante se registra en el *pool* de la víctima y asigna algunos de sus propios mineros a ese *pool* con la tarea de retener los bloques que construyan. Incluso si la potencia computacional del *pool* atacante

⁷⁰ Para conseguir una ventaja en la velocidad de producción de bloques, un atacante puede utilizar una estrategia de minería SPV, que es el acto de minar bloques sin validar el bloque anterior (y las transacciones incluidas en él) en la cadena. Al hacer eso, y para evitar de minar bloques inválidos, los mineros no incluyen ninguna transacción en los bloques porque no las han verificado. [24, p. 123]

se reduce por asignar algunos de sus recursos al *pool* de la víctima, las recompensas para cada miembro del *pool* infiltrado también se reducen: al retener sus bloques, los mineros infiltrados no contribuyen a crear bloques, pero aun así obtienen su parte de la recompensa a medida que gastan recursos informáticos. Tal estrategia podría resultar en una disminución del nivel de dificultad de la PoW y en un aumento en los ingresos del minero egoísta.

Contramedidas

Para contrarrestar la minería egoísta, Eyal y Sirer proponen en [52] que un nodo que es consciente de dos bloques competidores elija aleatoriamente uno para extender la cadena principal y que propague ambos bloques en la red. El objetivo es evitar que el minero egoísta, aprovechándose de una propagación más rápida, traiga más nodos para extender su cadena. Sin embargo, Gervais et al. [34] demostraron que esta medida puede ser contornada fácilmente negando la entrega de bloques, como se explicó anteriormente.

Peor aún, negar la entrega de bloques incluso puede aumentar la ventaja de un minero egoísta (ver sección 6.7). Sapirshtein y al. [50] demostraron que, en realidad, utilizar tal estrategia para contornar tal contramedida reduce el límite del 33% inicialmente establecido por Eyal y Sirer [52] y que los mineros con menos del 25% del poder computacional de la red todavía pueden obtener beneficios de la minería egoísta.

6.7 Combinaciones de Ataques

En esta sección se proporcionan varios ejemplos de cómo se pueden combinar varios ataques para realizar un ataque global más eficiente.

6.7.1 Doble gasto n-confirmaciones mediante un ataque eclipse

Un atacante puede gastar dos veces una transacción eclipsando a su víctima y particionando el poder computacional de la red: el objetivo es que la víctima tenga

una imagen incorrecta de la cadena principal que le hace creer que una cierta transacción fue confirmada. La Figura 15 ilustra un ataque de este tipo. En esta situación, el atacante (en rojo) es un cliente del comerciante (en verde). Primero, el atacante eclipsa a la víctima y a una parte minoritaria del poder computacional de la red, por ejemplo un *pool* de minería (en púrpura), del resto de la red (en negro), forzando todas sus conexiones entre pares con él sin sus conocimientos. Luego envía solo al comerciante y al minero eclipsados una transacción *M* que paga al comerciante. Al estar aislado del resto de la red y al recibir sólo la información que el atacante le permite ver, el minero eclipsado válida la transacción *M*, la incluye en un bloque, agrega el bloque a su cadena, y conecta otros bloques por encima del mismo a medida que el atacante le envía otras transacciones de la red. Esta versión de la cadena de bloques se muestra al comerciante y es la única que él ve, por lo que cree que la transacción *M* ha sido incluida en la cadena de bloques principal y confirmada *n* veces por los bloques posteriores. [24], [48], [56]

Por otro lado, en lugar de transmitir la transacción *M* al resto de la red, el atacante crea y emite una transacción *A* que le paga a él mismo los mismos fondos (es decir, los mismos UTXO) que envió al comerciante en la transacción *M*. Como el resto de la red no es consciente de la transacción *M* ni de ninguna otra cadena, crea una cadena que incluye solo la transacción *A* y no la transacción *M*. [24], [48], [56]

Debido a que el minero eclipsado posee una minoría del poder computacional de la red, su cadena eventualmente será obsoleta ya que el resto de la red ha construido una cadena de bloques más larga que lleva más prueba de trabajo. La transacción *M* se agregará de nuevo al *mempool*, pero fallará el proceso de validación y nunca se incluirá en un bloque porque gasta los mismos UTXO que la transacción *A* ya confirmada. Al final, el atacante habrá gastado con éxito en doble la transacción, sin pagar al comerciante y, en lugar, enviándose a sí mismo los fondos. [24], [48], [56]

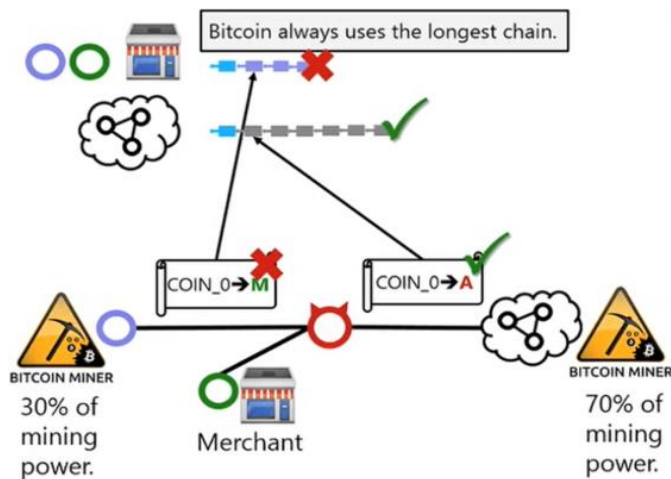


Figura 15: Esquema de un doble gasto n -confirmaciones mediante un ataque eclipse. Fuente: [56]

6.7.2 Ataque del 51% mediante un ataque eclipse

El ataque eclipse puede permitir realizar un ataque similar al del 51% con menos del 50% del poder computacional total, pero todavía requiere un fuerte poder del atacante. La idea es dividir estratégicamente la red para que el atacante compita con cada minero particionado y que ellos no puedan construir sobre los bloques de los demás. Como resultado, los mineros particionados trabajan cada uno en la extensión de su propia cadena que eventualmente abandonarán para aceptar la más larga del atacante como la cadena de bloques principal. Esto pone al atacante en una situación similar como en un ataque del 51%, ya que puede sostener continuamente la cadena más larga y controlar la red. La Figura 16 muestra cómo se puede lograr con solamente el 40% de la potencia computacional total de la red. [56]

