El Libro Blanco de Bitcoin, de Satoshi Nakamoto

Isabel Díaz /(26 dic. 2013)
http://www.di-fusion.com/wp/el-libro-blanco-de-bitcoin-de-satoshi-nakamoto/

Bitcoin: un sistema de dinero electrónico entre iguales (P2P).

Satoshi Nakamoto

satoshin@gmx.com
www.bitcoin.org

Resumen. Una versión de dinero electrónico que se basa sólo en principios P2P[1] permitiría efectuar pagos en Internet de forma directa, entre un usuario y otro, sin tener que pasar a través de una entidad financiera. Las firmas digitales[2]representan parte de la solución, pero los beneficios más importantes se pierden cuando tiene que intervenir un tercero de confianza para evitar que se lleve a cabo un gasto doble[3]. En este documento se propone el uso de una red P2P como solución al problema del gasto doble. Mediante un sello de tiempo[4], la red marca cada transacción[5] al codificarla en un *hash*[6] e introducirla en una cadena[7] en desarrollo que se basa en estos *hashes* a modo de comprobante de trabajo[8], formando así un registro que no se puede modificar a menos que se rehaga el comprobante de trabajo. La extensión de la cadena no sólo constituye una prueba que describe la secuencia de eventos ocurridos, sino que al mismo tiempo demuestra que proviene de la mayor reserva de recursos de la CPU[9]. Se podrá generar una cadena ilimitada que se mantenga con ventaja ante los ataques siempre que los nodos que no tienen intención de atacar a la red controlen la mayor parte de recursos de la CPU. La red en sí requiere una estructura mínima: los mensajes se difunden a través del esfuerzo colectivo; los nodos pueden unirse a la red o abandonarla en cualquier momento, rigiéndose siempre por la cadena de comprobantes de trabajo más extensa, que les mostrará lo que ha ocurrido mientras no estaban.

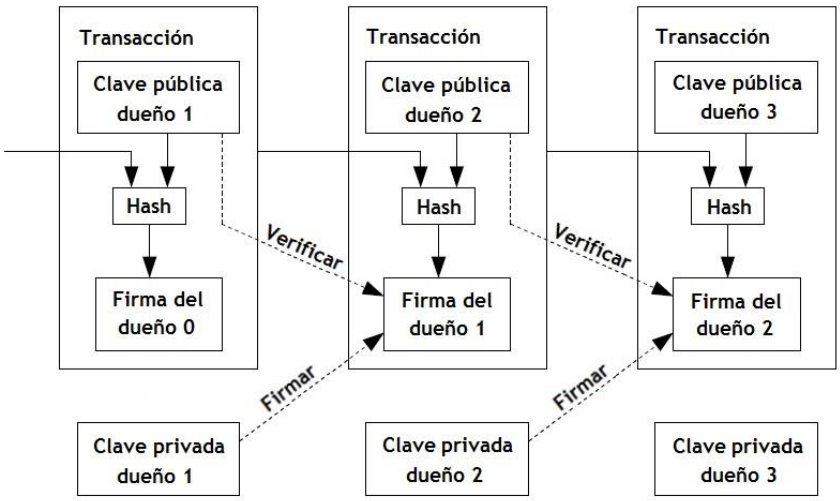
1. **Introducción**

El comercio en Internet depende casi de forma exclusiva de las entidades financieras, que actúan como terceros de confianza en el procesamiento de los pagos electrónicos. Aunque para la mayoría de transacciones resulte lo bastante efectivo, este sistema todavía carga consigo las debilidades inherentes del modelo de confianza en el que está basado. Las transacciones irreversibles no son del todo posibles, ya que las entidades financieras deben mediar para resolver las disputas. Esta intervención incrementa los costes de transacción, limitando así la cantidad mínima que resulta viable transferir y suprimiendo la opción de enviar pequeñas cantidades de dinero de forma puntual. Además, la imposibilidad de exigir pagos irreversibles a cambio de servicios que no se pueden devolver representa un coste aún mayor. La posibilidad de retroceder un pago hace que se extienda la necesidad de establecer una relación de confianza. Los comerciantes deben ser precavidos, de manera que exigen a sus clientes más información de la que hace falta. Resulta inevitable asumir cierto porcentaje de operaciones fraudulentas. Se pueden evitar estos costes, así como la inseguridad en los pagos, cuando se utiliza dinero físico en persona. Sin embargo, no existe ningún mecanismo que permita hacer pagos a través de otro canal de comunicación a menos que intervenga un tercero de confianza.

