EL DOCUMENTO QUE LO EMPEZÓ TODO · 2008

BITCOIN

Un Sistema de Efectivo Electrónico de Persona a Persona

EL WHITEPAPER · TRADUCIDO AL ESPAÑOL

SATOSHI NAKAMOTO

31 DE OCTUBRE DE 2008 · 9 PÁGINAS QUE CAMBIARON EL DINERO

TRADUCCIÓN Y EDICIÓN

MATRIX ACADEMY · CRYPTOMATA

ESTÁS A PUNTO DE LEER LA FUENTE ORIGINAL

El 31 de octubre de 2008, en medio de la peor crisis financiera en décadas, alguien bajo el seudónimo de Satoshi Nakamoto publicó este documento en una lista de correos de criptografía. Nueve páginas. Sin empresa, sin marketing, sin pedirle permiso a nadie.

Lo que tienes en las manos no es un resumen ni una opinión: es el **documento fundador de Bitcoin, traducido fielmente al español**. Todo lo que existe hoy — los exchanges, las wallets, los millones de personas usando crypto — nace de estas páginas.

Te doy tres consejos para leerlo:

- No necesitas entenderlo todo. Ni los ingenieros lo entendieron completo en 2008. Lee el resumen, la introducción y la conclusión; las secciones técnicas puedes saltarlas y regresar después.
- Fíjate en el problema, no en la fórmula. Todo el documento responde una sola pregunta: ¿cómo dos desconocidos pueden intercambiar dinero sin un banco de por medio?
- Recuerda el contexto. Esto se escribió cuando los bancos quebraban y los gobiernos los rescataban con dinero de todos. No es casualidad.

> **Nadie te puede volver a decir que Bitcoin "no tiene fundamento". El fundamento son estas 9 páginas, y ahora las tienes en tu idioma.**

NOTA DE ESTA EDICIÓN

Traducción al español realizada por Matrix Academy con fines educativos. El documento original, "Bitcoin: A Peer-to-Peer Electronic Cash System", fue publicado por Satoshi Nakamoto bajo licencia MIT y está disponible en bitcoin.org/bitcoin.pdf. Los diagramas fueron recreados fielmente a partir de los originales.

— Cryptomata

OCTAVIO MATA · MATRIX ACADEMY

BITCOIN: UN SISTEMA DE EFECTIVO ELECTRÓNICO DE PERSONA A PERSONA

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

RESUMEN

Una versión puramente de persona a persona del efectivo electrónico permitiría enviar pagos en línea directamente de una parte a otra sin pasar por una institución financiera. Las firmas digitales aportan parte de la solución, pero los principales beneficios se pierden si todavía se requiere un tercero de confianza para prevenir el doble gasto. Proponemos una solución al problema del doble gasto usando una red de persona a persona. La red pone sello de tiempo a las transacciones incluyéndolas en una cadena continua de prueba de trabajo basada en hashes, formando un registro que no puede modificarse sin rehacer la prueba de trabajo. La cadena más larga no solo sirve como prueba de la secuencia de eventos presenciados, sino como prueba de que provino del mayor conjunto de poder de cómputo. Mientras la mayoría del poder de cómputo esté controlado por nodos que no cooperan para atacar la red, estos generarán la cadena más larga y superarán a los atacantes. La red en sí requiere una estructura mínima. Los mensajes se difunden sobre la base del mejor esfuerzo, y los nodos pueden salir y volver a unirse a la red a voluntad, aceptando la cadena de prueba de trabajo más larga como prueba de lo que ocurrió durante su ausencia.

