Blockchain. Bitcoin

por

Mikel Lezaun

Resumen. En este artículo se describen de forma detallada el sistema *blockchain*, o cadena de bloques, y Bitcoin, primera moneda nativa digital. Bitcoin es una extensa red *peer-to-peer* (P2P), abierta a quien lo desee, descentralizada (todos los nodos son iguales, no hay un nodo central que gestiona la red) y en la que los nodos tienen una copia de la gran base de datos de todas las transacciones. La base de datos es una cadena de bloques formados por agrupaciones de transacciones. Una vez confirmados por los nodos, los bloques de transacciones se van encadenando uno detrás de otro de manera que una pequeña modificación en un bloque de la base de datos requiere modificar todos los bloques sucesores, lo cual técnicamente es irrealizable. Esto confiere seguridad a la base de datos, seguridad que reposa en la propia red, sin depender de terceros. Los bloques los confirman los denominados *nodos mineros* superando una *prueba de trabajo*. El primer minero que consigue realizar la prueba de trabajo recibe una recompensa en nuevos bitcoin. La prueba de trabajo también garantiza que no haya doble pago, es decir, que un mismo dinero no se utilice dos veces. La emisión total de bitcoin está limitada a 21 millones y, dado el ritmo fijado de acuñación de bitcoin, esta cantidad se alcanzará en el año 2140. Además de todos los procesos de Bitcoin, en el artículo se presentan los antecedentes de Bitcoin y las herramientas matemáticas que se utilizan en algunos de dichos procesos.

1. Introducción

Cuando a finales de los años ochenta y durante la década de los noventa del siglo pasado se desarrollaron las implementaciones prácticas de Internet, se fueron creando una serie de protocolos específicos para la transmisión por la red de los distintos tipos de contenido e información. Así, por ejemplo, el protocolo SMTP lo es para transmitir el correo electrónico, el FTP para transmitir ficheros y el protocolo HTTP para transmitir hipertextos. Todo ello hace que Internet funcione eficazmente, tal y como comprobamos a diario.

Con todo, este mapa de protocolos no estaba completo, faltaba el equivalente para las operaciones monetarias. En 1996, el Premio Nobel de Economía Milton Friedman apuntó que «lo único que falta, pero que pronto se desarrollará, es un método fiable de dinero electrónico, mediante el cual se puedan transferir en Internet fondos de A a B, sin que A tenga por qué saber quién es B o B quién es A». Se trataba, pues, de crear un protocolo específico para transmitir dinero nativo digital, para transmitir

valor. Hay que notar que el conocido sistema de pagos PayPal, fundado en 1998, no maneja dinero digital, sino dinero tradicional que la estructura bancaria pone para ser utilizado en Internet.

En 2008, Satoshi Nakamoto, del que hasta la fecha no se sabe nada, ni siquiera quién, quiénes o qué es, publicó un artículo ([3]) en la lista de correo *Cryptography* de Internet, en el que definía un protocolo de comunicaciones *peer-to-peer* (P2P) para transmitir valor, al que llamó Bitcoin. Una *red peer-to-peer* es una red entre pares, todos los nodos se comportan como iguales, no hay jerarquías. El año siguiente, él mismo publicó la primera versión del software Bitcoin, creó la red del mismo nombre y las primeras unidades de moneda, que también denominó bitcoin.[1] En 2010, Nakamoto entregó las riendas del proyecto y las claves del repositorio original a Gavin Andresen, que se convirtió en el desarrollador líder de Bitcoin Core en GitHub, y a continuación desapareció del mapa.

En el sistema bancario actual, para hacer una transferencia de una cuenta bancaria a una de otro banco, se cursa la orden, el banco comprueba si la cuenta del ordenante tiene fondos, resta de su saldo el dinero de la transferencia y comunica al otro banco que debe sumar esa cantidad al saldo de la cuenta del destinatario. Esta gestión no mueve billetes de un lado a otro, simplemente es un cambio en los saldos de las cuentas. Los bancos son los intermediarios y garantes de las transacciones, son quienes mantienen toda la información en sus bases de datos y por ello cobran sus comisiones. En realidad, todo esto lo hace un programa informático. Como se verá, el sistema Bitcoin no necesita de intermediarios ni de bancos, pues toda la gestión está descentralizada y el control de las transferencias es de la propia red y, en definitiva, de los usuarios.

Características de Bitcoin

Veamos las características más destacables de las monedas digitales, en particular de la red *blockchain* (cadena de bloques) de Bitcoin:

- La red blockchain de Bitcoin es abierta, todo el que quiera puede participar, no se necesita autorización ni identificación.
- La red es transfronteriza, se puede transferir de un punto a cualquier otro punto del mundo. En realidad los bitcoin no residen en ningún sitio, no son algo físico.
- La red es neutral, no hay que dar ninguna justificación a una transferencia. La red procesa cualquier transacción independientemente del remitente, destinatario y cantidad.
- La red es resistente a la censura, nadie puede evitar que se realice una transacción.
- La red es pública, todo es verificable por todos, la red es totalmente transparente.

[1] Usaremos bitcoin, con minúscula, para referirnos a la moneda y Bitcoin, con mayúscula, para todo lo relacionado con el protocolo en sí (la red, la base de datos, las claves...).

- La red por sí misma garantiza la seguridad tanto de las transacciones como de la base de datos. La confianza está en la propia red y no en un banco o similar.
- Las transacciones son seudónimas, no se sabe directamente quién es el emisor ni quién el receptor.
- Las transacciones son muy fáciles de realizar. El único coste es la tasa por transferencia que se abona a los denominados mineros que, además, es voluntaria. Ahora bien, hace tiempo que los mineros sólo aceptan las transacciones con las comisiones más altas. Esto hace que, en la práctica, para que una transacción sea aceptada para incluirla en la cadena de bloques la comisión tenga que ser alta, e incluso muy alta en los momentos de mucha demanda.

Orígenes

Se puede decir que la tecnología blockchain y Bitcoin se sustentan en cuatro pilares: las funciones *hash*, la criptografía asimétrica de claves pública y privada, las redes P2P y la prueba de trabajo. Veamos los orígenes de cada uno de estos cuatro pilares.

La primera función hash, denominada Cyclic Redundancy Check, la creó Wesley Peterson en el año 1961. La idea era obtener un resumen de un documento, de un conjunto de datos, que sirviera para comprobar su correcta transmisión y almacenamiento en redes y sistemas digitales. En Bitcoin se utilizan las funciones hash criptográficas SHA-256 y RIPEMD-160 para obtener un resumen criptográfico que identifique de forma única e irrepetible un bloque de datos. La función criptográfica RIPEMD-160 (acrónimo de RACE Integrity Primitives Evaluation Message Digest) la desarrollaron Hans Dobbertin, Antoon Bosselaers y Bart Preneel en Europa, y la publicaron por primera vez en 1996. El conjunto de funciones criptográficas SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512) las diseñó la National Security Agency de Estados Unidos. En 2001, el National Institute of Standards and Technology (NIST), también de Estados Unidos, las publicó como un estándar de seguridad.

Las transacciones que se realizan a través de redes no seguras, por ejemplo Internet, se pueden interceptar. Para proteger la información y dotar de seguridad a las comunicaciones se utilizan sistemas criptográficos. Se pueden considerar dos tipos de sistemas criptográficos, los simétricos y los asimétricos. En la criptografía simétrica se emplea la misma clave para cifrar y para descifrar un mensaje. Por contra, la tecnología de cifrado asimétrica, introducida por Whitfield Diffie y Martin Hellman en 1976, utiliza un par de claves, una pública y otra privada. Estas claves son interdependientes, se generan simultáneamente, y su particularidad es que, a pesar de esa dependencia, de una no se puede obtener computacionalmente la otra. Entre los sistemas criptográficos asimétricos de clave pública y clave privada más utilizados está el RSA, introducido por Ron Rivest, Adi Shamir y Leonard Adleman en 1977. Su seguridad se basa en la dificultad de la factorización de números enteros. En el caso del algoritmo Diffie-Hellman, que sirve únicamente para intercambiar públicamente claves de sistemas simétricos, la seguridad radica en la dificultad del cálculo del logaritmo discreto en un cuerpo finito, que se mostrará más adelante. En 1985,

Tahel ElGamal introdujo un sistema para cifrar mensajes, que hoy se conoce como criptosistema de ElGamal, basado en Diffie-Hellman. Bitcoin utiliza, para la autentificación de las transacciones, un sistema criptográfico asimétrico similar al de ElGamal, pero en el que se sustituye el cuerpo finito por una curva elíptica. Este sistema lo propusieron de forma independiente Neal Koblitz y Victor Miller en 1985. Su seguridad viene de la dificultad del problema del logaritmo discreto elíptico. La elección por Bitcoin de este sistema y no el RSA se debe a que presenta ventajas como, por ejemplo, su mayor eficiencia en la generación de claves, o que, para un mismo nivel de seguridad, sus claves son más cortas.