Se requiere un sistema de pago electrónico, basado en hechos criptográficos[10] y no en la confianza, que permita que dos individuos dispuestos a transferirse dinero realicen la transacción directamente sin precisar a un tercero de confianza. Las transacciones que no se pueden retroceder de ninguna manera a nivel computacional favorecerían a los vendedores ante posibles casos de estafa. Además, con el fin de proteger a los compradores, se podrían implementar con facilidad los mecanismos de garantía bloqueada[11] habituales. En este documento, se propone una solución al problema del doble gasto, que consiste en usar un servidor de sellado de tiempo[12] de tipo P2P para generar un registro computacional del orden cronológico de las transacciones. El sistema es seguro siempre que los nodos honrados controlen en conjunto un mayor porcentaje de los recursos de la CPU, manteniéndose con ventaja ante cualquier otro grupo de nodos que coopere en la red con intención de atacarla.

2. **Transacciones**

Definimos una moneda electrónica como la cadena de firmas digitales que la constituyen. Cada vez que un propietario transfiere una moneda a un nuevo poseedor, se firma digitalmente tanto el *hash* de la transacción previa como la clave pública del siguiente dueño. Estas firmas se añaden al final de la moneda correspondiente, de manera que el beneficiario de un pago puede comprobarlas para verificar la cadena de propiedad de dicha moneda.



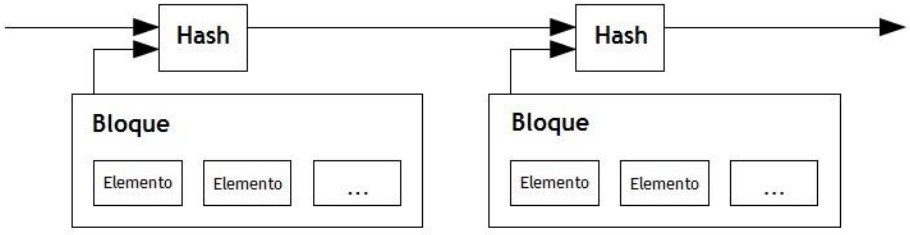
Resulta un problema evidente el hecho de que el beneficiario del cobro no sea capaz de comprobar si alguno de los propietarios ha realizado un doble gasto. Una solución habitual consiste en incorporar una autoridad central de confianza (una ceca o casa de monedas) que verifica cada transacción para comprobar que ningún fondo se haya gastado más de una vez. Tras cada transacción, la moneda debe volver a la ceca, que se encargará de emitir una nueva moneda. Con este sistema, sólo se puede confiar en las monedas emitidas directamente por la ceca para estar seguro de que no se ha producido un gasto doble. No obstante, esta solución presenta el inconveniente de que el destino del sistema monetario al completo depende de la entidad que controla la casa de monedas, ya que cada transacción tiene que pasar a través de ésta, como si se tratara de un banco.

Hace falta un sistema que permita al beneficiario de una transacción comprobar que los propietarios anteriores de ese dinero no han firmado ninguna transacción previa. Para este propósito, la primera operación es la que tiene valor, de manera que no se toma en cuenta ningún intento posterior para hacer un gasto doble. Sólo es posible verificar la ausencia de una transacción si se está al tanto de todas las operaciones. En el sistema de la

casa de monedas, esta autoridad está al corriente de todas las transacciones y decide cuál se efectúa antes. Para poder conseguir esto sin la necesidad de un intermediario de confianza, las transacciones deben ser públicas [1]. Por tanto, los participantes han de acordar un sistema que les permita acceder a un único historial que refleje el orden en el que se recibieron las transferencias. El beneficiario requiere una prueba que le demuestre que, en el momento en que se ejecuta la transacción, la mayor parte de nodos de la red coincide en que se trata de la primera petición.