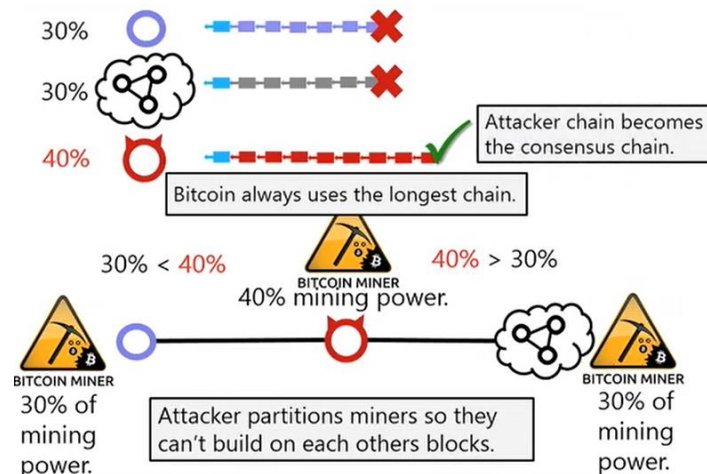


Figura 16: Esquema de un ataque del 51% mediante un ataque eclipse. Fuente: [56]

6.7.3 Doble gasto 0-confirmación mediante la denegación de entrega

El atacante crea una transacción T_l que paga al vendedor y una transacción T_d que le paga a él mismo, ambas gastando los mismos UTXO. El atacante envía a la víctima un mensaje *inv* para T_d y no responde a las solicitudes de la víctima para obtener la transacción: esto impide que el vendedor reciba T_d de la red. A continuación, el atacante propaga T_d en la red y, poco después, envía T_l al vendedor, que lo reenvía en la red. Dado que el vendedor solo es consciente de T_l , libera el producto o el servicio al atacante cuando recibe la transacción, sin esperar ninguna confirmación. Ambas T_d y T_l se propagan en la red, pero es más probable que T_d se incluya en la cadena de bloques principal porque la mayoría de los nodos la recibieron primero. Consecuentemente, T_l será inválida. [34]

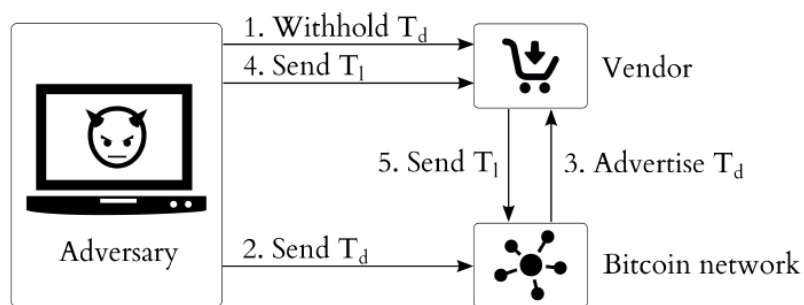


Figura 17: Esquema de un doble gasto 0-confirmación mediante la denegación de entrega de transacción. Fuente: [34]

6.7.4 Doble gasto 1-confirmación mediante la denegación de entrega

La idea es la misma que para el doble gasto 0-confirmación, pero el atacante también envía un mensaje *inv* para T_d a un minero que posee una minoría de poder computacional, y poco después, le envía T_l . Como los otros mineros reciben T_d primero, la incluyen en un bloque B_d a sus cadenas. Tan pronto como el atacante recibe B_d , no lo verifica e inmediatamente envía un mensaje *inv* para B_d tanto al vendedor y al minero débil para evitar que lo reciban de la red. Por lo tanto, el minero débil sólo conoce T_l y la incluye en un bloque B_l que agrega a su cadena. Más adelante, el vendedor ve que T_l se confirmó en un bloque B_l y libera el producto o el servicio al atacante. Después de los tiempos de espera para una transacción y

un bloque, el vendedor y el minero débil recibirán T_d y B_d pero será demasiado tarde y B_l será descartado al beneficio de una cadena más larga que incluye B_d . [34]

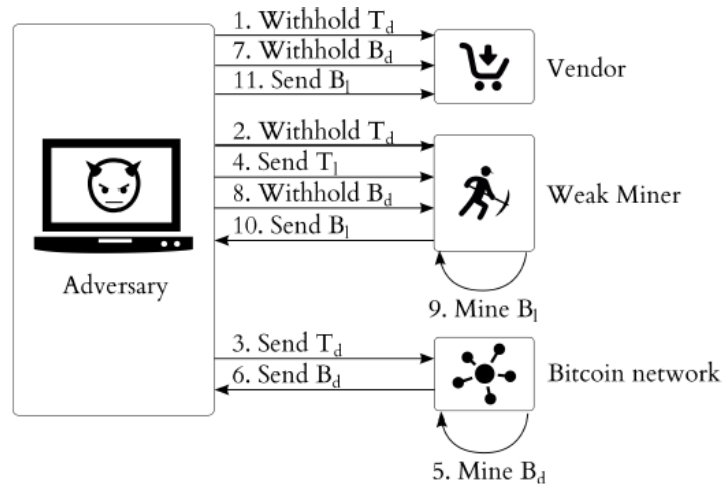


Figura 18: Esquema de doble gasto 1-confirmación mediante una denegación de entrega de transacción. Fuente: [34]

6.7.5 Minería egoísta y doble gasto

En el artículo *Optimal Selfish Mining Strategies in Bitcoin* [50], los autores demuestran que un minero egoísta rentable no pierde nada al intentar doble gastar al mismo tiempo. La idea para el minero egoísta es estar implicado regularmente en las transacciones e incluir transacciones de doble gasto a su cadena secreta. En el caso de que su cadena secreta se convierta en la más larga y reemplace a la cadena de bloques pública principal, es posible que los destinatarios de los fondos ya hayan aceptado los pagos y, por lo tanto, que se haya logrado el doble gasto. En el peor caso, en el que la cadena secreta sea rechazada y que el doble gasto falle, el atacante seguirá beneficiándose equitativamente de las transacciones justas.

6.7.6 Minería egoísta y denegación de entrega de bloque

Sapirshtein y al. [50] demostraron que un minero que puede contar con el retraso de la entrega de bloques (es decir, que siempre puede enviar sus bloques primero ($\gamma = 1$)), siempre puede beneficiarse de una estrategia de minería egoísta, independientemente de su tamaño. Los resultados de Gervais y al. [34] también reducen el límite del 33% indicado por Eyal y Sirer [52]: un atacante con el 26,5% del

poder computacional que niega la entrega de 2 bloques consecutivos al 50% de la red puede beneficiarse de la minería egoísta, e incluso puede controlar la red con el 35% de los recursos computacionales, como se muestra en la Figura 19.