Una *red peer-to-peer* (P2P), red entre pares o iguales, es una red de ordenadores en la que todos los nodos son iguales entre sí. Las redes P2P permiten el intercambio directo de información entre los nodos interconectados. En ella todos los participantes tienen la misma capacidad de actuación, y en los intercambios pueden ser tanto donantes como demandantes. La primera aplicación P2P la desarrolló Adam Hinkley en 1996. Aunque no fue la primera red P2P, el gran impulso a estas redes lo dio Napster, fundada por Sean Parker y Shawn Fanning a finales de 1999. Napster se creó como un servicio de distribución de archivos de música en formato MP3, que permitía a los aficionados a la música compartir sus colecciones. En sentido estricto, Napster no era una red P2P ya que, aunque las transferencias de los archivos tenían lugar directamente entre dos usuarios, Napster utilizaba servidores centrales para almacenar la lista de usuarios y los archivos que cada uno ponía a disposición de los demás. Así, estos servicios centrales informaban a los clientes sobre quién tenía lo que buscaban y establecía el correspondiente contacto. Por querellas judiciales de las grandes compañías discográficas, Napster se vio obligada a cerrar en el año 2001. La red blockchain, como red P2P abierta y descentralizada, es fruto del desarrollo de estas redes precursoras.

A mediados de los años 90, el *spam*, el correo basura, comenzó a ser un problema. Para intentar combatirlo, haciendo que esta práctica tuviera un coste y no fuera rentable, se introdujo la prueba de trabajo, en inglés *Proof of Work* (PoW). En 1997, Adam Back presentó el algoritmo *hashcash* de prueba de trabajo, que consiste en una prueba de computación que suele ser costosa de realizar, pero fácil de verificar. Su realización requiere tiempo de CPU y, en consecuencia, acarrea un coste ligado al consumo de energía. Así, un uso individual puede ser asumido, pero el coste de un uso masivo no es despreciable. La prueba de trabajo de Bitcoin se deriva directamente del *hashcash* introducido por Adam Back. El propio Nakamoto así lo menciona en su artículo fundacional de Bitcoin.

Antededentes de Bitcoin

En 1994 David Chaum introdujo eCash, la primera criptomoneda, que tenía propiedades de privacidad. Gracias a eCash, un banco podía emitir dinero electrónico y los usuarios transferir ese dinero para realizar pagos. El banco podía verificar la validez del pago, esto es, la existencia del dinero en algún saldo, pero no rastrear ese dinero, no podía saber de quién era ni a dónde iba dirigido. Este dinero electrónico

tuvo buena aceptación por parte de bancos importantes, pero no logró la suficiente «tracción» para sobrevivir y desapareció en 1999.

A principios de los años 90 del siglo pasado surgió en California el movimiento *cypherpunk*. Todos sus miembros compartían un espíritu sumamente libertario, con una preocupación constante por la privacidad, la tecnología, encriptación, política, criptoanarquía y dinero digital. En lo que respecta a la privacidad, constataban que, al igual que los servidores Napster, las monedas digitales de la época precisaban de un actor principal para validar y gestionar el libro de cuentas. Esto ofrecía un punto de falla, un punto atacable y, además, la seguridad residía en un tercero en el que había que confiar. El objetivo de los *cypherpunk* era crear monedas digitales descentralizadas, que no dependieran de terceros. Por su interés en relación con Bitcoin, destacamos dos propuestas nacidas en esta comunidad.

En 1998, Wei Dai propuso B-Money, primera criptomoneda para realizar transacciones entre usuarios anónimos. El registro contable era descentralizado, esto es, en vez de tener un registro contable residente en los bancos, cada participante tendría una copia del registro. La red permitiría a cualquier usuario acuñar moneda a partir de una prueba de trabajo. La propuesta pretendía que B-Money fuera estable, que siempre tuviera el mismo valor de adquisición, que no sufriera inflación. Muchos problemas prácticos se quedaron sin resolver y de hecho B-Money no se llegó a implementar, no pasó de ser una propuesta.

Poco después, Nick Szabo propuso Bit Gold. Ésta era, de nuevo, una moneda descentralizada, pero su objetivo ya no era que fuera estable, que tuviera un poder de adquisición constante, sino que fuera algo así como oro digital. De nuevo la emisión estaba basada en una prueba de trabajo, ya que cada unidad de prueba de trabajo generaría un *bit gold*, una pequeña pepita de oro. Cualquier persona podría realizar la prueba de trabajo y acuñar esa pepita de oro. La emisión de moneda era infinita, pero de alguna manera limitada y descentralizada. Los bit gold se encadenaban según se iban creando; se iba así generando una cadena, que se llamaba Bitchain. Incluso en el nombre, Bitchain fue precursora de Blockchain. Una vez más, esta criptomoneda no se llegó a implementar debido a que quedaron problemas sin resolver.

Hay que notar que ambas monedas, B-Money y Bit Gold, se enfrentaban al mismo reto: cómo mantener un registro contable descentralizado y, a la vez, evitar que el dinero se pudiera utilizar dos veces. La cuestión era: ¿cómo saber si una moneda ya ha sido gastada? Este problema se conoce como *el problema del doble gasto*, y se puede describir como sigue. Si en una red descentralizada se envía un pago a un lugar de la red y otro pago con el mismo dinero a un lugar distinto, ¿cómo pueden detectar los nodos que reciben esas transacciones que esa moneda ya ha sido gastada en otro lado? En el caso de las monedas centralizadas no existe este problema, ya que existe un nodo central que valida y registra las transacciones por orden de llegada. En las redes distribuidas, la cuestión es cómo consensuar el orden de las transacciones sin un actor principal. Y aquí es donde aparece Bitcoin. Lo que realmente hace Bitcoin es resolver el problema del doble gasto en una red descentralizada y abierta. Esto lo hace mediante una prueba de trabajo.

2. PRELIMINARES

NUMERACIÓN EN BASE HEXADECIMAL Y BASE58

En Bitcoin, los números se expresan, dependiendo de su cometido, en base decimal, hexadecimal o en base58. En base hexadecimal se utilizan las 16 cifras `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `a`, `b`, `c`, `d`, `e`, `f`, cuya equivalencia con el sistema decimal va ordenadamente de 0 a 15. El sistema base58 es una simplificación del base64, en el que para evitar posibles confusiones se han eliminado la cifra 0 (cero), las letras l (ele minúscula), I (i mayúscula) y O (o mayúscula) y los símbolos + y −. Así, en base58 se utilizan las siguientes 58 cifras:

`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`, `i`, `j`, `k`, `m`, `n`, `o`, `p`, `q`, `r`, `s`, `t`, `u`, `v`, `w`, `x`, `y`, `z`,
`A`, `B`, `C`, `D`, `E`, `F`, `G`, `H`, `J`, `K`, `L`, `M`, `N`, `P`, `Q`, `R`, `S`, `T`, `U`, `V`, `W`, `X`, `Y`, `Z`,

que se corresponden con los valores decimales 0 a 57.

Las 16 cifras de la base hexadecimal se corresponden con las agrupaciones de 4 bits, por lo que la conversión de hexadecimal a binario y viceversa es inmediata. La numeración en base58 se utiliza para facilitar la transcripción a los usuarios.

FUNCIONES HASH

Una *función hash* o *función resumen* parte de un bloque arbitrario de datos y obtiene un resumen o *hash*, formado por una lista de caracteres, normalmente de longitud fija. Entre las funciones hash se destacan las funciones hash criptográficas seguras. Como su propio nombre indica, se utilizan en criptografía. Sus principales características son las siguientes:

- Son unidireccionales, sirven para obtener el resumen que, en general, es fácil de calcular. Por contra, su seudoinversa[2] no es computable. En la práctica, de un resumen criptográfico no es posible recuperar computacionalmente el bloque de datos del que proviene.
- Son determinísticas, su aplicación a un mismo texto siempre produce el mismo resumen criptográfico.
- Son muy sensibles, una levísima modificación de la entrada produce un resumen completamente diferente.
- Son resistentes a las colisiones, es decir, no es posible en la práctica encontrar dos entradas distintas que produzcan el mismo resumen criptográfico.
- Su tamaño es estándar, es decir, sea cual sea el tamaño de la entrada, que en general suele ser grande, la longitud del resumen criptográfico es la misma.

La funcionalidad básica de las funciones hash criptográficas es que sus resúmenes sirven como una imagen representativa y compactada de una cadena de datos de entrada, y se pueden usar como un posible identificador único de esa cadena.

[2]No decimos simplemente *inversa* porque muy raramente una función hash es inyectiva.

En Bitcoin se utilizan las funciones hash SHA-256 ([5]) y RIPEMD-160 ([6]). Como sus nombres indican, un resumen SHA-256 siempre tiene 256 bits y uno RIPIMD-160 tiene 160 bits. Estos resúmenes criptográficos se expresan respectivamente con 64 y 40 caracteres hexadecimales.

En el anexo 1 se muestra cómo se calcula un resumen SHA-256.

Árbol de Merkle

Supongamos que tenemos un bloque de documentos, cada uno con su resumen criptográfico y que, para reforzar la seguridad del contenido del bloque, se quiere obtener un resumen de todos los resúmenes. Para ello se construye un árbol de Merkle, que consiste en lo siguiente. Se ordenan los resúmenes de los documentos del bloque (no hay que olvidar que los resúmenes no dejan de ser números) y se van juntando de dos en dos en un mismo texto. Si el número de documentos es impar, el último resumen se duplica en un texto. De cada uno de estos nuevos textos formados por un par de resúmenes, se obtiene su resumen. Con los resúmenes obtenidos se vuelve a repetir la operación, esto es, se forman los textos de pares de resúmenes y de cada uno se obtiene su resumen. Se sigue el proceso hasta llegar a un solo resumen criptográfico, que se denomina raíz de Merkle. Así, cualquier modificación en un documento implica la modificación de su resumen, de los resúmenes en los que éste aparece y, a la postre, de la raíz de Merkle.