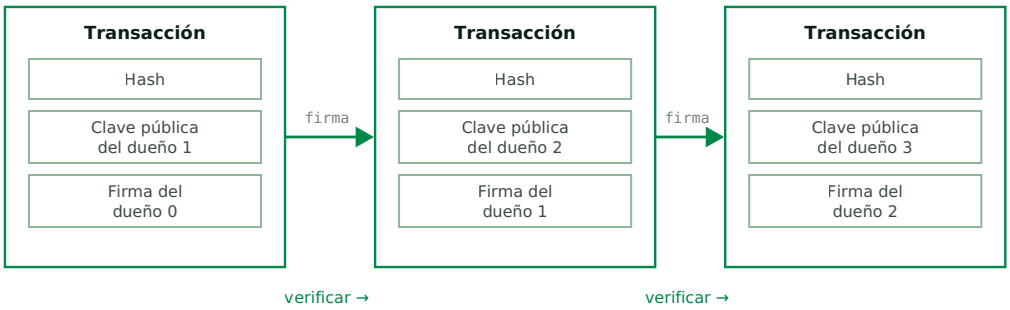
1. Introducción

El comercio en Internet ha llegado a depender casi exclusivamente de instituciones financieras que actúan como terceros de confianza para procesar los pagos electrónicos. Aunque el sistema funciona suficientemente bien para la mayoría de las transacciones, todavía sufre las debilidades inherentes del modelo basado en confianza. Las transacciones completamente irreversibles no son realmente posibles, ya que las instituciones financieras no pueden evitar mediar en disputas. El costo de la mediación incrementa los costos de transacción, limitando el tamaño mínimo práctico de una transacción y eliminando la posibilidad de pequeñas transacciones casuales, y existe un costo mayor en la pérdida de la capacidad de hacer pagos irreversibles por servicios irreversibles. Con la posibilidad de reversión, la necesidad de confianza se extiende. Los comerciantes deben desconfiar de sus clientes, molestándolos para obtener más información de la que de otro modo necesitarían. Un cierto porcentaje de fraude se acepta como inevitable. Estos costos e incertidumbres de pago pueden evitarse en persona usando moneda física, pero no existe ningún mecanismo para realizar pagos a través de un canal de comunicación sin un tercero de confianza.

Lo que se necesita es un sistema de pago electrónico basado en prueba criptográfica en lugar de confianza, que permita a dos partes dispuestas transaccionar directamente entre sí sin la necesidad de un tercero de confianza. Las transacciones que sean computacionalmente imprácticas de revertir protegerían a los vendedores del fraude, y los mecanismos rutinarios de depósito en garantía podrían implementarse fácilmente para proteger a los compradores. En este documento proponemos una solución al problema del doble gasto usando un servidor de sellos de tiempo distribuido de persona a persona, para generar prueba computacional del orden cronológico de las transacciones. El sistema es seguro mientras los nodos honestos controlen colectivamente más poder de cómputo que cualquier grupo cooperante de nodos atacantes.

2. Transacciones

Definimos una moneda electrónica como una cadena de firmas digitales. Cada dueño transfiere la moneda al siguiente firmando digitalmente un hash de la transacción anterior y la clave pública del siguiente dueño, y añadiendo ambos al final de la moneda. El receptor de un pago puede verificar las firmas para verificar la cadena de propiedad.



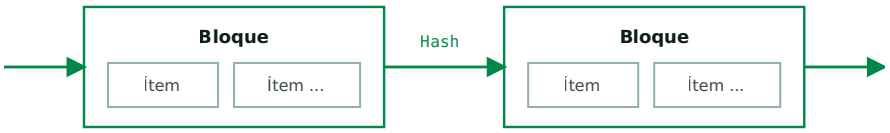
CADA DUEÑO FIRMA EL HASH ANTERIOR + LA CLAVE PÚBLICA DEL SIGUIENTE DUEÑO

El problema, por supuesto, es que el receptor no puede verificar que uno de los dueños no haya gastado dos veces la moneda. Una solución común es introducir una autoridad central de confianza, o casa de moneda, que revise cada transacción en busca de doble gasto. Después de cada transacción, la moneda debe regresar a la casa de moneda para emitir una nueva, y solo las monedas emitidas directamente por ella son confiables de no haber sido gastadas dos veces. El problema con esta solución es que el destino de todo el sistema monetario depende de la empresa que opera la casa de moneda, con cada transacción teniendo que pasar por ella, exactamente igual que un banco.