3. **Servidor de sellado de tiempo**

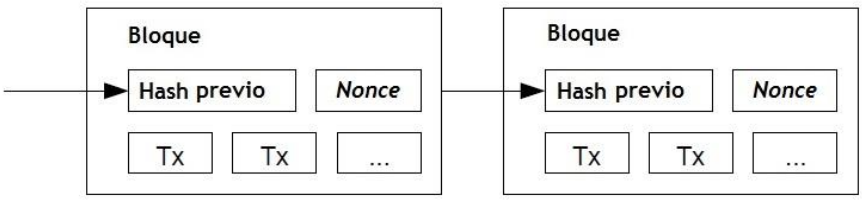
La solución que se propone comienza con un servidor de sellado de tiempo. Este tipo de sistemas tiene la función de asignar un sello cronológico al *hash* de cada bloque de elementos y publicar este *hash* para que la información sea de fácil acceso, como si se tratara de un periódico o de un artículo de Usenet[13] [2-5]. El sello de tiempo demuestra que la información existía en ese momento, como es evidente, ya que de lo contrario no podría formar parte del *hash*. Cada sello de tiempo contiene en su *hash* el sello cronológico previo, formando así una secuencia en cadena en la que cada sello refuerza los anteriores.



4. **Comprobante de trabajo**

Para implementar un servidor de sellado de tiempo en una red P2P, más que artículos de periódico o de Usenet, hace falta un sistema de comprobantes de trabajo similar a Hashcash de Adam Back [6]. El comprobante de trabajo implica la búsqueda de un valor que al ser codificado en un *hash*, como puede ser 'SHA-256′, produce un *hash* que comienza con un número determinado de bits a cero. El promedio del trabajo exigido es exponencial en el número de bits a cero requerido, y se puede comprobar al ejecutar un solo *hash*.

En nuestra red de sellado de tiempo, implementamos el comprobante de trabajo al ir añadiendo un número de un solo uso al bloque (*nonce*[14]), hasta que se halla el valor que proporciona el número de bits a cero requerido por el *hash*. Una vez que se destinan los recursos de la CPU a resolver el comprobante de trabajo, el bloque no se puede modificar a menos de que se repita la operación. A medida que se van enlazando nuevos bloques a la cadena, el trabajo que se requiere para cambiar un bloque implica rehacer también todos los bloques



posteriores. El comprobante de trabajo también sirve para resolver el problema que supone determinar la representación mayoritaria en la toma de decisiones. Si el recuento se basara en un sistema que atribuye un voto a cada IP, toda persona con

conocimientos para asignar múltiples direcciones IP podría manipular el resultado. El comprobante de trabajo asigna un voto a cada CPU. La voluntad de la mayoría está representada en la cadena de mayor extensión, que contiene el esfuerzo más grande de comprobantes de trabajo. Si la mayor parte de recursos de la CPU se encuentra controlada por nodos honrados, la cadena honrada crecerá más rápido que las demás y superará al resto de cadenas. Para modificar un bloque antiguo, el transgresor se vería obligado a rehacer el comprobante de trabajo del bloque en cuestión y, también, de todos los bloques posteriores. Por otro lado, tendría que alcanzar a los nodos honrados y superar su trabajo. Más adelante, demostraremos que la probabilidad de que un transgresor más lento se ponga al día disminuye de forma exponencial conforme se van agregando nuevos bloques a la cadena.

Para compensar el aumento de la velocidad de los sistemas informáticos y el interés variable de los nodos operativos a lo largo del tiempo, la dificultad asociada al comprobante de trabajo se determina mediante un promedio flexible que atiende al número de bloques que se produce de media en el transcurso de una hora. Si se generan demasiado rápido, la dificultad aumenta.

5. **Red**

Los pasos necesarios para el funcionamiento correcto de la red son los siguientes:

1. Las nuevas transacciones se comunican a todos los nodos.
2. Cada nodo agrupa las transacciones nuevas en un bloque.
3. Los nodos se esfuerzan para resolver el complejo comprobante de trabajo sujeto a su bloque.
4. Cuando un nodo resuelve el comprobante de trabajo, difunde el bloque a todos los nodos.
5. Los nodos sólo aceptan bloques si todas las transacciones que contienen son válidas y no han sido contabilizadas con anterioridad.
6. Los nodos dan su consentimiento para cada bloque cuando comienzan a trabajar en la creación del siguiente bloque de la cadena, empleando el *hash* del bloque aceptado como referencia para el próximo *hash*.