El árbol de Merkle también tiene otro cometido. Si se quiere comprobar que un documento está incluido en el bloque, no es necesario utilizar todos los resúmenes del bloque. Basta rehacer los resúmenes del camino que va desde el documento a la raíz para comprobar que la raíz de Merkle es la correcta. Por ejemplo, en la figura 1 se muestra que para comprobar si el documento que tiene el resumen 10 está en el bloque, basta rehacer el resumen 5 $(10+11)$, el 2 $(5+4)$ y terminar con el 1 $(2+3)$.

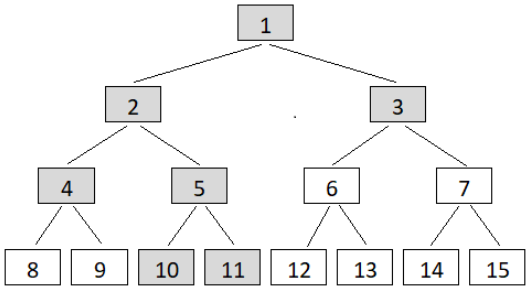

Figura 1: Árbol de Merkle.

En vez de un árbol de Merkle se podría haber considerado una cadena de resúmenes. Esto es, unir en un texto los dos primeros resúmenes y obtener su resumen, el resumen resultante unirlo al tercer resumen y obtener su resumen, a su vez este resumen unirlo con el cuarto y así sucesivamente. Se termina el proceso cuando se

haya pasado por todos los resúmenes del bloque y se obtenga el resumen criptográfico resultante. Ahora bien, este sistema lineal es más costoso que el sistema en árbol a la hora de comprobar la pertenencia al bloque de un resumen.

Firma digital

Una firma digital es un mecanismo criptográfico que permite garantizar al receptor de un mensaje firmado digitalmente que el documento es auténtico, que no ha sido alterado y que quien lo firma no lo puede repudiar. Para la autentificación de las transacciones, para la firma digital, Bitcoin utiliza un sistema criptográfico asimétrico de clave pública y clave privada sobre una curva elíptica criptográfica; resumiendo, un sistema ECDSA, de Elliptic Curve Digital Signature Algorithm. La clave pública es de libre acceso, mientras que la clave privada es secreta: quien haya creado las claves las debe guardar de forma muy segura. La firma de un documento se realiza utilizando la clave privada y para su verificación se utiliza la clave pública. La clave pública se determina a partir de la clave privada elegida y la seguridad del sistema radica en que la obtención de la clave privada a partir de la clave pública es el problema del logaritmo discreto elíptico, no resoluble en general computacionalmente.[3]

En el anexo 2 se muestran los fundamentos de este sistema de firma digital.

Curva elíptica de Bitcoin

Una *curva elíptica E sobre un cuerpo K* viene definida por una ecuación cúbica de la forma

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

donde a_1, a_2, a_3, a_4, $a_6 \in K$ y $\Delta \neq 0$, siendo Δ un invariante llamado *discriminante* de la curva. La condición $\Delta \neq 0$ asegura que no hay puntos en los que la curva tenga dos o más tangentes.

Si la característica del cuerpo K es distinta de 2 y 3, el cambio de variables

$$(x, y) \mapsto \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} + \frac{a_1^3 + 4a_1a_2 - 12a_3}{24} \right)$$

transforma la curva anterior en una curva de ecuación

$$y^2 = x^3 + ax + b,$$

donde a, $b \in K$. Esta última ecuación se denomina ecuación de Weierstrass de la curva, y el discriminante de la curva así descrita es $\Delta = -16(4a^3 + 27b^2)$.

La importancia de las curvas elípticas radica en que en el conjunto de sus puntos se puede definir una suma de forma que se obtiene un grupo abeliano. Para más detalles, ver el anexo 2.

[3]Por eso se eligen curvas elípticas *criptográficas*, aquéllas cuyo grupo de puntos (ver anexo 2) tiene propiedades que hacen difícil resolver en él el problema del logaritmo discreto.

La curva elíptica criptográfica que utiliza Bitcoin para la firma digital es la llamada secp256k1, una de las propuestas para su uso criptográfico en el documento [1], elaborado por la compañía Certicom para el Standards for Efficient Cryptography Group, y está definida sobre un cuerpo finito $\mathbb{F}_p$ por la ecuación

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

El nombre viene de Standards for Efficient Cryptography (sec), número primo de 256 bits (p256) y Koblitz (k). El número primo p de 256 bits que nos da el cardinal del cuerpo finito $\mathbb{F}_p$ sobre el que se define la curva elíptica de Bitcoin es

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1,$$

que expresado en forma hexadecimal es

$$p = \texttt{fffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f}.$$

El orden $\#E(\mathbb{F}_p)$ del grupo abeliano formado por los puntos de la curva elíptica secp256k1 con coordenadas en $\mathbb{F}_p$ es

$$n = 115792089237316195423570985008687907852837564279074904382605163141518161494337,$$

o, en forma hexadecimal,

$$n = \texttt{fffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141}.$$

Este número n es primo y próximo a p. Esto, junto al hecho de que en la curva secp256k1 resulta relativamente fácil realizar las operaciones aritméticas, son las principales razones de la elección de esta curva en particular. Por ser n primo, todo punto es generador de $E(\mathbb{F}_p)$. En Bitcoin se toma como generador el punto P, denominado punto base, cuyas coordenadas, que tienen una longitud de 256 bits, son, en expresión hexadecimal,

$$X = \texttt{79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798},$$
$$Y = \texttt{483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8}.$$

Así, para todo $Q \in E(\mathbb{F}_p)$ existe su logaritmo discreto en base P, esto es, existe un número s con $0 \leq s < n$, tal que $Q = sP$. Ahora bien, la obtención del logaritmo discreto, es decir, la resolución del problema de logaritmo discreto elíptico (PLDE), es intratable computacionalmente.

Terminados los preliminares, a continuación se describen las distintas fases y herramientas de Bitcoin.

3. Monedero o wallet

En el monedero se almacena la moneda y sirve para adquirirla y transferirla.

Crear una cuenta para adquirir bitcoin es muy fácil. Basta con registrarse en alguna de las múltiples páginas web dedicadas a ello e introducir los datos personales: nombre, apellidos, correo electrónico, número de teléfono, fotocopia del DNI, tarjeta de crédito para cambiar euros o dólares por bitcoin, y contraseña. Una vez registrado, para operar se necesita instalar un monedero o *wallet*, que no es más que una simple aplicación informática que permite transferir, recibir y almacenar moneda. Al igual que cuentas bancarias, una misma persona puede tener varios monederos. Instalado el monedero, él mismo crea tantos pares de claves pública y privada como quiera en el sistema de firma digital que se acaba de indicar. Como se verá más adelante, la clave pública de un monedero sirve para recibir bitcoin y la privada para transferirlos.

A partir de la clave pública, el monedero genera de forma determinista su dirección, que consta de 200 bits. Veamos con un ejemplo las distintas operaciones para obtener la dirección Bitcoin a partir de la clave pública. Todos los resúmenes criptográficos se expresan en base hexadecimal.

Sea la clave pública en forma comprimida del ejemplo que se presentará en el anexo 2, cuya expresión hexadecimal es

`020766fd4ffbbfa2086a96b335ba6eff24b58e4819936f8dc9d4002dabc9b11405`.

Para calcular la dirección, se calcula su resumen SHA-256,[4]

`5be0cdfe5b7d843440e13e12c6bcf78356fc33449ed9284817193afaa90c134a`.

De este resumen se calcula su resumen RIPEMD-160[5] y al resultado se le añade por delante `00`,

`009ef1e007b55c584f5a58d719a034e9339c04ed98`.

Se calcula su resumen SHA-256,

`2b0fe96e0bc4f6a467d0de3fdf05794b10a26e95a49032e6cff0086956f8da50`.

Se vuelve a calcular el resumen SHA-256 de esta última expresión,

`c5e29ab4130cee1666bf679b3ae86aef2a6a0c973197e8ca4f2c7559f93cc7c7`.

Se seleccionan los cuatro primeros bytes de este número, o, lo que es lo mismo, sus ocho primeras cifras hexadecimales `c5e29ab4`, y se añaden al final del resumen RIPEMD-160. Estos ocho dígitos son dígitos de control. Se tiene así la dirección Bitcoin con 200 bits, que escrita en hexadecimal es

`009ef1e007b55c584f5a58d719a034e9339c04ed98c5e29ab4`.

Para terminar, se escribe este número en base58,[6]

`1FVRX63gUDxkQ25C7ivMa27C9nT1DUafmD`.

[4] `https://emn178.github.io/online-tools/sha256.htm`
[5] `https://hash.rfctools.com/ripemd160-hash-generator/`
[6] `https://incoherency.co.uk/base58/`

Para la dirección Bitcoin se utiliza el resumen RIPEMD-160 por ser más corto que el SHA-256, y la codificación en base58 porque acorta la longitud final y es más fácil de retener. De las direcciones Bitcoin y su clave pública no se puede deducir directamente la identidad de su propietario. Como todas las transacciones de Bitcoin se hacen entre direcciones y, aunque sean públicas, son seudónimas, no se sabe directamente quién es el emisor ni quién el receptor.