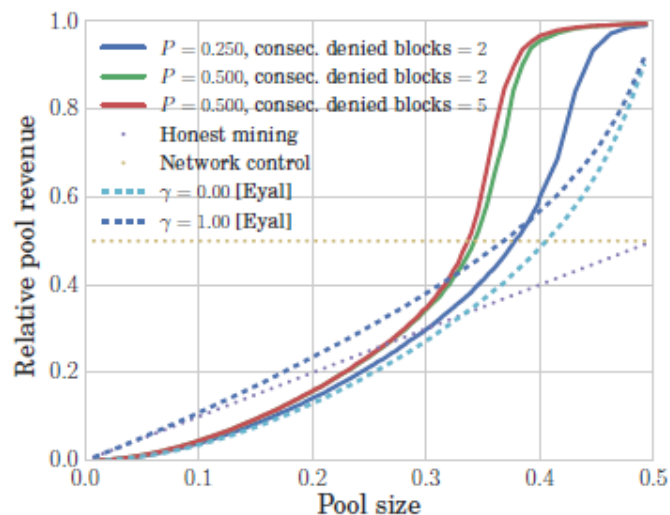


Figura 19: Ventaja de un minero egoísta denegando la entrega de bloques. Fuente: [34]

6.8 Preocupaciones Adicionales

6.8.1 Nodos Livianos

Estos nodos no almacenan toda la cadena de bloques ni validan todas las transacciones del sistema. Dependen de los nodos completos con los que están conectados⁷¹ para recibir las transacciones relevantes para ellos, usando filtros de Bloom, y solo realizan una cantidad limitada de verificaciones sobre estas, como el cumplimiento de la PoW y la presencia de la transacción en el árbol de Merkle. Este proceso se hace a expensas de la descentralización, ya que un componente crítico de la seguridad del sistema depende de pocos nodos completos. Si un nodo liviano está conectado al menos con un nodo completo que siga correctamente el protocolo y tiene una versión actualizada de la cadena, la seguridad no se ve realmente comprometida. Sin embargo, si todas sus conexiones están con nodos maliciosos, el nodo liviano se encuentra eclipsado de la red y puede ser controlado por ellos de la misma manera que en los ataques mencionados anteriormente. [24]

⁷¹ Un nodo liviano se conecta por defecto con cuatro nodos completos [24].

6.8.2 Privacidad

Además de algunos problemas de privacidad asociados con los nodos livianos y el uso de filtros de Bloom (ver sección 4.3.1), existe una preocupación general sobre la privacidad en Bitcoin: el hecho de que todas las transacciones se publican con transparencia en la cadena de bloques permite el seguimiento de las actividades de los usuarios. Es cierto que los usuarios que participan en transacciones usan seudónimos (es decir, direcciones), sin embargo, como el libro mayor consiste en una cadena de transacciones vinculadas con firmas digitales, los flujos de bitcoins se pueden rastrear públicamente. Desde una perspectiva de seguridad, al analizar las conexiones y el tráfico entre pares en la red, un atacante puede conseguir informaciones sobre direcciones y transacciones, y potencialmente vincular una transacción con la dirección IP del remitente. Para superar ese problema, existen empresas que pueden realizar transacciones en nombre de los usuarios ofuscando el origen del pago⁷², pero si bien la solución previene la trazabilidad, la participación de un tercero lleva a la necesidad de confiar en él, lo que Bitcoin busca evitar. En [24], Karame y Androulaki describen y analizan una serie de propuestas para habilitar la preservación de privacidad en Bitcoin, como CoinJoin⁷³ (protocolo de mezcla de transacciones) y las firmas Schnorr⁷⁴ (mejora criptográfica al protocolo), que eliminan la necesidad de un tercero, pero a expensas de la performance del sistema. [24]

6.8.3 Maleabilidad de transacciones

El script *signature* firma los datos de la transacción para asegurar su integridad, pero no protege el script en sí mismo, dejando la posibilidad de modificarlo y así cambiar el hash de la transacción (*txid*). Si bien las modificaciones posibles gracias a dicha maleabilidad no tienen impacto en los efectos de la transacción y no la invalidan, un atacante podría ocultar la existencia de la transacción modificando su *txid*, por ejemplo, agregando datos al script *signature* que se eliminarán antes de que se procese el script *pubkey*. Dicha estrategia se

⁷² Se dice de esas empresas que ofrecen un servicio de *mixing* [24].

⁷³ <https://en.bitcoin.it/wiki/CoinJoin>

⁷⁴ https://en.wikipedia.org/wiki/Schnorr_signature

puede utilizar para convencer al emisor de que vuelva a emitir los fondos. Esto no constituye un problema para las transacciones que se agregan a la cadena de bloques de inmediato, pero lo es cuando la salida de la transacción (UTXO) se gasta antes de que la misma se agregue a la cadena, ya que sigue siendo maleable [57]. Para superar el problema, la implementación del *BIP141: Segregated Witness* (SegWit) en agosto 2017 añade un identificador más a las transacciones (*wtxid*) que permite validar la existencia de una transacción sin tener en cuenta los datos de validación como el script *signature* [47].

Cuando no se usa SegWit, las nuevas transacciones no deberían depender de transacciones anteriores que aún no se han agregado a la cadena de bloques. Además, se recomienda hacer el seguimiento de las transacciones a partir de los UTXO (*output index*) en lugar del *txid*, ya que estos no se pueden modificar sin invalidar la transacción. Finalmente, si una transacción parece desaparecer de la red y necesita ser reemitida, se debería hacerlo de una manera que invalide la transacción perdida, por ejemplo, pagando de nuevo los mismos UTXO que gastó la transacción perdida. [57]

6.8.4 Ataques contra la ECC

Existen ataques [58] contra la mayoría de las implementaciones de criptografía de curvas elípticas (ECC) en las que un atacante, en una máquina virtual compartida con la víctima, puede lograr extraer una clave privada haciendo que la víctima firme con la misma clave cientos de veces. Por lo tanto, se recomienda evitar el uso de hosts compartidos y la reutilización de claves, y generar nuevos pares de claves para cada transacción. [59]

7 El Futuro de Bitcoin

Este capítulo presenta los aspectos principales que van a definir la evolución de Bitcoin, o sea el establecimiento de regulaciones para encuadrar el mercado de las criptomonedas, las soluciones alternativas de criptomonedas que se ofrecen a los usuarios, así como la capacidad de Bitcoin para mejorar su sistema.