Los nodos siempre dan por válida la cadena más larga y trabajan para continuar extendiéndola. Si dos nodos difunden diferentes versiones para el siguiente bloque de forma simultánea, el resto de nodos recibirán una de las dos versiones en primer lugar. En ese caso, seguirán trabajando sobre la primera que reciban, pero guardarán la otra rama por si llegara a ser más larga. El desempate ocurrirá cuando se resuelva un nuevo comprobante de trabajo y una de las dos ramas adquiera mayor extensión. Los nodos que se encontraban trabajando en la otra rama pasarán a ocuparse de la nueva versión.

No es necesario que todos los nodos estén al tanto de las nuevas transacciones. Basta con que haya suficientes nodos que las procesen para que pasen a formar parte de un nuevo bloque en poco tiempo. La divulgación de los bloques también está sujeta a la posible pérdida del mensaje. En caso de que un nodo pierda un bloque, lo solicitará cuando reciba el siguiente bloque y se percate de que se ha saltado uno.

6. **Incentivo**

Por convención, la primera transacción dentro de un bloque es de carácter especial, ya que genera una moneda nueva destinada al creador de dicho bloque. Esto aporta un incentivo para que los nodos continúen ofreciendo

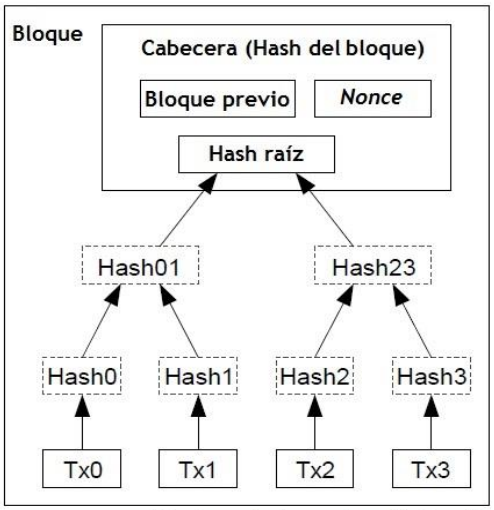
soporte a la red y, al mismo tiempo, permite en un inicio distribuir monedas con el fin de ponerlas en circulación, dado que no hay una autoridad central que las pueda emitir. La incorporación regular de una cantidad constante de monedas nuevas es análoga a los mineros de oro que invierten sus recursos para poner el oro en circulación. Este caso, los gastos representan el tiempo requerido por la CPU y el consumo eléctrico.

El incentivo también se puede financiar a través de las comisiones por transferencias. Cuando el valor de salida de una operación es inferior al valor de entrada, la diferencia se considera una tasa de servicio, que se añade al valor del incentivo asociado al bloque que contiene la transacción. Una vez que se introduce un número predeterminado de monedas en circulación, el incentivo puede basarse por completo en las comisiones por transferencias para no estar sujeto a ningún tipo de inflación.

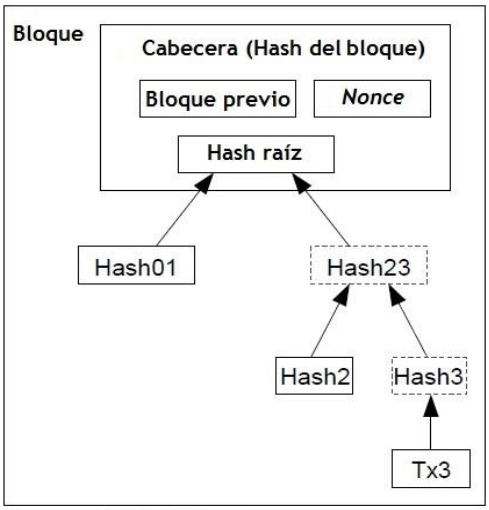
Por otro lado, el incentivo puede ayudar a mantener la honradez entre los nodos. Si se diera el caso de que un transgresor codicioso consiguiera producir una capacidad de CPU más potente que la de los nodos honrados, tendría que escoger entre la posibilidad de usarlo para recuperar sus fondos y defraudar así a los demás o aprovechar esta capacidad para generar nuevas monedas. La opción más rentable debería de ser la que obedece las normas (unas normas que le proporcionan a este individuo más monedas que a todos los otros nodos juntos), y no la que debilita al sistema y desaprueba la validez de las riquezas de uno mismo.