Necesitamos una forma de que el receptor sepa que los dueños anteriores no firmaron transacciones previas. Para nuestros propósitos, la transacción más antigua es la que cuenta, así que no nos importan los intentos posteriores de doble gasto. La única forma de confirmar la ausencia de una transacción es conocer todas las transacciones. En el modelo basado en casa de moneda, esta conocía todas las transacciones y decidía cuál llegó primero. Para lograrlo sin un tercero de confianza, las transacciones deben anunciarse públicamente [1], y necesitamos un sistema para que los participantes acuerden una única historia del orden en que fueron recibidas. El receptor necesita prueba de que, al momento de cada transacción, la mayoría de los nodos acordó que fue la primera recibida.

3. Servidor de Sellos de Tiempo

La solución que proponemos comienza con un servidor de sellos de tiempo. Este funciona tomando el hash de un bloque de elementos a sellar y publicando ampliamente ese hash, como en un periódico o una publicación de Usenet [2-5]. El sello de tiempo demuestra que los datos debieron existir en ese momento, obviamente, para poder entrar en el hash. Cada sello de tiempo incluye el sello anterior en su hash, formando una cadena, donde cada sello adicional refuerza a los que le preceden.

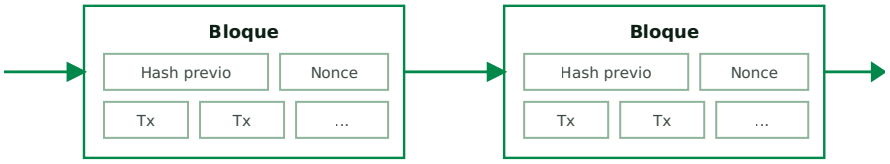


CADA SELLO INCLUYE EL HASH DEL ANTERIOR, FORMANDO UNA CADENA

4. Prueba de Trabajo

Para implementar un servidor de sellos de tiempo distribuido de persona a persona, necesitamos usar un sistema de prueba de trabajo similar al Hashcash de Adam Back [6], en lugar de publicaciones en periódicos o Usenet. La prueba de trabajo consiste en buscar un valor que, al ser procesado con un hash, como SHA-256, produzca un hash que comience con cierto número de bits en cero. El trabajo promedio requerido es exponencial en el número de bits en cero requeridos, y puede verificarse ejecutando un solo hash.

Para nuestra red de sellos de tiempo, implementamos la prueba de trabajo incrementando un nonce en el bloque hasta encontrar un valor que le dé al hash del bloque los bits en cero requeridos. Una vez que el esfuerzo de cómputo se ha gastado para satisfacer la prueba de trabajo, el bloque no puede modificarse sin rehacer todo el trabajo. Como los bloques posteriores se encadenan después de él, el trabajo para modificar un bloque incluiría rehacer todos los bloques posteriores.



LA CADENA DE BLOQUES: CADA BLOQUE REFERENCIA EL HASH DEL ANTERIOR

La prueba de trabajo también resuelve el problema de determinar la representación en la decisión por mayoría. Si la mayoría se basara en una-dirección-IP-un-voto, podría ser subvertida por quien pudiera asignarse muchas IPs. La prueba de trabajo es esencialmente **un-CPU-un-voto**. La decisión de la mayoría está representada por la cadena más larga, que tiene el mayor esfuerzo invertido. Si la mayoría del poder de cómputo está en nodos honestos, la cadena honesta crecerá más rápido y superará a cualquier competidora. Para modificar un bloque pasado, un atacante tendría que rehacer su prueba de trabajo y la de todos los bloques posteriores, y luego alcanzar y superar el trabajo de los nodos honestos. Más adelante mostraremos que la probabilidad de que un atacante más lento los alcance disminuye exponencialmente conforme se añaden bloques.