4. Transacciones

> *El propio sistema garantizará que la moneda transferida es propiedad de quien realiza la transacción.*

Una transacción se puede interpretar como el vehículo para transferir bitcoin de un usuario A a uno B. Una transacción esencialmente consta de entradas (*inputs*) y de salidas (*outputs*). Las entradas contienen la información de la o de las transacciones de las que se extraen las cantidades a transferir. Las salidas contienen la dirección a la que se destina el envío, que incluye su clave pública, y la cantidad a transferir. Las informaciones que contienen una salida de una transacción no utilizada y la dirección a que van dirigidas se denominan UTXO, del inglés *Unspent Transaction Output* (Salida de Transacción No Gastada). Las UTXO están depositadas en el monedero de la dirección de la salida, pero son públicas, cualquier usuario tiene acceso a ellas. Ahora bien, ¿de quién es la salida de una transacción?, ¿quién puede utilizar la moneda de esta salida de transacción para hacer un gasto, para realizar con ella una transacción? Únicamente lo puede hacer quien posea la clave privada de la clave pública de la salida de la transacción. Si se pierde la clave privada nadie puede utilizar el dinero de esa transacción y desaparece. Y si alguien se apodera de la clave privada, el dinero es suyo, lo puede utilizar sin ninguna cortapisa. Un usuario lo que realmente tendrá en propiedad son claves privadas de salidas de transacciones no gastadas.

Cuando un usuario quiere utilizar una UTXO, crea una transacción en la que en la entrada constan los datos de esa salida sin gastar. En la salida de la transacción incluirá la dirección de destino con su clave pública. Así pues, las transacciones están ligadas, las salidas de una son entrada de otra y, como son públicas, se puede seguir su pista. Las salidas de transacciones no gastadas sólo se pueden utilizar una única vez, en cuanto forman parte de la entrada de una transacción se gastan, dejan de ser no gastadas. La cuestión es cómo asegurar que quien crea la transacción es propietario de las UTXO que incluye en la entrada. Para ello, mediante el algoritmo de firma digital descrito, autentifica la transacción con la clave privada de la clave pública de la salida no gastada que utiliza en la entrada. Esta clave privada la tendrá bien guardada quien creó esas claves pública y privada, que no es otro que el destinatario de la salida de la transacción no gastada utilizada. Como se ha indicado, con la clave pública de la salida no gastada que se indica en la entrada, cualquiera puede verificar que la transacción es correcta; en particular, que dispone de los fondos que indica en la entrada de la transacción.

El montante de una salida de transacción siempre se transfiere en bloque. Ahora bien, ¿cómo se hace una transacción de 3.9 bitcoin, cuando se quiere utilizar una salida de transacción no gastada de 5.3 bitcoin? Como se ha indicado, en cuanto se utilice esta salida de transacción de 5.3 bitcoin, no se puede volver a utilizar en una entrada, y en principio los 1.4 bitcoin restantes se perderían. El procedimiento ideado para solventar esta dificultad es que, en estas situaciones, efectivamente la entrada de la transacción es la salida no gastada de 5.3 bitcoin, pero se crean dos salidas, una de 3.9 bitcoin a la dirección que se quiere enviar y otra de 1.4 bitcoin a una dirección propia. De esta forma, la salida no gastada de la autotransacción con 1.4 bitcoin sólo la podrá utilizar el emisor, que así no pierde los 1.4 bitcoin. Hay que destacar que toda transacción debe incluir también el pago de una pequeña comisión para quienes mantienen la red ayudando a validar las transacciones para incluirlas en un bloque (los mineros). Esta comisión es voluntaria, la decide quien emite la transacción, pero cuanto mayor sea más solicitada estará por los mineros para incluirla en el bloque que generan. En definitiva, en cualquier transacción, las entradas menos la comisión es igual a las salidas. Existen unas transacciones especiales, denominadas *coinbase*, que, como se explicará más adelante, son con las que los mineros reciben la remuneración principal por minar.

Para terminar, además de las entradas y salidas, las transacciones contienen una cabecera donde consta su resumen SHA-256 y el número de entradas y salidas. Como norma, de la misma forma que se ha mostrado al definir la dirección, en Bitcoin los resúmenes se duplican: una vez obtenido un resumen, se calcula el resumen del resumen.

En este sistema de transacciones, no hay ninguna cortapisa para realizar una transacción y es la propia red quien garantizará la validez de la transacción.

Para facilitar las transacciones, un bitcoin está fraccionado en satoshi. Un bitcoin son 10^8 satoshi.

5. Nodos de la red

> *La red es descentralizada. Los nodos de la red se encargan de distribuir las transacciones realizadas, de verificarlas, de agruparlas en bloques, de confirmar los bloques, de transmitirlos y de almacenarlos.*

La red Bitcoin es una red P2P descentralizada y abierta, cualquier persona que lo desee puede unirse libremente a la red con solo descargar e instalar en su ordenador el software adecuado. En Bitcoin existen varios tipos de nodo, cada uno con sus propias funciones específicas.

Los *nodos completos* son los nodos conectados a la red que almacenan una copia exacta, completa y actualizada de la cadena de bloques. Son los que validan las transacciones y los bloques de la cadena, y los que los propagan por la red a otros nodos. Además, verifican la profundidad de los bloques, esto es, cuántos bloques válidos siguen a un bloque, para comprobar la posibilidad de que el bloque se desestime por existir una cadena de bloques con mayor prueba de trabajo. Sobre esto se volverá más adelante. En definitiva, los nodos completos son los encargados de

verificar que se cumplen todas las reglas del protocolo Bitcoin y, por tanto, son los que verdaderamente otorgan robustez, seguridad y estabilidad a la red.

Una versión simplificada de los nodos completos son los *nodos podados*. Estos nodos cumplen casi todas las funciones de los nodos completos. Su única excepción es que no tienen una copia completa de la cadena de bloques, sólo mantienen una parte substancial de la misma, con la que pueden realizar procesos de verificación como cualquier nodo completo.

Entre los nodos completos se destacan los *nodos públicos*, que, como su nombre indica, operan en la red de forma pública y abierta. Estos nodos ejercen de forma continua e ininterrumpida como punto de comunicación e interconexión con otros nodos de la red para transmitir datos e información. Cualquier nodo que desee conectarse con un nodo público lo puede hacer de manera abierta cuando quiera.

Los *nodos de minería* son los nodos completos que, además, realizan lo que se denomina *minería*, que consiste en buscar las pruebas de trabajo. Hoy en día, y debido a la enorme complejidad de la red, para dedicarse a esta actividad hace falta disponer de hardware especializado denominado ASIC, siglas en inglés de Circuito Integrado para Aplicaciones Específicas (*Application Specific Integrated Circuit*). Para facilitar el trabajo de minería, también existe la posibilidad de unir el poder computacional de un grupo de mineros, constituyendo un *pool* de mineros. En este caso, los beneficios obtenidos por el minado se reparten entre los miembros del *pool* de forma proporcional al trabajo realizado.

Por último, están los *nodos ligeros*, también conocidos como clientes de Verificación de Pago Simplificada o SPV (abreviatura de *Simplified Payment Verification*). Los nodos ligeros hacen uso de la red Bitcoin pero no actúan como un nodo completo. Son nodos que no mantienen una copia de la cadena de bloques, no participan en el proceso de verificación y validación de las transacciones, y no contribuyen a la seguridad de la red. Los nodos ligeros pueden solicitar y recibir de los nodos completos públicos el estado de una transacción concreta, sin tener que descargar la base de datos de la cadena completa. En particular, pueden obtener información de sus propias transacciones y saber si están incluidas o no en un bloque. Así, los nodos ligeros dependen de las verificaciones de los nodos completos y confían en ellas para garantizar la seguridad. Los nodos ligeros trabajan únicamente como puntos finales de comunicación y son ejecutables en dispositivos móviles como teléfonos y tabletas.

6. Bloques de la cadena

> *La agrupación de las transacciones en bloques encadenados impide cualquier pequeña modificación en operaciones confirmadas y almacenadas en la base Bitcoin.*

Una ver realizadas, las transacciones se transmiten entre los nodos mediante el protocolo de comunicación de la red Bitcoin y quedan a la espera de ser confirmadas. Los nodos completos reciben las transacciones y, una vez que verifican su validez, las retransmiten. Los mineros agrupan las transacciones en un bloque en construcción. Una vez que se mina y confirma ese bloque, cada bloque posterior que se mine sobre

él le aporta un mayor nivel de seguridad. Se acepta como suficientemente seguro un bloque que tenga confirmados sus seis bloques siguientes.

Los nodos mineros son los encargados de recopilar y agrupar en un bloque las transacciones que están pendientes en la red. Cada minero selecciona las transacciones que desea incluir y construye su propio bloque. Como se ha indicado anteriormente, los mineros prioritariamente eligen las que aportan mayor comisión. Si existen transacciones ya confirmadas e incluidas en bloques anteriores, se eliminan de éste. Este nuevo bloque se conoce como *candidato*. El tamaño de un bloque tiene que ser inferior a un megabyte.

La primera transacción del bloque es especial y se denomina *transacción coinbase*. Sólo hay una transacción coinbase en cada bloque. Esta transacción no tiene entrada, y su salida está dirigida a una dirección del minero; contiene los nuevos bitcoin acuñados para pagar el minado. La salida del coinbase también contendrá las comisiones de las transacciones del bloque. Obviamente, las salidas las podrá utilizar el minero para cualquier transacción que quiera realizar.