7.1 Marco Legal de las Criptomonedas

Necesariamente, del uso de una nueva tecnología como las cadenas de bloques surgen problemas legales. En el caso de las criptomonedas, estos apuntan principalmente hacia tres aspectos [60]:

- ¿Cómo proteger a los consumidores y los inversores que no entienden necesariamente estos instrumentos financieros y sus riesgos?
- ¿Cómo los gobiernos pueden detener el uso de las criptomonedas para financiar actividades ilegales?
- ¿Cómo estos activos deberían ser gravados?

A pesar de las expectativas para la reunión de los ministros de finanzas y gobernadores de bancos centrales de los países del G20 en Buenos Aires en marzo de 2018, no se emitió un marco regulatorio o pautas específicas sobre las criptomonedas [61], si bien hoy no hay consenso internacional acerca de su regulación y cada jurisdicción tiene la libertad de establecer su propia regulación. Ciertas jurisdicciones permiten al mercado de las criptomonedas de operar mientras otras, como Irán y China, restringen el mercado principalmente al prohibir a las instituciones dentro de sus fronteras operar en dicho mercado. En las jurisdicciones en las que las operaciones sobre criptomonedas son permitidas, algunos países (como Bielorrusia y México) han desarrollado leyes propias que reconocen y regulan el mercado, mientras otros (como Argentina, Brasil y Francia) reconocen y permiten la existencia del mercado, pero todavía no han emitido leyes específicas a la industria. En cualquier caso, cuando las transacciones de criptomonedas son permitidas, muchos países imponen impuestos. Sin embargo, el tratamiento fiscal de

los ingresos generados por una transacción de criptomonedas puede variar dependiendo de cómo está categorizada. Y ahí está el desafío: la regulación con respecto a la emisión y a los impuestos de criptomonedas, así como el tratamiento de los activos digitales, plantean preguntas importantes sobre cómo estos activos digitales deben ser clasificados. [61]

De forma general, el establecimiento de una regulación sobre las criptomonedas cumple ciertas necesidades de control sobre el mercado para proteger los consumidores e inversores, y también para prevenir actividades fraudulentas y evasión fiscal. Por otro lado, una regulación conlleva ciertas restricciones en cuanto al acceso al mercado de las criptomonedas. Si las normas legales se vuelven demasiado restrictivas, tendrán el efecto de limitar el potencial de crecimiento de las criptomonedas.

7.2 Criptomonedas Alternativas

Desde Bitcoin, cientos de criptomonedas nuevas han inundado el mercado [62]. Incluso Facebook anunció recientemente su propio proyecto de criptomoneda, Libra⁷⁵, que ya se enfrenta a problemas regulatorios y críticas [63]. Más aun, el Banco Popular de China (PBOC) está a solo unos meses de lanzar la versión digital del Yuan chino, lo que convierte a China en el primer país del mundo en tener una moneda digital del banco central. Además de China, Irán y Rusia también parecen interesados en utilizar la moneda digital como base para un sistema financiero paralelo que los Estados Unidos no pueden controlar [64].

Estos proyectos de criptomonedas alternativas a Bitcoin confirman el interés generalizado de varias empresas y gobiernos para convertirse a las criptomonedas, a menudo con el objetivo de aumentar su control en el mercado financiero y la economía, a expensas de la cuota de mercado de Bitcoin. Últimamente, es la capacidad de Bitcoin para competir contra esas alternativas y atraer usuarios lo que definirá su nivel de éxito. Actualmente, además de su protocolo que ya hizo sus pruebas en los últimos años, Bitcoin tiene la ventaja en comparación con otras

⁷⁵ [https://fr.wikipedia.org/wiki/Libra_\(cryptomonnaie\)](https://fr.wikipedia.org/wiki/Libra_(cryptomonnaie))

criptomonedas similares de tener el mayor número de usuarios, lo que fortalece su seguridad al hacer más improbable la ocurrencia de un ataque a la red del tipo 51%, ya que requiere mucho más esfuerzos y costos para lograr controlar la mayoría del poder computacional de la red.

7.3 Mejora de las debilidades de Bitcoin

Las principales ventajas de Bitcoin para atraer usuarios, en comparación a los sistemas de monedas digitales propuestos por Facebook y China, son sus características de descentralización, privacidad y anonimato, que hacen difícil rastrear las transacciones y, consecuentemente, no permiten ejercer ningún tipo de control sobre ellas. Es importante que Bitcoin siga proporcionando estas propiedades a sus usuarios, asegurándose de la descentralización del sistema tanto al nivel de la gobernanza como al del protocolo, y mejorando ciertos aspectos que ahora permiten la trazabilidad de las transacciones y comprometen la privacidad, tal como mencionamos anteriormente. Con respecto al anonimato, es probable que el asunto se resuelva mediante regulaciones que Bitcoin deberá cumplir para ser aceptado por los gobiernos e instituciones.

También, Bitcoin debe mejorar su escalabilidad, que actualmente es un obstáculo importante para su adopción masiva. Como vimos en este trabajo, a través del mecanismo de la PoW, Bitcoin controla la velocidad de generación de bloques para crear uno nuevo cada diez minutos en promedio. Combinado con la imposición de un límite de tamaño de bloque de 1 MB, esto limita la cantidad de transacciones que pueden llevarse a cabo durante un cierto período de tiempo. El tema es actualmente un campo de estudio muy activo y se han propuesto varias soluciones, como aumentar el tamaño de bloque [26] o ejecutar las transacciones en un sistema externo (*Lightning Network* [65]), pero aun así, ninguna implementación resuelve el problema hasta el punto en que Bitcoin puede manejar la cantidad de transacciones necesaria para soportar una adopción masiva [66]. A modo de comparación, mientras Visa realiza alrededor de 1700 transacciones por segundo en promedio (según un cálculo derivado del reclamo oficial de más de 150 millones de transacciones por día), Bitcoin actualmente procesa 4,6 transacciones por segundo [67].

Conclusión

Bitcoin es un sistema de criptomoneda basado en la tecnología de cadena de bloques, la cual es una aplicación de la tecnología más amplia de registro distribuido (DLT⁷⁶). El objetivo de este trabajo fue analizar el funcionamiento de Bitcoin basándose en el marco propuesto en el documento *Distributed Ledger Technology Systems: A conceptual framework* [1] de la Universidad de Cambridge, que establece un marco y una terminología específica para los sistemas DLT. Además, se hizo también un análisis de la seguridad de Bitcoin en la segunda parte.