Para compensar el incremento en la velocidad del hardware y el interés variable en operar nodos a lo largo del tiempo, la dificultad de la prueba de trabajo se determina mediante un promedio móvil que apunta a un número promedio de bloques por hora. Si se generan demasiado rápido, la dificultad aumenta.

5. Red

Los pasos para operar la red son los siguientes:

1. Las nuevas transacciones se difunden a todos los nodos.
2. Cada nodo agrupa las nuevas transacciones en un bloque.
3. Cada nodo trabaja en encontrar una prueba de trabajo difícil para su bloque.
4. Cuando un nodo encuentra una prueba de trabajo, difunde el bloque a todos los nodos.
5. Los nodos aceptan el bloque solo si todas sus transacciones son válidas y no han sido gastadas.
6. Los nodos expresan su aceptación del bloque trabajando en crear el siguiente bloque de la cadena, usando el hash del bloque aceptado como el hash previo.

Los nodos siempre consideran correcta a la cadena más larga y seguirán extendiéndola. Si dos nodos difunden a la vez versiones distintas del siguiente bloque, algunos recibirán una u otra primero; trabajan sobre la primera recibida y guardan la otra rama por si llega a ser más larga. El empate se rompe con la siguiente prueba de trabajo: una rama se vuelve más larga y los nodos de la otra se cambian a ella.

Las difusiones de nuevas transacciones no necesitan necesariamente llegar a todos los nodos. Mientras lleguen a muchos nodos, entrarán en un bloque en poco tiempo. Las difusiones de bloques también son tolerantes a mensajes perdidos. Si un nodo no recibe un bloque, lo solicitará cuando reciba el siguiente y se dé cuenta de que le faltó uno.

6. Incentivo

Por convención, la primera transacción de un bloque es una transacción especial que inicia una nueva moneda cuyo dueño es el creador del bloque. Esto añade un incentivo para que los nodos apoyen la red, y proporciona una forma de distribuir inicialmente monedas en circulación, dado que no existe una autoridad central que las emita. La adición constante de una cantidad fija de monedas nuevas es análoga a los mineros de oro que gastan recursos para añadir oro a la circulación. En nuestro caso, lo que se gasta es tiempo de CPU y electricidad.

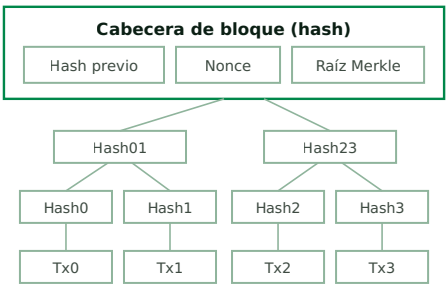
El incentivo también puede financiarse con comisiones de transacción. Si el valor de salida de una transacción es menor que su valor de entrada, la diferencia es una comisión que se suma al valor de incentivo del bloque que contiene la transacción. Una vez que un número predeterminado de monedas haya entrado en circulación, el incentivo puede pasar por completo a las comisiones de transacción y ser totalmente libre de inflación.

El incentivo puede ayudar a alentar a los nodos a mantenerse honestos. Si un atacante codicioso logra reunir más poder de cómputo que todos los nodos honestos, tendría que elegir entre usarlo para defraudar a la gente robando de vuelta sus pagos, o usarlo para generar monedas nuevas. Debería resultarle **más rentable jugar bajo las reglas** — reglas que lo favorecen con más monedas nuevas que las de todos los demás combinados — que socavar el sistema y la validez de su propia riqueza.