7. Recuperación de espacio en el disco

Cuando la última transacción que se realiza con una moneda determinada queda alojada bajo suficientes bloques, las transacciones previas asociadas a la misma se pueden desechar con el fin de ahorrar espacio en el disco. Para facilitar esta tarea sin tener que deshacer el *hash* del bloque, las transacciones se codifican en *hashes* estructurados en un árbol de Merkle[15] [7][2][5], donde sólo se incluye la raíz en el *hash* de dicho bloque. Los bloques anteriores se pueden compactar al recortar las ramas del árbol. De este modo, no es necesario almacenar los *hashes* interiores.



Transacciones codificadas en *hashes* según un árbol Merkle



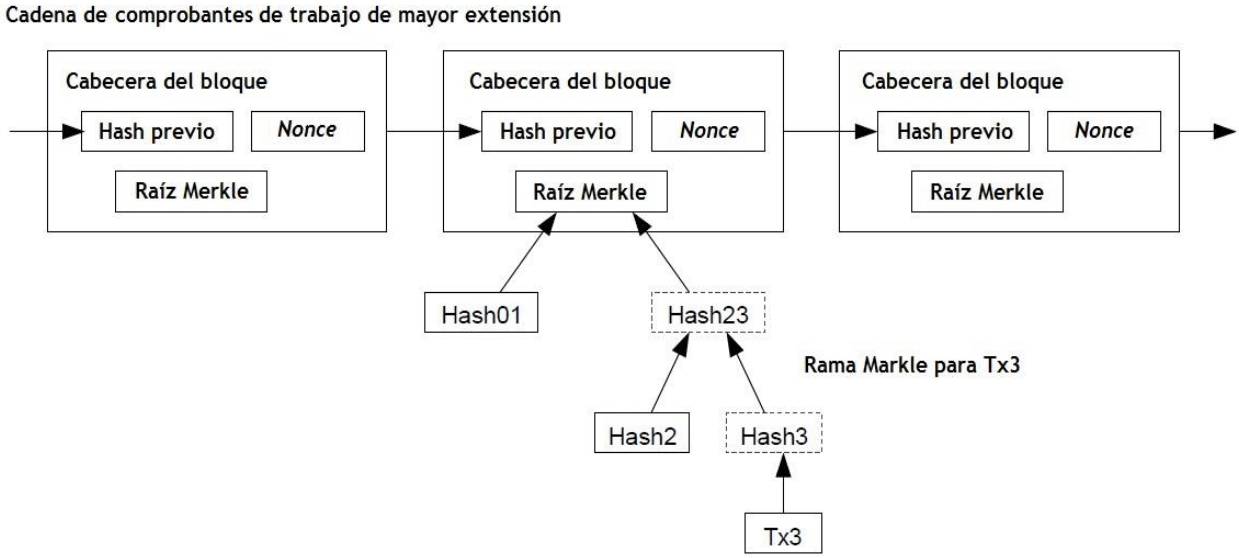
Tras cortar Tx0-2 del bloque

El encabezado de un bloque que no contiene transacciones ocuparía unos 80 bytes. Si partimos de que cada 10 minutos se genera un bloque nuevo, aplicamos el siguiente cálculo: 80 bytes * 6 * 24 * 365 = 4,2 MB al año.

Los sistemas informáticos que se comercializan a fecha de 2008 cuentan por lo general con 2 GB de memoria RAM. Por otro lado, la ley de Moore[16] estipula un crecimiento actual en la capacidad de almacenamiento de 1,2 GB al año. En vista de estos datos, el espacio no debería de plantear ningún problema si se tienen que guardar las cabeceras de los bloques en la memoria.

8. **Verificación de pago simplificada**

Es posible verificar los pagos sin tener que ejecutar todo un nodo en la red. Sólo es necesario guardar una copia de las cabeceras de los bloques que componen la cadena de comprobantes de trabajo más larga. Esta copia se obtiene mediante consultas a los nodos de la red, que el usuario puede realizar hasta que esté convencido de que dispone de la cadena más larga. De esta forma, obtendrá la rama de Merkle que vincula la transacción con el bloque donde ha sido registrado el sello de tiempo. Aunque el usuario no lo puede comprobar directamente, la transacción está registrada en un lugar concreto de la cadena, de manera que puede comprobar si algún nodo de la red la ha aceptado. A su vez, los bloques que se añaden con posterioridad confirman que la red ha aceptado la transacción.