En la formación de un nuevo bloque se incluye en el encabezado el resumen criptográfico del bloque anterior y la raíz de Merkle de los resúmenes de las transacciones contenidas en el bloque. La inclusión del resumen del bloque anterior es la forma en que se van encadenando los bloques. Así, cualquier modificación en una transacción almacenada requerirá cambiar el resumen de su bloque y, como este resumen está contenido en el bloque siguiente, también se deberá cambiar su resumen, y así sucesivamente. En definitiva, modificar una transacción almacenada en un bloque requiere modificar el resumen de su bloque y el de todos los bloques siguientes, lo cual resulta técnicamente irrealizable. Ésta es la gran garantía de seguridad de la base de datos Bitcoin.

Además del resumen criptográfico y la raíz del árbol de Merkle, en la cabecera del bloque se incluyen los datos del minado que se describen en la siguiente sección y una marca de tiempo.

La marca de tiempo, *timestamp* en inglés, la incluye el nodo minero y la comprueban los restantes nodos. Para aceptar el bloque, esta marca debe estar comprendida entre la mediana de las marcas de tiempo de los once últimos bloques y dos horas después de la hora actual de la red. Implementar una marca de tiempo imposibilita la duplicación posterior de un bloque, ya que, además de la hora, también se almacena la fecha de creación del bloque. En consecuencia, si se intenta más tarde replicar un bloque, para que el bloque se confirme hay que añadir la marca de tiempo actual, lo cual necesita, una vez más, modificar el resumen y los resúmenes de los bloques posteriores, lo que es irrealizable.

7. Minado de un bloque. Prueba de trabajo

La prueba de trabajo regula el ritmo de la formación de los bloques y evita comportamientos indeseados.

Una *prueba de trabajo* o PoW, del inglés *Proof of Work*, es un sistema que tiene por finalidad desincentivar y dificultar comportamientos indeseados, como ataques

que generen un gran flujo de información desde varios puntos hacia un mismo punto de destino, haciendo que se colapse la red. La forma más común de realizar estos ataques es mediante una red de robots informáticos (*bots*). Para ello, la prueba de trabajo exige que el usuario del servicio realice algún tipo de trabajo que tenga cierto coste y que su realización sea fácil de comprobar. La clave está en que es una estrategia asimétrica: la prueba de trabajo es factible pero más o menos costosa, mientras que la comprobación de su validez es sencilla.

En Bitcoin, una vez conformado un bloque, para confirmarlo hay que minarlo, hay que realizar una prueba de trabajo. La prueba de trabajo consiste en la creación del resumen del bloque de forma que cumpla la condición de que sea menor que un número fijado, que en hexadecimal es un número que comienza por unos cuantos ceros. Este número, de 256 bits, se denomina *target*. En Bitcoin, cada bloque se conforma y mina más o menos cada diez minutos, y el primer minero que supere la prueba de trabajo recibe una recompensa en bitcoin de nueva creación. Controlando la dificultad del minado, se consigue mantener un ritmo de más o menos diez minutos en la creación de los bloques.

La dificultad del minado es una medida de la cantidad de recursos necesarios para minar Bitcoin, que sube o baja dependiendo de la cantidad de potencia computacional disponible en la red. En Bitcoin, la dificultad, que notamos dfct, se ajusta automáticamente cada 2016 bloques, más o menos dos semanas, mediante la fórmula

$$\mathrm{dfct}' = \mathrm{dfct} \cdot \max\left(\min\left(\frac{2016T}{T_A}, 4\right), \frac{1}{4}\right),$$

donde T es el tiempo ideal entre dos bloques, en Bitcoin 10 minutos, y T_A el tiempo utilizado para minar los últimos 2016 bloques. Esta fórmula está así diseñada para que el coeficiente que multiplica a la dificultad dfct esté comprendido entre $1/4$ y 4, con el fin de que los cambios de dificultad no sean demasiado bruscos.

La dificultad del minado de un bloque se expresa con el cociente entre un target de referencia y el target actual,

$$\mathrm{dfct} = \frac{\text{target de referencia}}{\text{target del bloque}}.$$

En Bitcoin, el target de referencia es el del primer minado, es decir, el del bloque génesis, que es

00000000ffff0000000000000000000000000000000000000000000000000000.

Una vez calculada la nueva dificultad, inmediatamente se actualiza el target, que se mantendrá durante 2016 bloques. De esta forma, aunque varíe la capacidad computacional de la red, se consigue controlar el tiempo de unos 10 minutos entre cada bloque.

El procedimiento para superar la prueba de trabajo consiste en añadir al bloque un número arbitrario, que sólo se puede usar una vez, para que el resumen resultante cumpla la condición exigida. Este número se denomina *nonce* o *número de un solo uso*. La única forma de obtener un nonce es por fuerza bruta. Este proceso requiere un

intensivo trabajo computacional con hardware especializado (ASIC) y consume gran cantidad de energía eléctrica. A esto se debe la creación de *pools* de mineros en países como China, en los que la energía es barata y el minado resulta rentable. Como ya se ha indicado, la dificultad y coste de esta prueba de trabajo evita comportamientos maliciosos como, por ejemplo, ataques masivos que colapsen la red, y, además, regula el ritmo de confirmación de los bloques. Este ritmo regulado de la creación de bloques hace que al final vaya habiendo una única versión de la cadena de bloques, en lugar de una distinta por cada nodo de la red, lo que a su vez permite impedir el doble gasto.

Una vez que un minero encuentra un nonce, lo anuncia inmediatamente a los otros mineros, éstos comprueban la prueba de trabajo y las transacciones y deshacen sus bloques candidato. La comprobación del resultado del nonce es fácil e inmediata, los mineros la hacen y el bloque, junto con todas las transacciones que contiene, queda confirmado e incluido en la cadena de bloques.

8. Bloques huérfanos

La prueba de trabajo impide el doble gasto de moneda.

Un problema a solucionar es el del doble gasto, que una misma moneda no se pueda gastar dos veces. Esta dificultad es propia de las monedas descentralizadas, ya que, al no haber un centro que controle la llegada de las transacciones, ante dos iguales no siempre es posible decidir cuál se ha producido la primera y darle a ésa la validez. En Bitcoin se da con frecuencia el caso de que, a partir de un mismo bloque, el B_0 de la figura 2, dos mineros minen casi simultáneamente sus bloques, los bloques $B_{1,1}$ y $B_{1,2}$ de la figura, y los distribuyan a la red. Como son los propios mineros los que conforman los bloques, los dos bloques serán parecidos, pero no iguales, y tendrán distinto resumen. En esta situación, unos mineros trabajarán con uno de los bloques para conformar y minar un nuevo bloque, y otros lo harán con el otro bloque. ¿Cuál de los bloques simultáneos $B_{1,1}$ y $B_{1,2}$, y por tanto sus transacciones, se aceptará? Es muy raro que los bloques que seguirán a estos dos también se vuelvan a minar al mismo tiempo, habrá uno que se conseguirá antes, por ejemplo el B_2 que sigue al $B_{1,1}$. Los mineros que tenían como válido el bloque $B_{1,2}$ comprobarán que este bloque B_2 es incompatible como siguiente al suyo, profundizarán en las dos cadenas hasta llegar al primer bloque común, el B_0, y darán por válida la cadena que parte de un bloque con mayor prueba de trabajo, en el dibujo la cadena $B_{1,1} - B_2$. La otra cadena, en este caso el bloque $B_{1,2}$, quedará descartada. A este tipo de bloques descartados se los denomina *bloques huérfanos*.

Los bloques huérfanos son válidos, se han obtenido correctamente, pero no forman parte de la cadena de bloques, se descomponen, se ignoran. Las transacciones de un bloque huérfano que no estén incluidas en otros bloques no se pierden, quedarán en espera a ser incluidas en un nuevo bloque y ser validadas. Una vez invalidado un bloque, se invalida la recompensa a recibir por minado y el minero que realizó la prueba de trabajo se quedará sin recompensa.

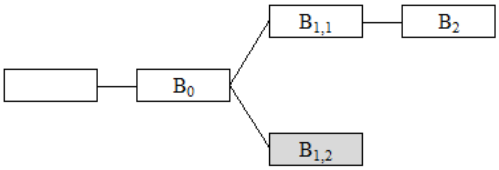

Figura 2: Bloques huérfanos.

Este mismo criterio se sigue cuando a partir de un mismo bloque se generan dos cadenas paralelas válidas en sí mismas, pero que son incompatibles, que no pueden coexistir. Siempre se validará la cadena que parte de un bloque con mayor prueba de trabajo. Los mineros que hayan trabajado en la cadena invalidada se quedarán sin recompensa. Así, cuando por azar aparecen dos cadenas incompatibles entre sí, no interesa insistir en imponer la propia, ya que se corre el riesgo de perder toda la recompensa.

9. Recompensa por minar. Limitación del número total de bitcoin

La acuñación de nuevos bitcoin se destina a pagar a los mineros que superan en primer lugar la prueba de trabajo.