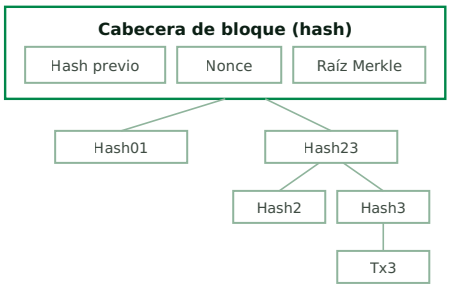
7. Recuperación de Espacio en Disco

Una vez que la última transacción de una moneda queda enterrada bajo suficientes bloques, las transacciones gastadas anteriores a ella pueden descartarse para ahorrar espacio en disco. Para facilitarlo sin romper el hash del bloque, las transacciones se procesan en un Árbol de Merkle [7][2][5], incluyendo solo la raíz en el hash del bloque. Los bloques viejos pueden entonces compactarse podando ramas del árbol. Los hashes interiores no necesitan almacenarse.

Una cabecera de bloque sin transacciones ocuparía unos 80 bytes. Si suponemos que los bloques se generan cada 10 minutos: 80 bytes × 6 × 24 × 365 = 4.2 MB por año. Con los sistemas de cómputo vendiéndose típicamente con 2 GB de RAM en 2008, y la Ley de Moore prediciendo un crecimiento de 1.2 GB por año, el almacenamiento no debería ser un problema incluso si las cabeceras de bloque deben mantenerse en memoria.



TRANSACCIONES EN ÁRBOL DE MERKLE



DESPUÉS DE PODAR TX0 A TX2

8. Verificación de Pago Simplificada

Es posible verificar pagos sin operar un nodo completo de la red. Un usuario solo necesita conservar una copia de las cabeceras de bloque de la cadena de prueba de trabajo más larga — que puede obtener consultando a los nodos de la red hasta convencerse de que tiene la cadena más larga — y obtener la rama de Merkle que vincula la transacción con el bloque en el que está sellada. No puede verificar la transacción por sí mismo pero, al vincularla a un lugar en la cadena, puede ver que un nodo de la red la aceptó, y los bloques añadidos después la confirman aún más.

Como tal, la verificación es confiable mientras los nodos honestos controlen la red, pero es más vulnerable si la red es dominada por un atacante. Aunque los nodos de la red pueden verificar las transacciones por sí mismos, el método simplificado puede ser engañado por transacciones fabricadas por un atacante mientras este pueda seguir dominando la red. Una estrategia para protegerse sería aceptar alertas de los nodos cuando detecten un bloque inválido, indicando al software del usuario descargar el bloque completo y las transacciones alertadas para confirmar la inconsistencia. Los negocios que reciben pagos frecuentes probablemente querrán seguir operando sus propios nodos, para una seguridad más independiente y una verificación más rápida.

9. Combinación y División de Valor

Aunque sería posible manejar las monedas individualmente, sería poco práctico hacer una transacción separada por cada centavo de una transferencia. Para permitir que el valor se divida y combine, las transacciones contienen múltiples entradas y salidas. Normalmente habrá una sola entrada proveniente de una transacción anterior más grande, o múltiples entradas que combinan cantidades menores, y a lo sumo dos salidas: una para el pago y otra que devuelve el cambio, si lo hay, al emisor.

MÚLTIPLES ENTRADAS, A LO SUMO DOS SALIDAS: PAGO Y CAMBIO

Cabe señalar que la ramificación — donde una transacción depende de varias transacciones, y esas dependen de muchas más — no es un problema aquí. Nunca existe la necesidad de extraer una copia completa e independiente del historial de una transacción.

10. Privacidad

El modelo bancario tradicional logra un nivel de privacidad limitando el acceso a la información a las partes involucradas y al tercero de confianza. La necesidad de anunciar públicamente todas las transacciones excluye este método, pero la privacidad aún puede mantenerse rompiendo el flujo de información en otro punto: **manteniendo anónimas las claves públicas**. El público puede ver que alguien está enviando una cantidad a alguien más, pero sin información que vincule la transacción con nadie. Esto es similar al nivel de información publicado por las bolsas de valores, donde la hora y el tamaño de las operaciones individuales — la "cinta" — se hace público, pero sin decir quiénes fueron las partes.