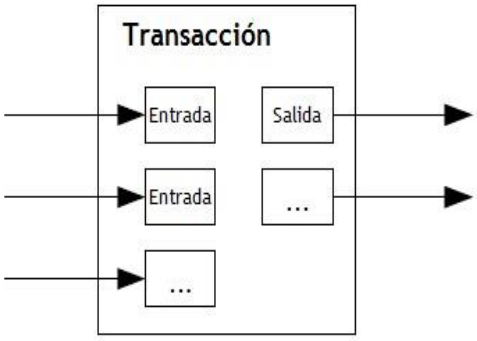


Con este sistema, la verificación resulta fiable siempre que los nodos honrados controlen la red. Sin embargo, es más vulnerable si la red está dominada por fuerzas transgresoras. A pesar de que los nodos de la red pueden verificar las transacciones directamente, el método simplificado puede ser falseado mediante transacciones inventadas por algún transgresor durante el espacio de tiempo que consiga dominar la red. Una posible solución para evitar este problema consistiría en recibir alertas de los nodos de la red cuando se detecte un bloque inválido. En estos casos, el software del usuario se descargaría el bloque entero, así como las transacciones sospechosas, para comprobar la incompatibilidad. Es probable que las empresas que reciben pagos con frecuencia prefieran operar sus propios nodos para una mayor seguridad independiente y una verificación más rápida.

9. **Combinación y partición del valor**

A pesar de que sería posible manejar las monedas de forma individual, resultaría poco práctico realizar una transacción independiente por cada céntimo que se quiera transferir. Para que sea posible dividir y combinar

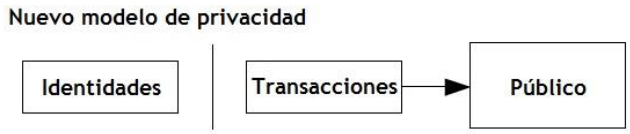
valores, las transacciones contienen diversas entradas y salidas. Por lo general, habrá una sola entrada que proviene de una transacción anterior de mayor tamaño, o bien, múltiples entradas que combinan cantidades más pequeñas. En cuanto a las salidas, habrá como máximo dos: una para el pago y otra para la devolución



del cambio al remitente en caso de que sea necesario. Cabe señalar que, cuando una transacción depende de múltiples transacciones y esas transacciones dependen a su vez de muchas otras, la difusión de estos datos no supone ningún problema. No existe la necesidad de extraer una copia independiente completa que refleje el historial de una transacción.

10. **Privacidad**

El modelo bancario tradicional consigue cierto nivel de privacidad al limitar el acceso a la información, que sólo está disponible para las partes involucradas y para el tercero de confianza. La necesidad de hacer públicas todas las transacciones va en contra de este método. No obstante, se puede conservar este nivel de privacidad si se detiene el flujo de la información en otro punto: al conservar la función anónima de las claves públicas. El público puede comprobar que alguien está mandando dinero a otra persona, pero no dispondrá de información que vincule la transacción a una persona determinada. Este sistema es similar al que se emplea en las bolsas de valores, donde la hora, la fecha y el importe de cada operación se hacen públicos, pero no se indica quiénes son las partes involucradas.



Con el fin de establecer un sistema de seguridad adicional, cada transacción debe ir asociada a un par de claves nuevo. De esta forma, se evita que alguien pueda vincular distintas transacciones a un mismo propietario. Resulta inevitable asumir cierto grado de vinculación cuando se trata de transacciones que contienen varias entradas, ya que en estos casos es posible constatar que todas las entradas pertenecían al mismo dueño. El riesgo está en que, si se descubre quién es el propietario de una clave, los vínculos podrían mostrar otras transacciones pertenecientes a la misma persona.

11. **Cálculos**

Se considera la posibilidad de que un transgresor intente generar una cadena alternativa más rápida que la cadena honrada. Aunque esto se consiga, el sistema no se ve sometido a cambios arbitrarios, tales como la creación de valor a partir de la nada o la adjudicación de fondos que no pertenecen al transgresor. Los nodos no aceptan transacciones inválidas a modo de pago, así que los nodos honrados rechazarán todo bloque que las contenga. Lo único que podría intentar el transgresor es cambiar una de sus propias transacciones para recuperar los fondos de una transferencia reciente.