Una de las claves de la superveniencia de Bitcoin es que, por cada bloque minado, quien lo consigue en primer lugar recibe una recompensa en nuevos bitcoin. En concreto, todos los bitcoin que se van acuñando son para los mineros que consiguen en primer lugar el minado. Ellos son también quienes reciben las comisiones de las transacciones. Está programado que cada 210 000 bloques, unos cuatro años a diez minutos por bloque, la emisión de bitcoin se reduzca a la mitad. En consecuencia, la remuneración del minado por bloque también se reduce a la mitad. En los cuatro primeros años, 2009 a 2012, se pusieron en circulación 10 500 000 bitcoin y por el minado de cada bloque se pagaron 50 bitcoin ($210\,000 \times 50 = 10\,500\,000$). A partir de mayo de 2020, el pago por minado se ha reducido a 6.25 bitcoin, por lo que en los próximos cuatro años se acuñarán en total 1 312 500 bitcoin. La producción total de bitcoin está limitada a 21 millones y a este ritmo finalizará en el año 2140. A partir de esa fecha, los mineros sólo recibirán por un minado las comisiones de las transacciones. Otra cuestión muy distinta es por cuántos euros o dólares se cambie un bitcoin.

Veamos como ejemplo el bloque número 672431,[7] de fecha 2021-02-27 13:28:

- Bloque: 672431
- Resumen: `0000000000000000000003120e9aa5b6f1247a730c4b0dfc6716db4b855c978da5`
- Raíz de Merkle: `6b44315f7b2621d8a21554e1dc96178bd935d042e23cb77e75bf4b1f4fd78ff1`
- Número de transacciones: 2096

[7] `https://www.blockchain.com/es/btc/blocks`

- Tamaño: 1 281 576 bytes
- Volumen de la transacción: 1446.34840523 bitcoin
- Recompensa por minado: 6.25 bitcoin
- Remuneración por comisiones: 0.41695539 bitcoin
- Dificultad: 21 724 134 900 047.27
- Target: `00000000000000000cf4e3000000000000000000000000000000000000000000`
- Nonce: `04e9a917`

Es inmediato comprobar que el resumen es menor que el target.

10. Comentario final

Bitcoin es una extensa red P2P abierta y descentralizada de ordenadores repartida por todo el mundo y una inmensa base de datos distribuida que contiene la historia completa de todas las transacciones que se realizan en la red. Como red abierta, cualquier persona que lo desee se puede integrar en ella. Como red P2P descentralizada, todos los nodos son iguales, no hay un punto central de conexión, y todas las partes actúan de forma autónoma siguiendo un mismo protocolo de comunicaciones y consenso. La red es distribuida, los nodos tienen una copia de la gran base de datos de todas las transacciones. Las transacciones se autentifican y validan con la firma digital. La base de datos es una cadena de bloques formados por agrupaciones de transacciones. Cada bloque contiene el resumen criptográfico del anterior, por lo que los bloques se van encadenando uno detrás de otro. Así, una pequeña modificación en un bloque requiere modificar todos los bloques sucesores, lo cual confiere seguridad a la base de datos, seguridad que recae en la propia red, sin necesidad de terceros. Los bloques los confirman, a un ritmo de unos diez minutos por bloque, los denominados nodos mineros, superando una prueba de trabajo. El primer minero que consigue realizar la prueba de trabajo recibe una recompensa en bitcoin acuñados para eso. Todos los bitcoin que se van creando se destinan a las recompensas para los mineros. El sistema de recompensas hace que el número total de bitcoin nunca va a ser superior a 21 millones. La prueba de trabajo también garantiza que no haya doble pago, que un mismo dinero no se utilice dos veces.

Para terminar, unas observaciones sobre Bitcoin.

Las monedas al uso tienen que cumplir tres funciones: permitir almacenar valor, realizar intercambios y transacciones, y servir como unidad de cuenta. Esto último significa servir de referencia de valor. Por ejemplo, basta con leer el precio de un teléfono móvil para hacerse una idea de su valor. Según estos criterios, no se puede decir que las criptomonedas sean monedas de verdad, ya que no cumplen estas tres propiedades. La más próxima a cumplirse es la de almacenamiento de valor. Pero como medio de intercambio no se acepta en demasiados sitios y, debido a su constante cambio de valor, tampoco sirve como referencia. Las criptomonedas no serán monedas de verdad mientras no se cumplan estas tres propiedades. En particular, mientras no se generalice cambiar bitcoin por, por ejemplo, un coche, un billete de avión o la realización de un trabajo.

En cuanto al valor, es importante destacar que en total se emitirán 21 millones de bitcoin y que en 2020 ya han sido acuñados alrededor de 18 593 000 bitcoin de los 21 millones totales. Se estima que alrededor de tres o cuatro millones de bitcoin han desaparecido por pérdida de las claves privadas, errores en las direcciones, etc. Dado que cada cuatro años se reduce a la mitad el número de bitcoin que se acuñan, se alcanzarán los 21 millones en el año 2140.

Existen en el mercado casas de cambio que permiten cambiar bitcoin por dólares o euros y viceversa. Así, quien quiera puede adquirir bitcoin e ingresar en la red Bitcoin. En sus dos primeros años, 2009 y 2010, el valor del bitcoin era prácticamente cero y el 2 de enero de 2011 valía 5 dólares. En enero de 2015 se cambiaba por alrededor de 300 dólares y desde entonces, con grandes altibajos, no ha dejado de acumular valor. ¿Hasta cuánto podrá aumentar el valor de un bitcoin? ¿Se está ante una burbuja?

Anexo 1. Cálculo del resumen SHA-256

Para simplificar la exposición, se denomina *palabra* a una secuencia de 32 bits, que se corresponde con 8 dígitos hexadecimales.

Paso 1. Extensión del texto binario a bloques de 512 bits

Se empieza añadiendo al final del texto en binario los cuatro bits 1000 (número 8 hexadecimal) y a continuación, si es necesario, tantos 0 como haga falta para que el número de bits sea congruente con 448 módulo 512. Se tiene así un último bloque de 448 bits (14 palabras) y, según sea la longitud del texto, uno, varios o ningún bloque de 512 bits (16 palabras). Una vez hecho esto, el último bloque de 448 bits se completa hasta 512 bits añadiendo en los últimos lugares el número de bits que tiene el texto original, y por delante de este número tantos bits 0 como sean necesarios para completar los 512 bits. Por ejemplo, el texto **covid-19** en codificación hexadecimal es `636f7669642d3139`. Este texto de 16 símbolos hexadecimales tiene 64 bits, que en hexadecimal es 40. En este caso, el bloque de 512 bits está formado por las siguientes 16 palabras:

$$\begin{array}{cccc} W_0 = \texttt{636f7669}, & W_1 = \texttt{642d3139}, & W_2 = \texttt{80000000}, & W_3 = \texttt{00000000}, \\ W_4 = \texttt{00000000}, & W_5 = \texttt{00000000}, & W_6 = \texttt{00000000}, & W_7 = \texttt{00000000}, \\ W_8 = \texttt{00000000}, & W_9 = \texttt{00000000}, & W_{10} = \texttt{00000000}, & W_{11} = \texttt{00000000}, \\ W_{12} = \texttt{00000000}, & W_{13} = \texttt{00000000}, & W_{14} = \texttt{00000000}, & W_{15} = \texttt{00000040}. \end{array}$$

Paso 2. Extensión de cada bloque de 512 bits (16 palabras) a 2048 bits (64 palabras)

La extensión de los bloques de 512 bits a 2048 bits se realiza por recurrencia con las siguientes fórmulas:

$$W_i = \sigma_1(W_{i-2}) \boxplus W_{i-7} \boxplus \sigma_0(W_{i-15}) \boxplus W_{i-16}, \quad i = 16, 17, \ldots, 63$$

con

$$\sigma_0(W) = (W \ggg 7) \oplus (W \ggg 18) \oplus (W \gg 3),$$
$$\sigma_1(W) = (W \ggg 17) \oplus (W \ggg 19) \oplus (W \gg 10),$$

donde $W \ggg k$ denota una rotación de k bits hacia la derecha de los bits de la palabra W; $W \gg k$ un desplazamiento de k bits hacia la derecha de los bits de la palabra W; $\oplus$ la suma bit a bit y $\boxplus$ la suma mód 16^8 (en hexadecimal mód 100000000), para que el resultado sea una palabra de ocho cifras hexadecimales, 32 bits.

PASO 3. OBTENCIÓN DE RESÚMENES INTERMEDIOS

De cada bloque de 64 palabras se va obteniendo de forma encadenada un resumen intermedio. Para ello se realiza una avalancha de 64 iteraciones, una por cada palabra del bloque. En los cálculos de cada paso interviene la correspondiente palabra W_i y una palabra constante K_i. Las constantes K_i, $i = 0, 1, \ldots, 63$, son los primeros 32 bits de la parte fraccionaria, en hexadecimal, de la raíz cúbica de los 64 primeros números primos.

En el primer bloque se parte de la palabra inicial $ABCDEFGH$ donde

$$A = \texttt{6a09e667}, \quad B = \texttt{bb67ae85}, \quad C = \texttt{3c6ef372}, \quad D = \texttt{a54ff53a},$$
$$E = \texttt{510e527f}, \quad F = \texttt{9b05688c}, \quad G = \texttt{1f83d9ab}, \quad H = \texttt{5be0cd19}.$$

Estas constantes son, respectivamente, los primeros 32 bits de la parte fraccionaria, en hexadecimal, de la raíz cuadrada de los ocho primeros números primos.