En primer lugar, en la sección 2, vimos como las primitivas criptográficas que soportan Bitcoin logran proporcionar seguridad al sistema. La función hash SHA-256 se usa a lo largo del procesamiento de las transacciones (direcciones, árboles de Merkle, hashes de transacción, hashes de bloque, PoW) principalmente para asegurar la integridad y la inmutabilidad de la información. Los árboles de Merkle son estructuras de datos jerárquicas formadas por hashes de transacciones que sirven para verificar la integridad de las mismas, además de permitir a los nodos livianos confirmar su inclusión en un bloque. En cuanto a las curvas elípticas, se utilizan para generar pares de claves, donde las claves públicas sirven como direcciones para enviar los fondos y las claves privadas sirven para firmar las transacciones y garantizar la legitimidad del destinatario.

En las secciones 3 a 5, diseccionamos el sistema de Bitcoin según el marco para los DLT, que se divide en tres capas: protocolo, red y datos. La capa de protocolo es la que define, administra y actualiza el conjunto de reglas que rigen el sistema, y codifica el diseño de su arquitectura. La capa de red es la implementación práctica de las reglas del protocolo que establece como se hace la comunicación entre los participantes de la red, como se comparten y procesan los datos, y como estos últimos se validan para ser parte del libro mayor. Finalmente, la capa de datos trata de la información procesada y almacenada en la cadena de bloques, y constituye la finalidad de todas las operaciones del sistema. La tabla siguiente presenta un resumen de las características de Bitcoin presentadas según el marco de referencia aplicado.

⁷⁶ *Distributed Ledger Systems* (https://en.wikipedia.org/wiki/Distributed_ledger)

Componente	Proceso	Bitcoin
------------	---------	---------

Capa de Protocolo	Génesis	Dependencias entre Sistemas	Sistema autosuficiente
		Creación de Código Base	Desde cero; código abierto
		Iniciación de Reglas	Reglas establecidas por el cliente de referencia (Bitcoin Core)
	Alteración	Gobernanza del Protocolo	Anárquica/Jerárquica: usuario vota corriendo el cliente de su elección, pero Bitcoin Core tiene una influencia significativa
		Cambios en el Protocolo	Bitcoin Improvement Proposal (BIP): usuario vota corriendo cliente de su elección

Capa de Red	Comunicación	Acceso a la Red	Abierta y sin restricción
		Difusión de Datos	Universal (pública)
		Iniciación de Transacción	Sin restricción: cualquiera con una clave privada puede crear y firmar una transacción y transmitirla a la red
	Procesamiento de Transacciones	Propuesta de Registro	Sin permiso: mineros agrupan transacciones no confirmadas de su mempool en un bloque. El bloque requiere un hash SHA-256 que cumple la prueba trabajo (PoW) para ser válido
		Reglas de Resolución de Conflictos	Cadena de bloque válida más larga (la que lleva la mayor PoW)
		Incentivos para el Procesamiento de Transacciones	Intrínseco y monetario : recompensa por bloque (nuevos BTC acuñados y cargos de transacción)
	Validación	Validación de Transacción	Nodos completos validan cada transacción antes de pasarlas a otros nodos
		Validación de Registro	Nodos completos validan cada bloque antes de pasarlos a otros nodos
		Finalización de Transacción	Nunca en teoría; probabilidad de alteración disminuye con la profundidad del bloque en la cadena; Bitcoin considera permanente confirmada una transacción con una profundidad de 5 bloques (es decir, 6 confirmaciones)

Capa de Datos	Operaciones	Entradas	Principalmente internas (UTXO, scripts); external (datos arbitrarios con propósito de marco de tiempo)
		Transacción Ejecutable Mediante Programación	Sin estado: lenguaje de script limitado permitiendo múltiples firmas y contratos <i>timelocked</i>
		Lotus de Ejecución	Operaciones <i>on-chain</i> sencillas
	Libro Mayor	Referencia	Endógeno: activo nativo exclusivo del sistema (BTC)

Tabla 8: Propiedades de Bitcoin según un marco para sistemas DLT. Fuente: [1]

En la sección 6, vimos los principales ataques teóricos contra Bitcoin: ataque del 51%, ataque Sybil, ataque eclipse, ataque de doble gasto, denegación de entrega de bloques y de transacciones, así como la minería egoísta. También

mencionamos ciertas contramedidas posibles y brindamos ejemplos de combinaciones de ataques realizables en la práctica. En particular, vale la pena resaltar que actualmente existen pocas medidas para evitar la denegación de entrega de bloques y transacciones, lo que permite llevar a cabo ataques de denegación de servicios (DoS), doble gasto y aumentar la eficiencia de la minería egoísta.

En la sección 7, expusimos elementos que se deben tener en cuenta en la valoración sobre el futuro de Bitcoin. La regulación del mercado de las criptomonedas, así como la capacidad de Bitcoin para mejorar sus debilidades y competir con criptomonedas alternativas son aspectos que influirán en la adopción y el éxito de Bitcoin.

A la luz de este análisis, se concluye que Bitcoin es un sistema eficiente y robusto, basado en una tecnología (cadena de bloques) que, incluso si es nueva, ya demostró su gran potencial. Aunque existen ciertas vulnerabilidades en el sistema, no constituyen un riesgo importante y su seguridad general está garantizada por primitivas criptográficas confiables. Si Bitcoin logra superar las limitaciones en relación con su escalabilidad y si logra mantener la descentralización del sistema, es razonable pensar que podrá soportar una adopción masiva exitosamente.