La carrera entre la cadena honrada y la cadena transgresora se podría describir como un camino aleatorio[17] binomial. El caso de éxito ocurre cuando la cadena honrada consigue extenderse con un bloque de ventaja y sube una posición (+1), mientras que el caso fallido tendría lugar si el transgresor adquiere la ventaja de un bloque y acorta la distancia, restando así una posición (-1).
La probabilidad de que un transgresor en desventaja consiga ponerse al día es análoga al problema de la ruina del jugador[18]. Supongamos que un jugador con crédito ilimitado comienza a apostar con un déficit y tiene un sinfín de intentos para alcanzar el punto de equilibrio en el que ni gana ni pierde. De la siguiente manera se puede calcular la probabilidad de que el jugador llegue a este punto, o de que un transgresor alcance a la cadena honrada [8]:

- p = probabilidad de que un nodo honrado encuentre el siguiente bloque.
- q = probabilidad de que un transgresor encuentre el siguiente bloque.
- q_z = probabilidad de que el transgresor alcance a la cadena partiendo de una desventaja de z bloques.

$$q_z = \begin{cases} 1 & if\ p \leq q \\ (q/p)^z & if\ p > q \end{cases}$$

Dada la hipótesis de que $p > q$, la probabilidad cae de forma exponencial a medida que aumenta el número de bloques que el transgresor tiene por delante. Con las probabilidades en su contra, el transgresor va perdiendo opciones conforme se va quedando atrás, a menos que consiga hacer una embestida afortunada que lo posicione por delante desde un principio.
Por otro lado, se considera el tiempo que debe esperar el receptor de una nueva transacción hasta estar lo bastante seguro de que el remitente no puede modificar la operación. Asumimos que el remitente es un transgresor que quiere convencer al receptor de que ha pagado, pero luego pretende cambiar la operación para recuperar los fondos. El remitente tendrá la esperanza de que, para cuando el receptor se dé cuenta de lo ocurrido, sea ya demasiado tarde.

Ante esta situación, el beneficiario genera un nuevo par de claves y proporciona la clave pública al pagador poco antes de que se firme la operación. Esto evita que el pagador pueda preparar con antelación una cadena de bloques y trabajar en ésta de forma continua hasta tener la suerte de adquirir suficiente ventaja con respecto a la cadena principal, para así ejecutar la transacción en ese momento. Una vez que se envía la transferencia, el pagador deshonesto comienza a trabajar en secreto sobre una cadena paralela que contiene una versión alternativa de su transacción.

El receptor espera hasta que la transacción se introduzca en un bloque, que tiene enlazado un número de z bloques posteriores. Desconoce el progreso exacto que el agresor ha conseguido; pero, si suponemos que los bloques auténticos se han construido en un espacio de tiempo esperado, el progreso potencial del

$$\lambda = z\frac{q}{p}$$

transgresor será una distribución de Poisson[19] con valor esperado: Para calcular la probabilidad de que el transgresor alcance su objetivo, multiplicamos la densidad de la distribución de Poisson de cada cantidad de progreso posible por la probabilidad de que el agresor alcance a la cadena a partir de ese punto: Reorganización para evitar la suma

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

infinita de la distribución:
Convertido a código C:

```
#include

double AttackerSuccessProbability(double q, int z)

{

	double p = 1.0 - q;

	double lambda = z * (q / p);

	double sum = 1.0;

	int i, k;

	for (k = 0; k <= z; k++)

	{

		double poisson = exp(-lambda);

		for (i = 1; i <= k; i++)

			poisson *= lambda / i;

		sum -= poisson * (1 - pow(q / p, z - k));

	}
```

```
        return sum;

}
```