A partir de esta palabra inicial $ABCDEFGH$ se obtiene una primera palabra $A_1B_1C_1D_1E_1F_1G_1H_1$, de ésta una segunda palabra, de la segunda una tercera y así, siempre siguiendo el mismo esquema, hasta llegar a la palabra 64. El diagrama del proceso iterativo viene dado en la figura 3.

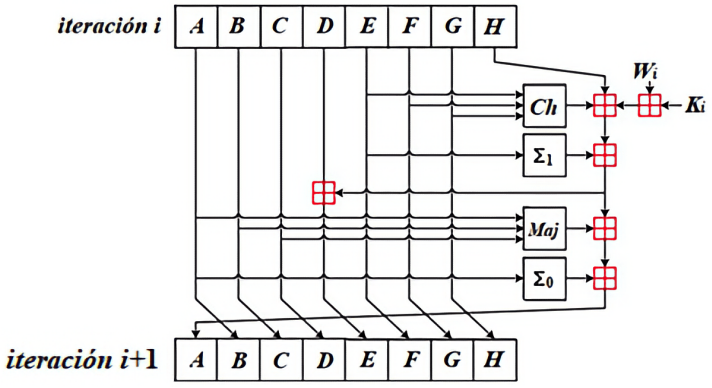

Figura 3: Diagrama de SHA256.

En él, las funciones Ch, Σ_1, Maj y Σ_0 son las siguientes:

$$\begin{aligned}
\mathrm{Ch}(E,F,G) &= (E \wedge F) \oplus (\neg E \wedge G),\\
\mathrm{Maj}(A,B,C) &= (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C),\\
\Sigma_0(A) &= (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22),\\
\Sigma_1(E) &= (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25).
\end{aligned}$$

En estas funciones intervienen los operadores binarios XOR ($\oplus$), AND ($\wedge$), OR ($\vee$) y NOT ($\neg$) definidos en la tabla 1.

X	Y	$X \oplus Y$	$X \wedge Y$	$X \vee Y$	X	$\neg X$
0	0	0	0	1	0	1
0	1	1	0	0	1	0
1	0	1	0	0	-	-
1	1	0	1	0	-	-

Tabla 1: Operadores binarios.

Una vez calculadas las últimas palabras $A_{64}B_{64}C_{64}D_{64}E_{64}F_{64}G_{64}H_{64}$, para terminar el bloque se suman palabra a palabra mód 16^8 con las primeras palabras del bloque. Se obtiene así el primer resumen intermedio. Si lo hay, estas palabras pasan a ser las iniciales del siguiente bloque. Cuando se termina el último bloque, las siete palabras finales se pegan y se obtiene el resumen de 256 bits.

El resumen RIPEMD-160 se calcula de forma similar.

Como ejemplo, el resumen SHA-256 del texto **covid-19** es[8]

`88529c3ac8ebd2dcb21a432c4ea0190c8370850b73ef95a527d150d4d424bc62`,

y, el resumen RIPEMD-160,[9]

`f11afb898948b9a9b179ffde02b3614eafb50767`.

Anexo 2. Criptografía de curvas elípticas. Firma digital

En este anexo se describe el sistema criptográfico que se apoya en el problema del logaritmo discreto en curvas elípticas y, en particular, la firma digital, usando una curva elíptica criptográfica.[10]

[8] `https://emn178.github.io/online-tools/sha256.htm`

[9] `https://hash.rfctools.com/ripemd160-hash-generator/`

[10] Recordemos que decimos que una curva elíptica es *criptográfica* si es difícil resolver el problema del logaritmo discreto en su grupo de puntos.

PROBLEMA DEL LOGARITMO DISCRETO (PLD)

Sea G un grupo y $g \in G$ un elemento de orden n. Sea $\langle g \rangle = \{1, g, g^2, \ldots, g^{n-1}\}$ el subgrupo cíclico de G generado por g. Se puede identificar $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$ con el grupo aditivo de clases residuales mód n. La aplicación $\mathbb{Z}_n \to \langle g \rangle$ dada por $x \mapsto g^x$ es un isomorfismo entre el grupo aditivo $\mathbb{Z}_n$ y el subgrupo $\langle g \rangle$ de G. El isomorfismo inverso consiste en, dado $h \in \langle g \rangle$, hallar $x \in \mathbb{Z}_n$ tal que $g^x = h$. Esto se puede expresar como $x = \log_g h$. Este isomorfismo inverso se denomina *logaritmo discreto*. El problema del logaritmo discreto (PLD) en un grupo G consiste en, dados g, $h \in G$, hallar $\log_g h$, siempre que exista, esto es, si $h \in \langle g \rangle$.

La dificultad del PLD depende mucho del grupo G y del elemento g elegidos. Si se elige como grupo $G = \mathbb{Z}_p$, grupo aditivo de los enteros mód p con p primo, para cualquier $g \in \mathbb{Z}_p$ no nulo se tiene $\langle g \rangle = \mathbb{Z}_p$. El logaritmo discreto consiste en, dado $h \in \mathbb{Z}_p$, hallar $x \in \mathbb{Z}_p$ tal que $xg = h$. En este caso el PLD es fácil de resolver, basta con hallar g^{-1} en el grupo multiplicativo $\mathbb{Z}_p^*$ y $x = hg^{-1}$.[11]

Diffie y Hellman, en su protocolo para intercambiar claves secretas, se centraron en el grupo multiplicativo $\mathbb{Z}_p^*$ con p primo. Éste es un grupo cíclico, pero su orden $p-1$ no es un número primo. No todos los elementos distintos de 1 son generadores; los generadores de $\mathbb{Z}_p^*$ son los elementos primitivos de $\mathbb{Z}_p$, esto es, las raíces $(p-1)$-primitivas de la unidad módulo p. Por tanto, este sistema requiere elegir un p primo y un g elemento primitivo de $\mathbb{Z}_p$. Pero esto no basta para que el PLD no sea fácil de resolver. Para asegurar la dificultad del PLD sobre $\mathbb{Z}_p^*$ hay que elegir un p primo «grande» tal que el orden $p-1$ de $\mathbb{Z}_p^*$ tenga al menos un factor primo «grande». El National Institute for Standards and Technology (NIST) recomienda como mínimo 2048 bits para el orden del cuerpo y 224 para el factor primo.

CURVAS ELÍPTICAS

Como se ha indicado, los criptosistemas basados en la dificultad del PLD sobre el grupo Z_p^* exigen usar números primos «grandes», lo cual hace que requieran gran cantidad de memoria. En 1985, Neal Koblitz y Victor Miller propusieron utilizar criptosistemas basados en el PLD sobre grupos aditivos de curvas elípticas, ya que requieren parámetros mucho más pequeños para un nivel de seguridad dado. Debido a que es el caso que nos interesa, nos centraremos en curvas definidas sobre un cuerpo $\mathbb{Z}_p$ de números enteros módulo p, con p primo distinto de 2 y 3, que a partir de ahora denotaremos (para recordar que lo vemos como un cuerpo) por $\mathbb{F}_p$.

Consideremos las curvas dadas por una ecuación de Weierstrass

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_p, \quad 4a^3 + 27b^2 \neq 0.$$

El conjunto de puntos de $\mathbb{F}_p \times \mathbb{F}_p$ que verifican la ecuación anterior en $\mathbb{F}_p$ se denomina curva elíptica sobre $\mathbb{F}_p$.

Los puntos de una curva elíptica sobre $\mathbb{F}_p$ no forman un continuo, sino que en realidad son una nube finita de puntos en el plano. Pero una referencia muy

[11] Observemos que, al ser p primo, $\mathbb{Z}_p^*$ son todos los elementos no nulos de $\mathbb{Z}_p$, y que el cálculo de g^{-1} a partir de g es muy rápido utilizando el algoritmo de Euclides.

ilustrativa sobre su naturaleza es conocer cómo son estas curvas sobre el cuerpo $\mathbb{R}$. La condición $4a^3+27b^2 \neq 0$, esto es, que el discriminante de x^3+ax+b no sea nulo, equivale a que el polinomio cúbico x^3+ax+b no tenga ceros múltiples, y a que la curva elíptica definida por este polinomio sea no singular. Estas curvas tienen las siguientes propiedades:

- son simétricas respecto al eje de las x;
- dados dos puntos P y Q, si la recta que los une no es vertical, esta recta vuelve a cortar a la curva en un único punto distinto de P y Q;
- dado un punto P de la curva, si la recta tangente a la curva en P no es vertical, esta recta vuelve a cortar la curva en un único punto;
- si la recta que une dos puntos P y Q o la recta tangente en un punto P son verticales, estas rectas no cortan a la curva en otro punto.

Estas propiedades propician definir una suma en el conjunto de puntos de la curva elíptica sobre $\mathbb{R}$ más un punto denominado *punto del infinito*, que notamos por 0. Esta suma se ilustra en la figura 4 y se describe como sigue.

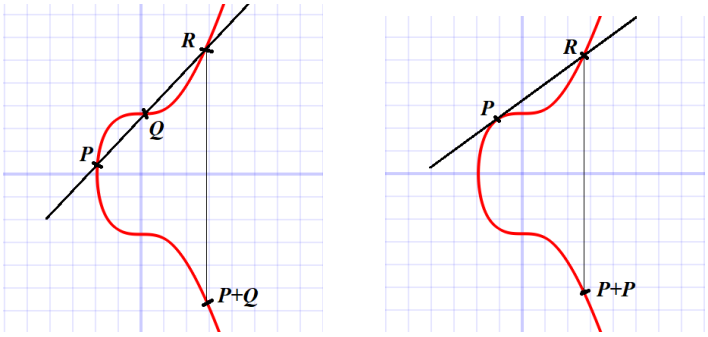

Figura 4: Suma de puntos en una curva elíptica.