Como cortafuegos adicional, debería usarse un nuevo par de claves para cada transacción, para evitar que se vinculen a un dueño común. Cierta vinculación sigue siendo inevitable en transacciones con múltiples entradas, que necesariamente revelan que sus entradas pertenecieron al mismo dueño. El riesgo es que, si se revela la identidad del dueño de una clave, la vinculación podría revelar otras transacciones que pertenecieron al mismo dueño.

11. Cálculos

Consideramos el escenario de un atacante intentando generar una cadena alternativa más rápido que la cadena honesta. Incluso si lo lograra, esto no abre el sistema a cambios arbitrarios, como crear valor de la nada o tomar dinero que nunca perteneció al atacante. Los nodos no van a aceptar una transacción inválida como pago, y los nodos honestos nunca aceptarán un bloque que las contenga. Un atacante solo puede intentar cambiar una de sus propias transacciones para recuperar dinero que gastó recientemente.

La carrera entre la cadena honesta y la cadena de un atacante puede caracterizarse como una Caminata Aleatoria Binomial. El evento de éxito es que la cadena honesta se extienda un bloque, aumentando su ventaja en +1, y el evento de fracaso es que la cadena del atacante se extienda un bloque, reduciendo la brecha en -1.

La probabilidad de que un atacante alcance a la cadena desde un déficit dado es análoga al problema de la Ruina del Jugador. Supongamos que un jugador con crédito ilimitado comienza en déficit y juega un número potencialmente infinito de intentos para alcanzar el punto de equilibrio. Podemos calcular la probabilidad de que alguna vez lo alcance — es decir, de que un atacante alcance a la cadena honesta — de la siguiente manera [8]:

p	probabilidad de que un nodo honesto encuentre el siguiente bloque
q	probabilidad de que el atacante encuentre el siguiente bloque
q_z	probabilidad de que el atacante alcance la cadena desde z bloques atrás

$$q_z = 1 \;\; si\ p \leq q \quad | \quad q_z = (q/p)^z \;\; si\ p > q$$

Dada nuestra suposición de que p > q, la probabilidad cae exponencialmente conforme aumenta el número de bloques que el atacante debe alcanzar. Con las probabilidades en su contra, si no logra un avance afortunado al principio, sus posibilidades se vuelven insignificantes conforme queda más rezagado.

Ahora consideramos cuánto necesita esperar el receptor de una nueva transacción antes de estar suficientemente seguro de que el emisor no puede cambiarla. Suponemos que el emisor es un atacante que quiere hacerle creer al receptor que le pagó durante un tiempo, para luego cambiar la transacción y pagarse de vuelta a sí mismo. El receptor será alertado cuando eso ocurra, pero el emisor espera que sea demasiado tarde.

El receptor genera un nuevo par de claves y entrega la clave pública al emisor poco antes de la firma. Esto evita que el emisor prepare una cadena de bloques por adelantado trabajando en ella continuamente hasta tener la suerte de adelantarse lo suficiente, y ejecutar la transacción en ese momento. Una vez enviada la transacción, el emisor deshonesto comienza a trabajar en secreto en una cadena paralela que contiene una versión alternativa de su transacción.

El receptor espera hasta que la transacción haya sido añadida a un bloque y z bloques se hayan encadenado después de ella. No conoce la cantidad exacta de progreso que el atacante ha logrado pero, suponiendo que los bloques honestos tomaron el tiempo promedio esperado por bloque, el progreso potencial del atacante seguirá una distribución de Poisson con valor esperado:

$$\lambda = z \cdot (q / p)$$

Para obtener la probabilidad de que el atacante aún pueda alcanzar la cadena, multiplicamos la densidad de Poisson por cada cantidad de progreso que pudo haber logrado, por la probabilidad de que pueda alcanzarla desde ese punto. Reacomodando para evitar sumar la cola infinita de la distribución, y convirtiéndolo a código C:

```
#include <math.h>
double ProbabilidadExitoAtacante(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double suma = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        suma -= poisson * (1 - pow(q / p, z - k));
    }
    return suma;
}
```