Referencias Específicas

- [1] Rauch M. et al., «Distributed Ledger Technology Systems: A conceptual framework,» Cambridge Center for Alternative Finance, 2018.
- [2] I. Bashir, Mastering Blockchain, Birmingham: Packt Publishing Ltd., 2017.
- [3] L. Lamport, R. Shostak y M. Pease, «The Byzantine Generals Problem,» *ACM Transactions on Programming Languages and Systems*, pp. 382-401, 1982.
- [4] «CB Insights - What is Blockchain Technology,» 11 Septiembre 2018. [En línea]. Available: <https://www.cbinsights.com/research/what-is-blockchain-technology/>. [Último acceso: 20 Diciembre 2019].
- [5] «Wikipedia - Cryptographic Hash Function,» [En línea]. Available: https://en.wikipedia.org/wiki/Cryptographic_hash_function. [Último acceso: 15 September 2019].
- [6] «Bitcointalk.org,» 24 Septiembre 2011. [En línea]. Available: <https://bitcointalk.org/index.php?topic=45456.0>. [Último acceso: 23 Enero 2020].
- [7] «Wikipedia - Merkle-Damgård Construction,» [En línea]. Available: https://en.wikipedia.org/wiki/Merkle%E2%80%93Damgård_construction. [Último acceso: 15 September 2019].
- [8] «Steemit - Introduction to The SHA-256 Hash Function,» 9 Enero 2017. [En línea]. Available: <https://steemit.com/cryptocurrency/@f4tca7/introduction-to-the-sha-256-hash-function>. [Último acceso: 15 September 2019].
- [9] A. M. Antonopoulos, Mastering Bitcoin, Sebastopol: O'Reilly Media, 2015.
- [10] «bitcoin.it - Transaction,» [En línea]. Available: <https://en.bitcoin.it/wiki/Transaction>. [Último acceso: 16 September 2019].
- [11] C. Barski y C. Wilmer, Bitcoin for the Befuddled, San Francisco: No Starch Press, Inc., 2015.
- [12] «Bitcoin.stackexchange.com,» 29 Julio 2018. [En línea]. Available: <https://bitcoin.stackexchange.com/questions/8443/where-is-double-hashing-performed-in-bitcoin/8461#8461>. [Último acceso: 23 Enero 2020].
- [13] C. Wright, «Medium.com - The puzzle of the double hash,» 30 Abril 2019. [En

- línea]. Available: https://medium.com/@craig_10243/the-puzzle-of-the-double-hash-968196edb06d. [Último acceso: 26 Enero 2020].
- [14] J. M. Miret, J. Valera y M. Valls, «Criptored - Criptografía con Curvas Elípticas,» 12 Mayo 2015. [En línea]. Available: <http://www.criptored.upm.es/crypt4you/temas/ECC/leccion1/leccion1.html#02>. [Último acceso: 19 September 2019].
- [15] H. Knutson, «Hackernoon - What is the math behind elliptic curve cryptography,» 5 Abril 2018. [En línea]. Available: <https://hackernoon.com/what-is-the-math-behind-elliptic-curve-cryptography-f61b25253da3>. [Último acceso: 18 September 2019].
- [16] S. f. E. C. Group, «Standards for Efficient Cryptography 2 (SEC2),» Certicom Corp., 2010.
- [17] E. J. Cousiño Godoy, *Criptomonedas: Estudio de las redes de Bitcoin y Ethereum*, Buenos Aires: Universidad de Buenos Aires, 2018.
- [18] «Medium - How Schnorr signatures may improve Bitcoin,» 13 Junio 2018. [En línea]. Available: <https://medium.com/cryptoadvance/how-schnorr-signatures-may-improve-bitcoin-91655bcb4744>. [Último acceso: 31 Enero 2020].
- [19] «SSL.com - Comparing ECDSA vs RSA,» [En línea]. Available: <https://www.ssl.com/article/comparing-ecdsa-vs-rsa/>. [Último acceso: 19 September 2019].
- [20] "Elliptic Curve Digital Signature Algorithm," [Online]. Available: https://fr.wikipedia.org/wiki/Elliptic_curve_digital_signature_algorithm. [Accessed 17 September 2019].
- [21] A. Back, «Hashcash - a denial of service counter-measure,» 2002.
- [22] W. Dai, "b-money," 1998.
- [23] «Wikipedia - Bitcoin Core,» [En línea]. Available: https://en.wikipedia.org/wiki/Bitcoin_Core. [Último acceso: 25 August 2019].
- [24] G. Karame y E. Androulaki, *Bitcoin and Blockchain Security*, London: Artech House, 2016.
- [25] «Github - Contributing to Bitcoin Core,» [En línea]. Available: <https://github.com/bitcoin/bitcoin/blob/master/CONTRIBUTING.md>. [Último

acceso: 26 August 2019].

- [26] M. Hearn, «Medium.com - Mike's Blog - The resolution of the Bitcoin experiment,» 14 Enero 2016. [En línea]. Available: <https://blog.plan99.net/the-resolution-of-the-bitcoin-experiment-dabb30201f7#.9g9iryds0>. [Último acceso: 26 August 2019].
- [27] «Bitcoinwiki.org - Bitcoin Improvement Proposals,» [En línea]. Available: https://en.bitcoinwiki.org/wiki/Bitcoin_Improvement_Proposals. [Último acceso: 26 August 2019].
- [28] «bitcoin.stackexchange.com - How do people who work on Bitcoin Core get paid,» [En línea]. Available: <https://bitcoin.stackexchange.com/questions/62141/how-do-the-people-who-work-on-bitcoin-core-get-paid>. [Último acceso: 27 August 2019].
- [29] «Github - Bitcoin XT,» [En línea]. Available: <https://github.com/bitcoinxt/bitcoinxt>. [Último acceso: 25 August 2019].
- [30] «Wikipedia - Bitcoin Cash,» [En línea]. Available: https://fr.wikipedia.org/wiki/Bitcoin_Cash. [Último acceso: 26 August 2019].
- [31] «Bitcoin.org - Consensus Rules Changes,» [En línea]. Available: <https://bitcoin.org/en/blockchain-guide#consensus-rule-changes>. [Último acceso: 20 Enero 2020].
- [32] D. Yaga, P. Mell, N. Roby y K. Scarfone, «Blockchain Technology Overview,» 2018.
- [33] «Github,» [En línea]. Available: <https://github.com/bitcoin/bitcoin/blob/master/doc/reduce-traffic.md>. [Último acceso: 28 August 2019].
- [34] A. Gervais, H. Riztdorf, G. O. Karame y S. Capkun, «Tampering with the Delivery of Blocks and Transactions in Bitcoin,» Zurich, 2015.
- [35] J. Valenzuela, «Dash News - Can Bitcoin Survive “China Mining Pool Attack” Without ChainLocks?,» 2 Septiembre 2019. [En línea]. Available: <https://dashnews.org/can-bitcoin-survive-china-mining-pool-attack-without-chainlocks/>. [Último acceso: 31 Enero 2020].
- [36] «Alkazeera - China's bitcoin miners have greater production power: research,» 11 Diciembre 2019. [En línea]. Available:

- <https://www.aljazeera.com/ajimpact/china-bitcoin-miners-greater-production-power-research-191211190946896.html>. [Último acceso: 31 Enero 2020].
- [37] «Wikipedia - History of Bitcoin,» [En línea]. Available: https://en.wikipedia.org/wiki/History_of_bitcoin. [Último acceso: 8 September 2019].
- [38] «Coingecko - Bitcoin Halving,» [En línea]. Available: https://www.coingecko.com/fr/explain/bitcoin_halving. [Último acceso: 8 September 2019].
- [39] «bitcoin.it - Protocol Rules,» [En línea]. Available: https://en.bitcoin.it/wiki/Protocol_rules. [Último acceso: 10 September 2019].
- [40] «Bitcoin.org - Non Standard Transactions,» [En línea]. Available: <https://bitcoin.org/en/transactions-guide#non-standard-transactions>. [Último acceso: 12 September 2019].
- [41] «Bitcoin.org - Standard transactions,» [En línea]. Available: <https://bitcoin.org/en/transactions-guide#standard-transactions>. [Último acceso: 1 September 2019].
- [42] «Bitcoin.org - Transaction guide - Introduction,» [En línea]. Available: <https://bitcoin.org/en/transactions-guide#introduction>. [Último acceso: 3 Septiembre 2019].
- [43] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System,» 2008.
- [44] «Bitcoin.org - Simplified Payment Verification,» [En línea]. Available: <https://bitcoin.org/en/operating-modes-guide#simplified-payment-verification-spv>. [Último acceso: 24 Enero 2020].
- [45] A. Gervais, G. O. Karame, D. Gruber y S. Capkun, «On the Privacy Provisions of Bloom Filters in Lightweight Bitcoin Clients,» Proceedings of the 30th Annual Computer, New Orleans, 2014.
- [46] B. Sherry, «Investopedia - What is the Genesis Block in Bitcoin Terms,» 25 Junio 2019. [En línea]. Available: <https://www.investopedia.com/news/what-genesis-block-bitcoin-terms/>. [Último acceso: 15 September 2019].
- [47] «Github - BIP 141,» 10 Marzo 2018. [En línea]. Available: <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>. [Último acceso: 22 Diciembre 2019].

- [48] E. Heilman, A. Kendler, A. Zohar y S. Goldberg, «Eclipse Attack on Bitcoin's Peer-to-Peer Network,» 24th USENIX Security Symposium, Washington, 2015.
- [49] I. Eyal, «The Miner's Dilemma,» Cornell University, 2014.
- [50] A. Sapirshtein, Y. Sompolinski y A. Zohar, «Optimal Selfish Mining Strategies in Bitcoin,» School of Engineering and Computer Science, Israel, 2015.
- [51] J. Frankenfield, «Investopedia.com - 51% Attack,» 6 May 2019. [En línea]. Available: <https://www.investopedia.com/terms/1/51-attack.asp>. [Último acceso: 6 October 2019].
- [52] I. Eyal y E. G. Sirer, «Majority is not Enough: Bitcoin Mining is Vulnerable,» Financial Cryptography and Data Security, 2014.
- [53] P. Hooda, «GeeksForGeeks - Sybil Attack,» [En línea]. Available: <https://www.geeksforgeeks.org/sybil-attack/>. [Último acceso: 6 October 2019].
- [54] «bitcoin.it - Sybil Attack,» 24 July 2018. [En línea]. Available: https://en.bitcoin.it/wiki/Weaknesses#Sybil_attack. [Último acceso: 7 October 2019].
- [55] «bitcoin.stackexchange.com - Sybil Attack Countermeasure,» 24 November 2017. [En línea]. Available: <https://bitcoin.stackexchange.com/questions/63324/sybil-attack-countermeasure>. [Último acceso: 7 October 2019].
- [56] E. Heilman, «Eclipse Attacks On Bitcoin's Peer-To-Peer Network,» de *24th USENIX Security Symposium*, Washington, 2015.
- [57] «Bitcoin.org - Transaction Malleability,» [En línea]. Available: <https://bitcoin.org/en/transactions-guide#transaction-malleability>. [Último acceso: 22 Diciembre 2019].
- [58] N. Benger, J. van de Pol, N. P. Smart y Y. Yarom, «“Ooh Aah... Just a Little Bit” : A small amount of side channel can go a long,» Australia, 2014.
- [59] «GitHub - Release Notes 0.10.0,» [En línea]. Available: <https://github.com/bitcoin/bitcoin/blob/master/doc/release-notes/release-notes-0.10.0.md>. [Último acceso: 22 Diciembre 2019].
- [60] Groupe IG, «La réglementation des crypto-monnaies dans le monde,» 19 Diciembre 2018. [En línea]. Available: <https://www.ig.com/fr/strategies-de->

- trading/la-reglementation-des-crypto-monnaies-dans-le-monde-181219.
[Último acceso: 27 Octubre 2019].
- [61] The Law Library of Congress, «Regulation of Cryptocurrency in Selected Jurisdictions,» Junio 2018. [En línea]. Available: <https://www.loc.gov/law/help/cryptocurrency/regulation-of-cryptocurrency.pdf>. [Último acceso: 27 Octubre 2019].
- [62] N. Fulmer, «Exploring the Legal Issues of Blockchain Applications,» Akron Law Review, Ohio, 2019.
- [63] «Wikipedia - Libra,» [En línea]. Available: [https://fr.wikipedia.org/wiki/Libra_\(cryptomonnaie\)#R%C3%A9actions_et_critiques](https://fr.wikipedia.org/wiki/Libra_(cryptomonnaie)#R%C3%A9actions_et_critiques). [Último acceso: 28 Diciembre 2019].
- [64] M. Orcutt, «Technology Review - Sorry, but cryptocurrency will likely make you uncomfortable in 2020,» 19 Diciembre 2019. [En línea]. Available: <https://www.technologyreview.com/s/614977/sorry-but-cryptocurrency-will-likely-make-you-uncomfortable-in-2020/>. [Último acceso: 28 Diciembre 2019].
- [65] J. Poon y T. Dryja, «The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments,» 14 Enero 2016. [En línea]. Available: <https://lightning.network/lightning-network-paper.pdf>.
- [66] «Wikipedia - Bitcoin Scalability Problem,» [En línea]. Available: https://en.wikipedia.org/wiki/Bitcoin_scalability_problem. [Último acceso: 28 Diciembre 2019].
- [67] «Hackernoon - The Blockchain Scalability Problem & the Race for Visa-Like Transaction Speed,» 26 Enero 2019. [En línea]. Available: <https://hackernoon.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44>. [Último acceso: 28 Diciembre 2019].
- [68] «bitcoin.org - Standard Transactions,» [En línea]. Available: <https://bitcoin.org/en/transactions-guide#standard-transactions>. [Último acceso: 22 Diciembre 2019].
- [69] «Wikipedia - Máscara de red,» [En línea]. Available: https://es.wikipedia.org/wiki/M%C3%A1scara_de_red.
- [70] «Bitcoin.org - 51 Percent Attack,» [En línea]. Available: <https://bitcoin.org/en/glossary/51-percent-attack>. [Último acceso: 7 September

2019].