- Sean dos puntos distintos P y Q de la curva, tales que la recta que los une no es vertical. Si R es el punto de corte de la recta con la curva, la suma $P+Q$ es el punto simétrico de R respecto del eje x. Si la recta que los une es vertical, entonces, $P+Q=0$.
- Si P es un punto de la curva tal que la recta tangente a la curva en ese punto corta a la curva en el punto R, entonces la suma $P+P$ es el simétrico de R respecto del eje x. Si la recta tangente en P es vertical, entonces $P+P=0$.

Esta suma descrita de forma geométrica se traslada a la curva elíptica que hemos definido sobre un cuerpo de cardinal primo $\mathbb{F}_p$, con $p>3$. Sea $E(\mathbb{F}_p)$ el conjunto de los puntos con coordenadas en $\mathbb{F}_p$ de dicha curva elíptica, junto con un punto que, como antes, se denomina *punto del infinito* y que denotamos por 0. Siguiendo la definición geométrica de la suma sobre los puntos de la curva, en $E(\mathbb{F}_p)$ se define la suma como sigue (todas las operaciones aritméticas se hacen en $\mathbb{F}_p$, es decir, son operaciones módulo p):

1. $P + 0 = 0 + P = P$, $\forall P \in E(\mathbb{F}_p)$.
2. Si $P = (x, y) \in E(\mathbb{F}_p)$, entonces $(x, -y) \in E(\mathbb{F}_p)$ y se define $(x, y)+(x, -y) = 0$. El punto $(x, -y)$ se denomina $-P$ y se dice que es el opuesto de P. Además, $-0 = 0$.
3. Si $P = (x_1, y_1)$ y $Q = (x_2, y_2)$ son dos puntos de $E(\mathbb{F}_p)$ tales que $P \neq \pm Q$, entonces $P + Q = (x_3, y_3)$ donde

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1$$

con $m = (y_2 - y_1)/(x_2 - x_1)$.

4. Si $P = (x_1, y_1)$ es un punto de $E(\mathbb{F}_p)$ tal que $P \neq -P$, entonces $P+P = (x_3, y_3)$ donde

$$x_3 = m^2 - 2x_1, \quad y_3 = m(x_1 - x_3) - y_1$$

con $m = 3x_1^2 + a/(2y_1)$.

Con esta suma, el conjunto de puntos $E(\mathbb{F}_p)$ es un grupo abeliano finito.

El problema del logaritmo discreto elíptico (PLDE)

El problema del logaritmo discreto elíptico (PLDE) consiste en, dada una curva elíptica E definida sobre $\mathbb{F}_p$, un punto $P \in E(\mathbb{F}_p)$ de orden n y un punto $Q \in E(\mathbb{F}_p)$, hallar el entero s con $0 \leq s < n$, tal que $Q = P + \cdots + P$ (o sea, $Q = sP$), siempre que exista, esto es, si $Q \in \langle P \rangle$.

Aunque todas las curvas elípticas sobre un cuerpo primo $\mathbb{F}_p$ son estructuralmente similares, la dificultad del PLDE depende de los parámetros elegidos, en particular del orden $\#E(\mathbb{F}_p)$ de la curva elíptica. Para elegir una curva elíptica para uso criptográfico hay que elegir un número primo p grande tal que el orden de la curva $\#E(\mathbb{F}_p)$ sea un número primo o, al menos, el producto de un primo por un entero pequeño. También hay que descartar los casos muy improbables en los que la curva sea anómala, $\#E(\mathbb{F}_p) = p$, o supersingular, $\#E(\mathbb{F}_p) = p + 1$.

El último paso es seleccionar un punto $P \in E(\mathbb{F}_p)$ que sirva de base para el logaritmo discreto. Elegir un punto que sea generador de $E(\mathbb{F}_p)$ o, en su defecto, de orden grande, no tiene mayor dificultad. Para más detalles y demostraciones sobre criptografía y curvas elípticas, consultar [2].

Tanto el NIST, el Instituto Nacional de Estándares y Tecnología de Estados Unidos ([4]), como el Standards for Efficient Cryptography Groups ([1]), han publicado listas de curvas elípticas recomendadas para distintos tamaños de los parámetros.

Algoritmo de firma digital

A continuación se describe el algoritmo de firma digital ECDSA (Elliptic Curve Digital Signature Algorithm) de clave privada y pública basado en una curva elíptica criptográfica, que es estándar propuesto por el NIST en [4]. Para obtener las claves

pública y privada, se comienza eligiendo un escalar d tal que $0 < d < n$. Este número es la clave privada. A partir de este número y del punto base P, se calcula

$$Q = P + \cdots + P \quad \text{en } \mathbb{F}_p \quad (d \text{ sumandos}), \quad \text{esto es,} \quad Q = dP.$$

Entonces $Q = (x, y)$ es la clave pública.

Esta clave pública se expresa de forma no comprimida con las dos coordenadas xy seguidas y precedidas del byte indicativo `04`. En forma comprimida se expresa con la sola coordenada x. Como a partir de x se obtienen dos valores opuestos y_1 e y_2, uno par y otro impar; si se elige el par, delante de la x se añade el identificativo `02` y si se elige el impar, se añade el indicativo `03`. Un ejemplo de clave privada es

$$d = 87443244391001342896527975361878627920717378785443876955045552084285386647786;$$

en forma hexadecimal,

`c1531f574fc8a64ae5cfdc1cf87b969f4385f862c8c3348a006c6ae013ffe8ea`.

La correspondiente clave pública en forma comprimida es

`020766fd4ffbbfa2086a96b335ba6eff24b58e4819936f8dc9d4002dabc9b11405`.

Supongamos que Ander desea firmar un mensaje m con el algoritmo criptográfico de Bitcoin que se acaba de introducir. Previamente Ander elige la clave privada d y, junto con el punto base P, calcula la clave pública $Q = dP$. Para firmar el mensaje, Ander:

- Calcula el resumen e de m con la función SHA-256.
- Elige un número aleatorio k tal que $0 < k < n$ y calcula $kP = (x, y)$.
- Obtiene el valor $r = x \mod p$. Si $r = 0$, elige otro k.
- Calcula $s = k^{-1}(e + dr) \mod p$. Si $s = 0$, elige otro k.
- Tiene así la firma digital de m, que es el par (r, s).

Ander publica el mensaje m y su resumen e, junto con la firma digital (r, s).

Si Beatriz quiere verificar la firma:

- Comprueba si r y s son enteros. Si no es así, la firma no es válida.
- Comprueba que e es el resumen de m con la función SHA-256.
- Calcula $w = s^{-1} \mod p$.
- Calcula $u = es^{-1}$ y $v = rs^{-1}$.
- Calcula $R = uP + vQ = (x, y)$.
- Acepta la firma como válida si y sólo si la coordenada x de R es $x = r \mod p$.

La justificación es que

$$(x, y) = uP + vQ = es^{-1}P + rs^{-1}Q = e\,\frac{k}{e + dr}\,P + r\,\frac{k}{e + dr}\,dP = kP \quad \text{en } \mathbb{F}_p,$$

por lo que necesariamente tiene que ocurrir $x = r \mod p$.

Es importante destacar que, para cada firma, Ander debe elegir un k distinto, ya que en caso contrario se descubriría la clave privada d.

La seguridad de este procedimiento de firma deriva de la dificultad del PLDE, ya que se firma con la clave privada d y sólo se puede validar la firma con la correspondiente clave pública Q, y hoy en día la obtención de la clave privada a partir de la clave pública no es computable: del conocimiento de la clave pública no se puede obtener computacionalmente la clave privada.

AGRADECIMIENTOS. El autor agradece a Mikel Peñagaritano, de la UPV/EHU, y al revisor de *La Gaceta* sus observaciones y recomendaciones en la fase de revisión del artículo.

REFERENCIAS

[1] CERTICOM RESEARCH, *Standards for Efficient Cryptography 2 (SEC 2): Recommended Elliptic Curve Domain Parameters*, versión 2.0 (2010), `https://www.secg.org/sec2-v2.pdf`.

[2] JOSÉ LUIS GÓMEZ PARDO, Criptografía y curvas elípticas, *Gac. R. Soc. Mat. Esp.* **5** (2002), no. 3, 737–777.

[3] SATOSHI NAKAMOTO, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008), `https://bitcoin.org/bitcoin.pdf`.

[4] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, *Digital Signature Standard*, versión 4 (2013), `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`.

[5] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, *Secure Hash Standard*, versión 4 (2015), `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf`.

[6] BART PRENEEL, HANS DOBBERTIN Y ANTOON BOSSELAERS, The Cryptographic Hash Function RIPEMD-160, CryptoBytes **3** (2), (1997), 9–14, disponible en `https://www.esat.kuleuven.be/cosic/publications/article-317.pdf`.

MIKEL LEZAUN, DEPARTAMENTO DE MATEMÁTICAS, UNIVERSIDAD DEL PAÍS VASCO/EUSKAL HERRIKO UNIBERTSITATEA - UPV/EHU, 48940 LEIOA (BIZKAIA)
Correo electrónico: `mikel.lezaun@ehu.eus`