Ejecutando algunos resultados, vemos que la probabilidad cae exponencialmente con z:

```
q=0.1
z=0    P=1.0000000
z=1    P=0.2045873
z=2    P=0.0509779
z=3    P=0.0131722
z=4    P=0.0034552
z=5    P=0.0009137
z=6    P=0.0002428
z=7    P=0.0000647
z=8    P=0.0000173
z=9    P=0.0000046
z=10   P=0.0000012
```

```
q=0.3
z=0    P=1.0000000
z=5    P=0.1773523
z=10   P=0.0416605
z=15   P=0.0101008
z=20   P=0.0024804
z=25   P=0.0006132
z=30   P=0.0001522
z=35   P=0.0000379
z=40   P=0.0000095
z=45   P=0.0000024
z=50   P=0.0000006
```

Resolviendo para P menor a 0.1%:

```
P < 0.001
q=0.10  z=5      q=0.15  z=8      q=0.20  z=11     q=0.25  z=15
q=0.30  z=24     q=0.35  z=41     q=0.40  z=89     q=0.45  z=340
```

12. Conclusión

Hemos propuesto un sistema de transacciones electrónicas que no depende de la confianza. Comenzamos con el marco habitual de monedas hechas de firmas digitales, que proporciona un fuerte control de la propiedad pero está incompleto sin una forma de prevenir el doble gasto. Para resolverlo, propusimos una red de persona a persona que usa prueba de trabajo para registrar una historia pública de las transacciones, que rápidamente se vuelve computacionalmente impráctica de modificar para un atacante si los nodos honestos controlan la mayoría del poder de cómputo. La red es robusta en su simplicidad no estructurada. Los nodos trabajan todos a la vez con poca coordinación. No necesitan identificarse, ya que los mensajes no se enrutan a ningún lugar en particular y solo necesitan entregarse sobre la base del mejor esfuerzo. Los nodos pueden salir y volver a unirse a la red a voluntad, aceptando la cadena de prueba de trabajo como prueba de lo que ocurrió durante su ausencia. **Votan con su poder de cómputo**, expresando su aceptación de los bloques válidos trabajando en extenderlos, y rechazando los bloques inválidos negándose a trabajar sobre ellos. Cualquier regla e incentivo necesarios pueden hacerse cumplir con este mecanismo de consenso.

Referencias

[1] W. Dai, "b-money", http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila y J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements", en 20th Symposium on Information Theory in the Benelux, mayo de 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document", en Journal of Cryptology, vol. 3, no. 2, páginas 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping", en Sequences II: Methods in Communication, Security and Computer Science, páginas 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings", en Proceedings of the 4th ACM Conference on Computer and Communications Security, páginas 28-35, abril de 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure", http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems", en Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, páginas 122-133, abril de 1980.

[8] W. Feller, "An introduction to probability theory and its applications", 1957.

YA LEÍSTE LA FUENTE

AHORA YA SABES MÁS QUE EL 95%

La mayoría opina de Bitcoin sin haber leído estas 9 páginas. Tú acabas de hacerlo. El siguiente paso no es leer más: es aprender a usar lo que ya entiendes.

Escanea y mira el video de 3 minutos donde te explico las 5 formas de generar ingresos con crypto.

bitcoinlallavedelamatrix.com

¿Estás leyendo desde el celular? No escanees: escribe el enlace directo en tu navegador.

¿Aún no descargas la guía "Escapa de la Matrix"?
Las 5 llaves, el vocabulario completo y el mapa del escape te esperan en los recursos gratis de Matrix Academy → academy.matrixco.store

— Cryptomata

OCTAVIO MATA · MATRIX ACADEMY