- [71] «bitcoin.it - Block hashing algorithm,» [En línea]. Available: https://en.bitcoin.it/wiki/Block_hashing_algorithm. [Último acceso: 5 September 2019].
- [72] «bitcoin.it - Hash Time Locked Contracts,» [En línea]. Available: https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts. [Último acceso: 12 September 2019].
- [73] «Computer Hope - Logic bomb,» 8 Febrero 2019. [En línea]. Available: <https://www.computerhope.com/jargon/l/logibomb.htm>. [Último acceso: 2 September 2019].
- [74] «bitcoin.it - Multisignature,» [En línea]. Available: <https://en.bitcoin.it/wiki/Multisignature>. [Último acceso: 12 September 2019].
- [75] «Wikipedia - Turing Completeness,» [En línea]. Available: https://en.wikipedia.org/wiki/Turing_completeness. [Último acceso: 12 September 2019].

Referencias Generales

- [1] «Wikipedia,» [En línea]. Available: <https://en.wikipedia.org/wiki/Merkle%E2%80%93tree>.
- [2] «Wikipedia - Lightning Network,» [En línea]. Available: https://en.wikipedia.org/wiki/Lightning_Network.
- [3] «Wikipedia - SHA2,» [En línea]. Available: <https://fr.wikipedia.org/wiki/SHA-2>.
- [4] «bitcoin.it - Miner Fees,» [En línea]. Available: https://en.bitcoin.it/wiki/Miner_fees. [Último acceso: 5 Septiembre 2019].
- [5] «bitcoin.it - Payment Channels,» [En línea]. Available: https://en.bitcoin.it/wiki/Payment_channels.
- [6] «bitcoin.it - Lightning Network,» [En línea]. Available: https://en.bitcoin.it/wiki/Lightning_Network.
- [7] «bitcoin.it - Off-Chain Transactions,» [En línea]. Available: https://en.bitcoin.it/wiki/Off-Chain_Transactions.
- [8] «bitcoin.it - Base58Check Encoding,» [En línea]. Available: https://en.bitcoin.it/wiki/Base58Check_encoding.
- [9] «Wikipedia - Cryptographie sur les courbes elliptiques,» [En línea]. Available: https://fr.wikipedia.org/wiki/Cryptographie_sur_les_courbes_elliptiques.
- [10] J.P. Morgan, «J.P. Morgan deploys blockchain with new correspondent banking network,» 16 Octubre 2017. [En línea]. Available: <https://www.jpmorgan.com/country/US/en/detail/1320562088910>. [Último acceso: 25 Octubre 2019].
- [11] «Medium - The Bitcoin Network,» 12 Febrero 2018. [En línea]. Available: <https://medium.com/coinmonks/the-bitcoin-network-6713cb8713d>. [Último acceso: 5 Septiembre 2019].
- [12] O. Beigel, «99bitcoins - Bitcoin forks,» [En línea]. Available: <https://99bitcoins.com/bitcoin-forks/>. [Último acceso: 7 Septiembre 2019].
- [13] O. Beigel, «99bitcoins - Mempool,» [En línea]. Available: <https://99bitcoins.com/bitcoin/mempool/>. [Último acceso: 7 Septiembre 2019].
- [14] M. Apostolaki, A. Zohar y L. Vanbever, «Hijacking Bitcoin: Routing Attacks on Cryptocurrencies,» IEEE Symposium on Security and Privacy 2017, San Jose, 2017.

- [15] «bitcoin.it - Transaction Malleability,» [En línea]. Available: https://en.bitcoin.it/wiki/Transaction_malleability. [Último acceso: 22 Diciembre 2019].
- [16] «Bitcoin.org,» Binns, Will, [En línea]. Available: <https://bitcoin.org>.
- [17] «BitcoinWiki.org - Simplified Payment Verification,» [En línea]. Available: https://en.bitcoinwiki.org/wiki/Simplified_Payment_Verification. [Último acceso: 28 Agosto 2019].
- [18] «bitcoin.it - Common Vulnerabilities and Exposures,» [En línea]. Available: https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures. [Último acceso: 23 Diciembre 2019].
- [19] M. Apostolaki, G. Marti, M. Jan, A. Zohar y L. Vanbever, «BTC-Hijack,» [En línea]. Available: <https://btc-hijack.ethz.ch>. [Último acceso: 20 Diciembre 2019].
- [20] «bitcoin.it - Hash Time Locked Contracts,» [En línea]. Available: https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts. [Último acceso: 12 September 2019].
- [21] A. Hertig, «CoinDesk - Bitcoin Cash Miners Undo Attacker's Transactions With 51% Attack,» 25 Mayo 19. [En línea]. Available: <https://www.coindesk.com/bitcoin-cash-miners-undo-attackers-transactions-with-51-attack>. [Último acceso: 14 Diciembre 2019].
- [22] «bitcoin.it - Multisignature,» [En línea]. Available: <https://en.bitcoin.it/wiki/Multisignature>. [Último acceso: 12 September 2019].
- [23] «Crypto.Stackexchange.com - What is an elliptic curve cofactor?,» [En línea]. Available: <https://crypto.stackexchange.com/questions/56344/what-is-an-elliptic-curve-cofactor?rq=1>. [Último acceso: 7 Octubre 2019].
- [24] «Wikipedia - Turing Completeness,» [En línea]. Available: https://en.wikipedia.org/wiki/Turing_completeness. [Último acceso: 12 September 2019].
- [25] «bitcoinwiki.org - Bitcoin Weaknesses,» [En línea]. Available: https://en.bitcoinwiki.org/wiki/Bitcoin_weaknesses. [Último acceso: 14 Enero 2020].