En vista de algunos resultados, se observa que la probabilidad disminuye de forma exponencial 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
```

Cálculo de *P* cuando es menor que 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**

El presente documento propone un sistema de transacciones electrónicas que no depende de la confianza. Comenzamos con la descripción de la infraestructura habitual en la que funcionan las monedas compuestas por firmas digitales. Este sistema brinda un fuerte control sobre la propiedad, pero queda incompleto si no existe un método para evitar el gasto doble. Para resolver esta cuestión, hemos propuesto una red P2P que emplea un sistema de comprobantes de trabajo mediante el que se registran las transacciones en un historial público. De esta forma, en poco tiempo resulta inviable a nivel computacional modificar las transacciones, siempre que los nodos honrados controlen la mayor parte de los recursos de la CPU. La red es robusta dentro de una simplicidad no estructurada. Los nodos funcionan de forma simultánea sin requerir gran coordinación. No necesitan identificarse, ya que los mensajes no se dirigen a un lugar concreto, sino que se difunden mediante el esfuerzo colectivo. Los nodos pueden entrar y salir de la red cuando lo deseen, rigiéndose siempre

por la cadena de comprobantes de trabajo que refleja lo que ha ocurrido mientras no estaban. Se vota con la potencia de la CPU. De esta forma, se puede expresar la aprobación de los bloques válidos en el momento en que se continúa trabajando para extender la cadena o, por el contrario, se pueden rechazar los bloques inválidos cuando se decide no trabajar sobre ellos. Toda regla o incentivo que se requiera puede acordarse a través de este sistema de consenso.

Fuentes

[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*, mayo1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document", en *Journal of Cryptology*, vol. 3, no. 2, págs. 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*, págs. 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*, págs. 28-35, abril 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, págs. 122-133, abril 1980.

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

Documento original

Bitcoin: A Peer-to-Peer Electronic Cash System

Notas

1**P2P**: del inglés *peer-to-peer*, describe un tipo de red descentralizada que permite compartir o transferir información entre dos o más nodos de forma directa. Las tareas se reparten entre los distintos nodos conectados, que ponen sus recursos a disposición de la red sin que sea necesario un servidor central que coordine las operaciones.

2**Firma digital:** mecanismo que permite al receptor de una transacción determinar el origen de la misma y confirmar que no ha sido alterada desde que se emitió.

3**Gasto doble**: acción que consiste en transferir los mismos fondos más de una vez.

[4]**Sello de tiempo**: secuencia de caracteres que revela la hora y la fecha en que ocurre una transacción o movimiento en la red.

[5]**Transacción**: movimiento de datos con firma digital que se registra en un bloque para así publicarlo en el libro de contabilidad de la red.

[6]***Hash***: función computable mediante un algoritmo que convierte datos arbitrarios en cadenas alfanuméricas de longitud fija. Estas cadenas de números y letras representan el resumen de toda la información que han procesado, pero a partir de ellas no se puede averiguar cuáles han sido los datos insertados. Sirven para compactar un conjunto de datos y aportar confidencialidad.

[7]**Cadena**: base de datos en la que se almacenan todas las transacciones que se realizan a través de los nodos de la red.

[8]**Comprobante de trabajo**: sistema de seguridad que consiste en solicitar un esfuerzo por parte de los clientes de la red para evitar posibles ataques o abusos en el servidor.

[9]**CPU**: del inglés *central processing unit* o unidad central de procesamiento, hace referencia a los componentes de ordenadores y otros dispositivos programables encargados de procesar los datos e interpretar las instrucciones de los programas.

[10]**Criptografía**: conjunto de técnicas que se aplican para alterar las representaciones lingüísticas de los mensajes mediante el cifrado o codificado con el fin de hacerlos ininteligibles ante personas no autorizadas.

[11]**Garantía bloqueada**: sistema que facilita el pago por adelantado con el fin de cubrir gastos futuros.

[12]**Servidor de sellado de tiempo**: unidad cuyo mecanismo permite establecer sellos de tiempo dentro de una red, aportando información relativa a la fecha y la hora de una transacción.

[13]**Usenet**: sistema de discusión en Internet que permite leer y enviar mensajes en forma de artículos a distintos grupos de noticias.

[14]**Nonce**: anglicismo empleado en criptografía para referirse a un número o una secuencia de números que se utilizan sólo una vez.

[15]**Árbol de Merkle**: estructura de árbol que se construye para organizar los *hashes* de manera que se pueda verificar de forma eficaz y segura los contenidos de una estructura de información más amplia.

[16]**Ley de Moore**: teorema que expresa que la capacidad de almacenamiento de un sistema informático se duplica cada cierto tiempo. En la actualidad, el número de transistores de un circuito integrado se multiplica por dos cada dos años aproximadamente.

[17]**Camino aleatorio**: conceptualización matemática que describe la trayectoria que resulta de dar pasos aleatorios de forma sucesiva.

[18]**Problema de la ruina del jugador**: incógnita que se resuelve al calcular la probabilidad de que un jugador venza a sus oponentes en un juego con un número indeterminado de partidas.

[19]**Distribución de Poisson**: teoría de probabilidad que calcula la posibilidad de que ocurra un número determinado de acontecimientos en un plazo de tiempo establecido.